Bodo Bauer, Michael Burghart, Roland Dyroff, Karl Eichwalder, Werner Fink, Klaus Franken, Jürgen Geck, Carsten Höger, Dirk Hohndel, Richard Jelinek, Florian La Roche, Volker Lendecke, Hans Lermen, Hubert Mantel, Martin Scherbaum, Burchard Steinbild, Klaus Wagner,

S.u.S.E. Linux 5.2

Installation, Configuration and First Steps

Your registration ID for technical support:
Support contact see: page 9

S.u.S.E., LLC
458 Santa Clara Avenue
Oakland, CA 94610
U.S.A.

Toll free phone number within the US and Canada:	1-8888-UR-LINUX (1-888-875-7873)
Phone.:	+1-510-TEL-SUSE (+1-510-835-7873)
Fax.:	+1-510-835-7875
email:	info@suse.com
	orders@suse.com
	support@suse.com
	webmaster@suse.com
WWW:	http://www.suse.com

Europe:
S.u.S.E. GmbH
Gebhardtstr. 2
D-90762 F—rth
Germany

Tel.:	+49-911-74053 31
Fax.:	+49-911-74177 55
E-Mail:	suse@suse.de
WWW:	http://www.suse.de

Bodo Bauer, Michael Burghart, Roland Dyroff, Karl Eichwalder, Werner Fink, Klaus Franken, Jürgen Geck, Carsten Höger, Dirk Hohndel, Richard Jelinek, Florian La Roche, Volker Lendecke, Hans Lermen, Hubert Mantel, Martin Scherbaum, Burchard Steinbild, Klaus Wagner,

Installation, Configuration and first steps with S.u.S.E. Linux 5.2
10^{th} edition 1998
S.u.S.E. GmbH

Layout: LaTeX 2ε
Cover layout: Sabine Zieseniss - using a picture by Stephan Endraß.
Printing: Graphische Betriebe F. Willmy GmbH, Nürnberg.

Contents

Chapter 1

Introduction

Since **Linus Torvalds** started working on a UNIX-like operating system for Intel-based PCs lots has been done. Linux has evolved from a “hacker’s toy” to a fully operational operating system whose power and efficency can compete with other operating systems (that became slow and heavy in the meantime).

Our joy concerning this fact does not only derive from mercantilism but from its inherited anarchic creativity. Even after long terms of development this idealism does still exist and is increasing. The more Linux tries to get hold of places where other operating system dominate the better the development succeeds. Up to now it is a serious alternative to these existing systems.

For many years, owners or users of high-end workstations were the only **Unix**™ users. This might be the reason for the fact that there was an air of mystery about Unix. Companies would offer training courses for their employees after changing to a Unix operating system (despite the fact that several feet of user manuals would usually come with the system). Now, Linux offers the PC user the chance to learn a multi-user-multi-tasking operating system at a reasonable price[1]. An increasing number of users will enter the Unix world through Linux.

There are many arguments in favour of Linux. First, a complete system is available for less than $50. When used commercially, an arbitrary number of machines can be run under Linux because there are no license fees. Furthermore there are many PC users who are turning to “fascination Linux” nowadays and want to adjust their amount of work individually instead of having to visit a three days crash course on Linux/Unix.

There are a lot of arguments in favour of Linux’ commercial side. Provided by Linux’ state of being free one can configure quite some machines with a powerful system instead of paying sums on end for proprietary Unix systems.

Linux is an ideal development platform because of source code compatibility with most Unix systems and the availability of the standard **OSF Motif** user interface. Apart from that, there is no cheaper alternative for an X-Terminal; because even a cheap, old 386 machine is sufficient for this task.

The strongest argument, however, is the availability of the complete source code. Apart from the fact that you can modify the system to suit your needs, you can find and correct bugs even if they are in the operating system itself.

[1] The first PC Unix system was **BSD-Unix**; it was born in the eighties

Instead of saying "we have to live with that" (as is the case with other popular operating systems), you can say "we have to change that". This is what makes the development of Linux so fast and leads to surprisingly robust versions. Linux is also very cooperative; it is no problem to install Linux together with other systems on your machine, and there are even ways to exchange data between the different operating systems.

1.1 What is "Linux"?

By now there are many different Linux distributions and versions. Therefore we would like to explain a few of the terms used here.

If we are talking about "Linux", we have to define what exactly we mean by that phrase. The "real Linux" is the *kernel*, the "heart" of every Unix operating system.

But a kernel alone does not make an operating system. Especially for Unix, there are several free software packages, and almost all known Unix tools are available under Linux. In most cases, these are the **GNU** versions of the well-known programs, most of which offer additional features (the **GNU-C/C++ compiler**, which some people think is one of the better compilers currently available, is probably the most famous program, perhaps with the exception of Emacs).

What makes Linux complete is **XFree86**™ (current version 3.3.2). This is an **X Window System** (Release 6.3) for PC-based Unix systems. This port is part of the official X11 R6.3 distribution of the **X Consortium, Inc.**, and therefore fully compatible with that standard.

All these components, together with many tools and goodies (e. g., games), make up the system known as *Linux*. As mentioned earlier, there is a large amount of free software available, and there are many ways to put a Linux system together. This is why there are so many different Linux-*distributions* on the market.

At last selection of a distribution is a question of philosophy of life which divides the Linux community into several ideologies.

1.2 Purpose of this book

The purpose of this book is to help you install Linux on your machine. It is not and cannot be a replacement for existing literature on Unix and the world of *high performance computing*.

It is difficult to write manuals for Linux. Because of the highly dynamic development of Linux (a new kernel patch every day is not uncommon) it is difficult to keep the manual up to date. But of course we try our best.

To install Linux for the first time, read and follow the simple instructions in chapter 3. We strongly recommend not to try to install Linux without studying this book first, unless you are an experienced Linux user who knows his way around. And even then, it might prove valuable to at least glance at the instructions to make sure things go smoothly.

Linux’ dynamic – sometimes in younger days, there were several kernel patches a day – makes it rather difficult to write a book that keeps pace and is adequate for beginners. This book now tries to fulfill this claim.

Instead of claiming that this is all complete we emphasize on helping new Linux users of finding their way and to support them while installing their system.

Installation This section escorts the Linux novice with a documented example installation (chapter 3). It shows error prone commands and sources and provides concrete help if encountering problems.

Here also the S.u.S.E. installation- and administration tool **YaST**™ is covered (chapter 4).

Configuration If the base system is up and running there are several other things that need to be done. This might be special hardware, e. g. sound cards (kernel parameter), **xfree**™ GUI or connection to the internet. This mainly is done from chapter 7 to chapter 18.

Crash Course Basic Linux commands are discussed in chapter 19, page 325.

Special topics For those of you who are familiar with Linux/Unix there are items that might even interest you. According to the actual version there are Linux specific topics such as compiling a kernel or function of the LILO bootloader. Furthermore there is an example configuration of an X11 window manager.

Appendix Besides a complete register of the most common errors/problems there is a list of important commands and keys as well as numerous example configuration files.

Index and glossary If you missed something you should consider having a look at the index or/and glossary.

1.3 Legend—or what does it mean: “root@earth: # ls”

In the installation guide we try to obey a general text layout; the same things should typographically look the same; the most important markup strategies are explained in table 1.1.

Text layout	Meaning
Linus Torvalds	important persons
GNU Emacs (**emacs**)	das program **GNU Emacs**, to be invoked with the command **emacs**
Applix*ware*	the product **Applix*ware***
/etc/passwd	file or directory name
<PATH>	name of the variable PATH
192.168.1.2	value of a variable
ls	the command **ls**
‘news’	the user ‘news’

Table 1.1: to be continued...

root@earth:/tmp # help	'root' shell in directory /tmp, type the command **help**
newbie@earth:/tmp > ls	shell of the user 'newbie' in directory /tmp, type the command **ls**
C:\> fdisk	DOS prompt, type the command **fdisk**
Alt	key to press, here the key "Alt"
Strg + Alt + Entf	with '+' are "added" all keys to be pressed at the same time; keys to be pressed sequentially are separated by a space only
"Permission denied"	system messagess
'System update'	menu entry 'System update'
Düsentrieb	company "Düsentrieb"
☞	Reference to the glossary (cf. chapter B)

Table 1.1: Legend of the text layout

1.4 The documentation

Instead of trying to cover everything, we've concentrated on helping beginners to install Linux and explore its fascinating possibilities. As the range of available software for Linux endlessly grows towards infinity, unfortunately it's impossible to describe everything in printed form. Nevertheless, we should not forget to mention the on-line documentation that is available.

1.4.1 Hypertext help

A major part of the documentation is available in *hypertext* form. To start the hypertext system, run **help**. If you are running X Window an additional program to read the documentation will be started. Additional options to the help system can be displayed by invoking

newbie@earth:/home/newbie > help --help

You can find the help system, in package susehelp, series docDocumentation.

1.4.2 Texinfo

Some program packages include documentation in Texinfo format, which is another hypertext variant. These files can be read with **Infoviewer** (**info**) or by using the **Emacs** (**emacs**) info-mode. The most compfortable way, of course is to use **tkInfo** (**tkinfo**) or the old fashioned **xinfo** (under X11).

1.4.3 Unix Manuals

The usual way to get information about programs or commands is to invoke the command **man**.

newbie@earth:/home/newbie > man <command>

displays the manual page for the command, which usually lists all command options and explains the command's usage.

1.4.4 FAQ, HowTo, and README files

The directory `/usr/doc` contains subdirectories with information about the corresponding packages. There you will often find the missing command option, the name of the configuration file you could not find anywhere else, or the name of the developer's dog. In any case, it is worth looking there before dumping the software.

Especially important is the directory `/usr/doc/faq` which holds answers to frequently asked questions about the most common problems. The directory `/usr/doc/howto` contains "recipes" exlaining how to install certain packages or what to do when you encounter problems.

In `docu` on the first CD there are the latest (by time of press of the CD) versions of FAQ and HOWTO files. It might be of interest to look there! Moreover these files are not compressed so they can be viewed in advance (e. g., with **`type`** under DOS). Under Linux one uses **less**[2] to read text files:

```
newbie@earth:/home/newbie > cd /usr/doc/howto
newbie@earth:/usr/doc/howto > less DOS-to-Linux-HOWTO.gz
```

Kernel specific questions and answers are to be obtained from `/usr/src/linux/Documentation`; this, of course, is only available if you have installed the kernel sources (package `linux` resp. package `lx_suse`), which is highly recommended anyway. Furthermore there are many useful hints in the sources subdirectories (e. g., for the sound driver); and for the brave even the kernel sources themselves.

If you have a question which is not answered in this book, please look at these sources for additional information. The scope of this book is limited. This printed document can be up to date for only a very limited time, because the development of Linux proceeds so fast,

1.4.5 Free Books

package `books`, series `doc` (Documentation) has some books in PostScript format; you can view these books with package `gsview`, series `gra` (Grafic) or package `gv`, series `gra` (Grafic)—if you don't care about trees you can print them as well. Before printing you should question yourself whether it is more economic to buy such a book.

1.5 Laurels

Besides everyone who contributed to the large success of Linux, we would like to especially thank **Florian La Roche** whose experience is of unpredictable value to us. His experience and knowledge helped us building a Linux system in the first place. Thanks to his long experience with his own **jurix** distribution (which was a real "tip" on the internet) he could participate in the creation of S.u.S.E. Linux. Without his help this distribution certainly would not have been possible.

[2] Yes, you are right our `less` is smart enough to handle even compressed files `;-)`

Also many thanks to **Dirk Hohndel** and **Harald König** of the XFree86™team, who have given us valuable tips and advice concerning the X Window System and to **Eberhard Mönkeberg** who was very cooperative as CD-ROM drivers are concerned.

Many thanks go to Hans Lermen, author of the well-known **loadlin.exe**, which allows Linux to be started from DOS, and the DOS **setup.exe** of this Linux system.

Many thanks of course go to the dauntless beta testers who risked their running system: Christian Hüttermann, Dirk Ulbrich, Eberhard Mönkeberg, Frank Hofmann, Harald Wieland, Ingo Schröck, Karlo Gross, Karsten Keil, Martin Schulze, Matthias Urlichs, Norbert Eicker, Ralf Geschke, Stefan Bliesener, Ulrich Goebel, Ulrich Windl und Wolfgang Barth .

Further thanks go to **Stefan Endraß**, who computed the cover picture of the CD.

For the mathematically interested:

The Steiner roman surface

The picture on the titlepage exhibits the so called Steiner roman surface, which is defined by the equation

$$x^2y^2 + x^2z^2 + y^2z^2 - xyz = 0.$$

It is named after the swiss mathematician Jakob Steiner (1796-1863, `http://www.www-groups.dcs.st-and.ac.uk/~history/Mathematicians/Steiner.html`). This surface is an image of the projective plane (which in turn is just a plane with coordinates x and y plus some "points at infinity"). One can see that the Steiner roman surface admits selfintersections along three lines. In fact it has been proven that such selfintersections occur whenever one tries to produce an image of the projective plane inside the threedimensional space with coordinates x, y and z.

More information on the Steiner roman surface ca be found at `http://www.ipfw.edu/math/Coffman/steinersurface.html`.

*

* *

*

The global team of Linux developers is still heavily working on Linux - most of them on a voluntary basis. We want to thank them for their effort - this CD would not exist without them. Our work aims at making their work accessible to a large number of interested users.

Last but not least, special thanks to **Patrick Volkerding** for supporting our work and, of course, many thanks to **Linus Torvalds**!

And Angela, Barbara, Bouchra, Marcus, Marko, Mohammed, Rudolf, Simone, Udo, Ulrich, Virgilio und Wolfgang. And of course: Alan, Alexander, Birgit, Bodo, Burchard, Camelia, Carsten, Christian, Christiane, Dimitrij, Dirk, Doris, Florian, Frank, Gerda, Gerlinde, Helmut, Holger, Hubert, James, Jan, Jürgen, Karl, Klaus B., Klaus F., Klaus W., Lars, Marcus, Margit, Marius, Martin, Melanie, Michael A., Michael B., Michael S., Reinhard

K., Reinhard M., Reinhold, Remo, Richard, Roland, Rolf, Rüdiger B., Rüdiger O., Sabine, Simon, Stefan D., Stefan W., Thomas, Tilman, Uli, Werner, Winfried, and last but not least: Frank Zappa und Pawar!

Fürth, 31st March 1998

Have a lot of fun!

Your S.u.S.E. Team

Chapter 2

Support and services

Certain problems occur over and over again while installing Linux. Many of these experiences made their way into this book.

So if you encounter problems please make sure the solution is not already mentioned in *this* book or in our support data base (section 2.2, page 15) before contacting our support team. Furthermore you will find a lots of answers in the `README` files on the first CD.

Of course it is not possible to cover all possibilities in this book. Adding every known problem to the book is not economical and makes the book unreadable. There are always new questions and problems anyway.

Should you however be confronted with a situation in which all documentations including the manual and the support database can not help you out, please feel free to to contact the free Installation support. The installation supportshould be able to solve you problems with a high probability. Our support team is made up out of specialist, which are able to handle even the most complicated problems.

We hope to have straightened out most installational problems. So if you either do not find your way or you have a "tricky" question in Linux, do not hesitate to contact us. Our technical support team is there to help you `:-)`

S.u.S.E. offers two kinds of support:

- 60 days of *installation support* which is included in the price of this S.u.S.E. Linux, and
- *Business support*, commercial support which extends beyond installation support.

Please call us or see our web site for details Apart from this we offer free of charge information on Linux and information about our products online via the internet.

These offers will be covered in subsequent sections.

2.1 Installation support

2.1.1 Registration

Requirements

We would like to help you fast and unbureacratic within the free 60-day intallation support service.

Please assist us – and yourself – in providing a fast and relyfull service by including your registration code and your full name in your request.

In case you have not registered yet, it can also be done via our support representative. Additional to the registration code and your full name we need your complete address. Please also submit your fone, fax and electronic mail address.

We prefer to assist you in questions concerning technical problems rather than in spending time to loacte your registration data.

The submiting of these datas in every support request (independent from the medium used) is the base for handling the request.

In following we will shortly describe the ways on how to register as a S.u.S.E. Linux user.

Registration code There is a registration card that came with your S.u.S.E. Linux. There is a registration code printed on this card; this code is unique and serves for verifying that you purchased a genuine S.u.S.E. Linux. This code matches the code printed on the first page of you reference manual.

Registration card Online If you send in your registration card or register via our WWW–server `http://www.suse.com/Customer/index.html` you will become a registered S.u.S.E. Linux user. Then you are entitled to get installation support. Registration is obligatory if you want to receive support.

Since this code is product-based we request you to register every S.u.S.E. Linux products you purchase, even if it is an update or if you purchased S.u.S.E. Linux directly.

Customers number After having sent in your registration card you will be assigned a customer number. We will not send you this number unless you ask (e. g.) for support. If you receive our leaflets there is the customer number printed next to your address. If you order directly from S.u.S.E., LLC the number is printed on the bill.

We are obliged to help you fast and in an unburocratical manner. Please help us by filling in any “formalities”, these are your customers number and your registration code (electronic mail in subject). If you do not know your customers number we still ask you to send in your registration code and your complete name and address (where you are registered, e. g., the company’s name). Our employee will tell you your new customer number.

2.1.2 Scope of our installation support

So, what is this “installation support”? We want to outline what the item installation support stands for.

The installation support is for getting your S.u.S.E. Linux up and running. Our definition of installation support includes the following items:

- Installing the S.u.S.E. Linux system from CD
- Basic configuration of a standalone machine
- Basic configuration of the X11 GUI
- Connecting your standalone machine to a mini-network

- Basic installation of network services, electronic mail and printers for one machine.

We gladly help you to connect your machine to your internet provider. Please mae sure however that before contacting us that the problems that occur don't refer to the provider itself.

Please understand that the following topics are not included in our installation support:

- Configuration of large or commercially used networks
- Configuring servers (i. e. mail-server, news-server, NFS-server, Samba-server, ISDN-server, Modem-server etc.);
- Support of DOS, Windows (3.1, 3.11, '95, NT), OS/2, SCO and any other system.
- Support for commercial Linux programs like **Applix*ware***, Adabas D, Maple, Netscape, etc., also in case if they' re distributed by us.
- Support for other Linux distributions (Caldera, Debian, DLD, Red Hat, etc.);
- Error analysis or support for homemade or self-compiled programs.

Our installation support should be regarded as help for installing Linux, it is not a complete lecture or a course on Linux. So please contact it only in case of installational purposes and not for background information. For more detailed or complex requests, please feel free to contact our business-support service.

Generally speaking our support team takes any effort to help you. We assume that you have exploited all information you could gather (this manual and the on-line help on a running system).

However we can not exclude the case the S.u.S.E. Linux does not run at all or not completely on a given hardware platform. We ask for your comprehension that we can not give you a 100 % guarantee for success concerning our installation support.

Support for software packages

Besides the basic Linux kernel system we offer a wider variety (at the moment app. 1200) software packages on our S.u.S.E. Linux CD. We try to keep these packages up to date. Anyway it is not possible for us to know each and every package's documentation by heart nor to programm additional features. Please understand if we refer you to the supplied information (manpages, `README` files and on-line help).

Linux Snapshot!, Linux Decathlon, Internet

Besides all this installation support is only available for tools and packages that came along with the S.u.S.E. Linux. Excluded are e. g. questions on other packages or on "hacker kernels". Also excluded are packages that can be found on the internet or on our other products **Linux Snapshot!** or **Linux Decathlon**.

Feedback

We always appreciate tips, hints and problem descriptions. We will help you if the encountered problem is a basic problem or if we already have some help at hand. Anyway your solution provides us with useful information for avoiding this problem in our next release, thus helping other S.u.S.E. Linux customers via our WWW-server or the SupportDataBase. On the other hand we try to accustom our S.u.S.E. Linux product to the wishes and ideas of our customers. Therefore we highly appreciate any criticism on our CD or on the manual. In our opinion this is the best way to rectify vital errors and to keep up to our high quality standard.

Feedback can be sent any time to `feedback@suse.de` via electronic mail; alternatively you can send us a letter or a fax.

Period of time for installation support

The period of time for installation support is restricted to 60 days from the day of purchase; but no longer than 60 days after the follow-up release has been released.

2.1.3 What we need to know for helping you

For an effective and fast support we need to know a lot of your machine, the software running and the problem that ocurred. Further indices as error messages and configuration files will help too.

Before you contact via fone, electronic mail, fax or snail mail the S.u.S.E.support, please check the following list. The more information we get in advance, the faster and effective we can help you. In lots of cases, the error can be be solved by yourself, if you try to describe it as explicit as possible for the support.

> There are many solutions to well-known problems already avaiable. These are presented in our SupportDataBase at `http://www.suse.de/Support/sdb\_e/index.html`. We recommend having a look here first!
>
> Besides the on-line SupportDataBase the complete content of the SupportDataBase can also be found in package `sdb`, series `doc` on your S.u.S.E. Linux CD. Those articles can be read using any WWW browser (**arena**, **lynx**, **netscape**).

Now we list what we need to know from your or from your system respectively:

Registration
1. Always hand in your registration data as described in section 2.1.1

Describe
2. Please describe exactly
   - *what* happened (exact error messages, screen phenomena, disk-noises etc.); please submit us the original message not an interpretation!
   - *when* did it happen (while booting from disk, while accessing the CD-ROM, while moving the mouse etc.)

- can it be reproduced?
- what you have done to solve the problem

3. Try to remember whether you have changed anything (even if it might seem irrelevant to you) *before* in the system configuration before the error ocurred. This might have happened the day before or before booting for the previous time.

4. **Hardware** Many errors and phenomena that occur are due to faulty or incorrectly working hardware. Most often there is no visible connection between the error and the hardware components. So, for being sure we need detailed background information of your system. Please check the list below and hand us as many pieces of information as possible.

   - Processor: manufacturer, type, frequency, mask stepping.
   - Motherboard: manufacturer, name, BIOS version, onboard controller.
   - Memory: RAM-size, number of modules, type of RAM (FP, EDO, BEDO, SDRAM), access time (60 ns, 70 ns), cache-size, type of cache.
   - Hard disk controller: SCSI, IDE, EIDE, onboard or module, name, BIOS-version, chipset.
   - Hard disks: Type, manufacturer, name, size, bussystem (SCSI, IDE, EIDE, SSA), partitoning.
   - CD-ROM drive: Type, manufacturer, name, size, speed, internal or external, SCSI EIDE (ATAPI), parallel port.
   - Other media: CD-writer, streamer (type, manufacturer, SCSI or EIDE, floppy-streamer, capacity), ZIP-drives, Jazz-drives, MO-drives, flopticals.
   - Graphics device: Type, manufacturer, chipset, bus (PCI, VL-bus, ISA), memory size. memory type (DRAM, VRAM, WRAM, SGRAM, EDO).
   - LAN card: type, manufacturer, chipset, bus (PCI, VLBus, ISA), LAN type (Ethernet, Token Ring etc.), Connection type (Coax-Cable, Twisted-Pair), portaddress, irq.
   - Other peripheral devices: Printer, parallel port, serial port, modem, ISDN card, scanner, soundcard.
   -

5. **XFree86™** If you encounter difficulties while installing the **XFree86™** system the following information might come in handy:

   - Graphics device: Type, manufacturer, chipset, bus (PCI, VL-bus, ISA), memory size, memory type (DRAM, VRAM, WRAM, SGRAM, EDO).
   - Monitor: Type, manufacturer, max. horizontal frequency, max. vertical frequency, band with.
   - Mouse: Type, manufacturer, connection (serial, PS/2-Port), date of purchase
   - Installed X-server: Package name, if there are multiple servers installed please tell us the order of installation.

- How did you configure X? (e. g. using **XF86Setup** or **xf86config**).
- The file `\tld/.X.err` which is to be found in the home directory of the user who invoked a **startx**.
- If you are running **xdm** (runlevel 3):
  - `/var/lib/xdm/xdm-errors`: error messages of the X-server
  - `\tld/.xsession-errors`

Important files and screen messages

The information just mentioned can be gathered by consulting the manual or the following files or screen messages:

- loadable modules

  `/etc/conf.modules`
  `/var/log/messages`

- Kernel

  `/usr/src/linux/.config`

- Boot messages

  `/var/log/messages`
  `/var/log/boot.msg`

> The last lines of `/var/log/messages` are sufficient. Please do not send us the complete `/var/log/messages`!

Screen messages:

- PCI devices: **cat /proc/pci**
- Interrupts: **cat /proc/interrupts**
- Port addresses: **cat /proc/ioports**
- Memory: **cat /proc/meminfo**
- managed devices: **cat /proc/devices**
- Networking devices: **cat /proc/net/dev**
- SCSI devices: **cat /proc/scsi/scsi**
- SCSI controller: **cat /proc/scsi/<controller>/0**

These can be redirected into a file. Example:

```
root@earth: # cat /proc/pci > /tmp/pcidevices
```

Now you can send this file `/tmp/pcidevices` to our support team.

> The "files" in `/proc` can only be viewed using **cat**, **less**, or **more**, but can not be edited! Well, nowadays even this is possible, but do not rely on it!

If you do not manage to get a Linux system up and running you can gather the information mentioned above via **linuxrc**. Please refer to section 3.15, page 32.

2.1.4 How to contact the S.u.S.E. support team

You can reach our support team via electronic mail, fax, regular mail und telephone.

Contacting S.u.S.E. via the fone is on the first look the fastest way. The spelling of error messages from your side and the spelling of command entries or file names from ourside on the fone is in the majority of cases quite difficult and boring `;-)`

Relating to our experience, it is in most cases the fastest way to describe your request in a fax, electronic mailor letter. On the one hand you've the chance to formulate your request more precisely, and on the other our employee can tell in most cases on the first sight where the problem is located.

In addition, we handle requests via electronic mail, fax or letter 7 days a week, while the hotline can only be reached 2 times a week.

For the fast and easy composition of a support request, we supply you with an electronic support request questionaire on 2 different locations:

- on the first CD of the installation set
- on the installed system

The easiest is to send us an electronic mail.

You can reach our support time via:

- **via electronic mail**
  Address: `support@suse.com`
  Address (Europe): `support@suse.de`
- **via fax**
  Fax: `+1-510-835-7875`
  Fax (Europe): `+49-911-741\,77\,55`
- **via fon**
  Fon: `+1-510-835-7873`
  Mo & Th 11:00am – 5pm (PST)

2.2 Further services

Additionally we would like to draw your attention to our free of charge services that are available round the clock:

- **S.u.S.E.'s web site**
  `http://www.suse.com`
- **S.u.S.E. mailinglists** (information and discussions via electronic mail):
  - `suse-announce-e@suse.de` – announcements concerning S.u.S.E. Linux (english)
  - `suse-announce@suse.de` – announcements concerning S.u.S.E. Linux (german)
  - `suse-linux-e@suse.de` – all about S.u.S.E. Linux (english)
  - `suse-linux@suse.de` – all about S.u.S.E. Linux (german)

  - `suse-isdn@suse.de` – ISDN and S.u.S.E. Linux (mainly in german)
  - `suse-applix@suse.de` – the **Applix*ware*** office suite (mainly in german)
  - `suse-adabas@suse.de` – about Adabas D (mainly in german)
  - `suse-informix@suse.de` – about Informix (mainly in german)

  Send an electronic mail message to `majordomo@suse.de` with **`subscribe`** in the body: **`subscribe`** <**`liste`**>, the subject does not count; example:

  ```
  subscribe suse-announce-e
  ```

  to receive all announcements. To unsubscribe a list write again an electronic mail message to `majordomo@suse.de`; this time with the body:

  ```
  unsubscribe suse-announce-e
  ```

  Please note, that **`unsubscribe`** has to be done from the same electronic mail account.

- **S.u.S.E.'s ftp site**

  ftp.suse.com

  Latest information, updates and bug fixes

  Login as '`ftp`'.

Chapter 3
Installation

3.1 Don't panic! You can do it!

It is not easy to find a good middle ground for an installation manual. If you go into any gory details or if you describe problems and their solutions there is a risk of scaring neophytes away. On the other hand, if you leave out all the details, experienced Linux freaks will lay the book aside. If you sacrifice these details for clarity, the manual might not be useful in case you `do` encounter any problems.

That is why we decided to split this manual into two parts:

- A quick-install section that should work for most users (section 3.2).
- A detailed installation manual where we cover background and pitfalls. But, as we go into detail, this section obviously will get quite voluminous (section 3.3).

In case of doubt try the quick install; this will work in almost any case. If this fails you should try the detailed installation.

3.2 Linux in 30 minutes – the quick install

There are several prerequisites to doing a quick install. First you have to make sure that you have either extra space on your hard disk for Linux, a free partition you might want to use, or another operating system you want to remove. If this is not the case you should prepare your hard disk in advance. **fips** on our first CD might come in handy; additional tips can be found at section S1, page 21 in section 3.28, page 60 and section 3.29, page 61.

Then you perform the following steps. The output of the several steps is displayed on screen. Additionally you may scroll through the help texts.

- Insert the S.u.S.E. boot disk included with this product and boot your machine. If the greeting appears just press ⏎ . If you have a suitable BIOS you can even boot from CD.
- **linuxrc** starts. Select language, screen and keyboard.
- Load necessary hard disk kernel modules (You *do not* need modules if you *only* have (E)IDE disks and/or CD-ROM drives installed!) and for the installation media (via CD-ROM, network card or PCMCIA). This either works automatically or has to be done manually. For some modules you can add parameters. Information on parameters can be found in section 7.3.2, page 102

- Select `'Start installation'` for invoking YaST. The source media normally is either `'CD-ROM'` or `'Network'`.
- YaST starts and requests a disk. Since we have not created one yet, we answer `'No'`. – Online help is always available via F1 .
- Now you assign your Linux partition. To do so move to `'Settings for installation'` and `'Partition hard disk'`. Do not forget to assign a swap partition. The type of the swap partition has to be set manually! Additional information on the philosophy of partitioning is found in section 3.28, page 60 and section 3.29, page 61
- Now you have to assign mountpoints with `'Settings for installation'` and `'Assign partitions'`.
- Now leave YaST. Save the data on a "save disk" (not the S.u.S.E. boot disk!)[1]. Now answer `'yes'` and the system will reboot.
- Do not forget to re-insert the S.u.S.E. boot disk!
- Press ↵ at **linuxrc**'s greeting. Now again you have to choose language, screen and keyboard. Load the necessary kernel modules and select `'Start installation'` for invoking YaST as you have already done before. Nothing new.
- Now you can answer `'Yes'` if asked for a disk—if you have saved the data on the disk successfully. Insert "save-disk" created earlier. YaST now formats your Linux partitions.
- Confirm the swap partition and allow it to be initialized
- Via `'Start/Setup installation'` `'Change/create configuration'` you select which packages you want to install and start the installation procedure afterwards with `'Start installation'`. The selected packages are installed (first only those that are on CD 1).
- Now quit YaST pressing `'Finish up installation'` and choose the kernel you want to install for future use.
- Do create a boot disk for booting Linux. This might not only be handy if there is an emergency. Here you will need to select your CD-ROM once again.
- You can now install LILO, the bootmanager.
- Now you give a name to your machine and select the network you have. Afterwards the machine boots again and you can log in to your freshly created Linux.
- Now YaST starts again and all packages that were not on CD 1 are installed. Now you configure all those tiny details of your system.
- Now you can login as user `'root'`. You can start YaST and administer your system with `'Administrating your system'`. You are done!

3.3 This installation manual

There are many roads leading to a correct and stress-free Linux installation. But some of them are more complicated than others. We at S.u.S.E. have tried to determine how less-experienced Linux users can be supported better than before.

[1] Some times saving on a disk doesn't work; in this case just skip this step.

It's quite difficult for the beginner to have an overview and of course to always choose the correct alternative. Therefore we separated the installation part from the rest of our book and tried to focus on background and coherence. Moreover, since we excluded everything about system management and the like this material becomes more compact. If you are an advanced Linux user you might consider some explanations as useless or too long. Please bring back to mind your first computer, happily taking advantage of any information you could get. This installation guide is not considered as an instruction for the S.u.S.E. tool YaST nor for other programs and does not describe Linux completely. Only those things considered helpful for the installation and understanding are included here.

3.3.1 Finding your way through this installation guide:

The chart on page 20 shows you the myriad of possible installation options. You should start at the top and work through right to the bottom, until you (hopefully successful) reach the end of the installation. Of course you should not skip steps in this plan. Steps marked as `D1` to `Dx` are designate decisions, whereas `S1` to `Sx` are working steps.

All decisions and working steps will be thoroughly explained later on in this installation with all necessary background information. Please read this carefully and follow the instructions. If you encounter any problems you might find the links to problem descriptions which are mentioned together with the working steps, of some use. Problem descriptions are named `P1` through `Px` which we describe in a separate chapter.

Concentrated information on a certain point you can find in info blocks `I1` to `Ix`. You'll find links to the Infoblocks in various places; this makes life easier as it reduces the amount of redundant data. Furthermore there are links, if necessary, to other chapters in this book.

If you have mastered your installation you should have a running base Linux on your system. If this fails for any reason you should contact S.u.S.E. technical support. You'll find our contact information in chapter 2, page 9.

3.4 Basic principles of Linux installation

We are going to describe the *real* Linux installation which requires you to create one (or more) partitions and a separate filesystem; the *demo mode* which is based on MS-DOS filesystems and provides less performance and features is not mentioned here.

Linux installation mainly consists of the necessary preparation and three steps:

- create space (partitions) for your Linux on your hard disk.
- start an independent ☞*Base-Linux*. Here you have three different possibilities:
  - You boot with the supplied S.u.S.E. boot disk (for really obscure hardware you might have to create your own own boot-disk) and tell the

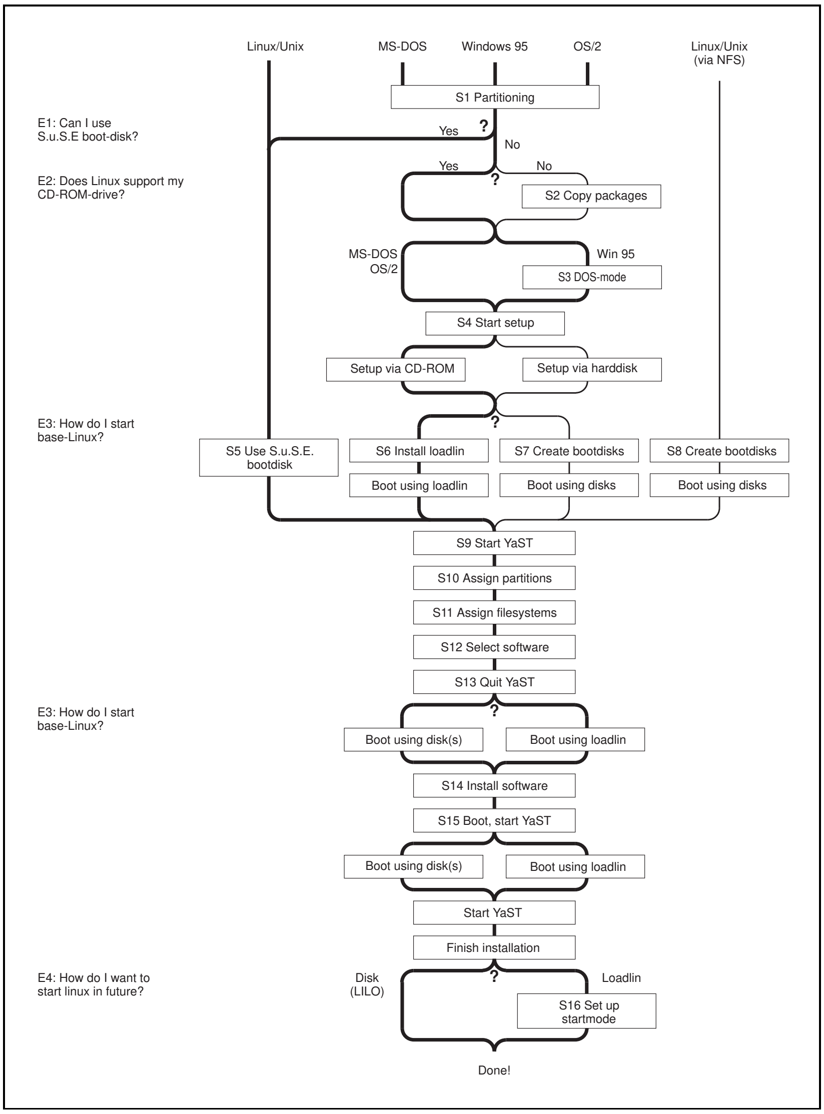


Figure 3.1: Scheme of installation

kernel to load the rest of the installation system from CD. If you have suitable hardware and BIOS, you can even boot from CD.

– You prepare an installation disk; boot with the supplied S.u.S.E. boot-disk (for really obscure hardware you might have to create your own boot-disk) and the prepared install-disk containing the root-image and the installation tools is loaded into RAM disk.

– The Linux kernel is loaded directly from CD (under DOS) via **loadlin**; the root image and the installation tools are loaded from CD or HD into a RAM-disk.

- Now you can log in to your ☞*Base-linux*. You now load the necessary drivers using **linuxrc**. With YaST you customize your system, partition and format your HDs and install the base packages for your Linux system.

- Now YaST creates partitions and filesystems, the selected packages are installed from CD or from an HPFS-partition. Now a *real* Linux is running on your machine.

Now you have a real Linux running; but for you becoming the system administrator there are still a lot of things that need to be done. The X-Window system, networking, accessing online services, compiling of your own kernel and much more . . .
These items are far beyond the aim of a basic installation and are therefore covered in separate chapters.

3.5 S1: Partitioning

What's it about?

This step prepares your HD for Linux partitions.

Requirements

You do not want a demo installation (you would not need a new partition for demo mode). You have some HD space, time, S.u.S.E. Linux, perhaps some disks or tapes for backups and a boot disk of your former OS.

Summary

By partitioning, your HD can be separated into several independent parts. One reason for partitioning your HD might be the coexistence of several different operating systems with different file systems on your HD .

HDs are divided into cylinders, one cylinder always contains exactly the same amount of tracks, cylinder 0 being the innermost of them.

How do you get space for your Linux partition?

- Single partitions can be deleted but, of course, all data which resides on these partitions will be lost. This space is now available for new partitions (e. g. Linux). Of course you can also split this partitions into two or more partitions.
- Single partitions can be assigned to a different file system. As with deleting, all data on these partitions will be lost.
- If you are using MS-DOS or Windows you can minimize the last partition without losing your data. If all data resides on low-numbered cylinders, special programs (e. g. **fips**) can decrease the upper cylinder limit to reduce the size of your partition leading to the possibility of creating a new partition for Linux. Using a defragmentation program ensures you that all data is moved to the beginning of the partition.
- The easiest method of course is buying a new HD. This of course will cost you some money.

Every change of partition tables must be done with extreme caution! Please read the documentation of the corresponding programs! Severe problems can occur while handling partition tables; you can even lose all your data. We at S.u.S.E. cannot be held responsible for this! It is strongly recommended to have at least a boot disk and a backup of your important data.

Step by step

This is how to proceed to partition your HD:

1. If you don't know already, you have to find out how many partitions your HD consists of and, of course, the size of the partitions. Use **fdisk** of your OS for this task.
2. Check how many partitions you really need and which size you're going to assign to them. If you are unsure about the correct size we recommend 3 partitions (`/` (root), `/home`, and `swap`). The root partition should be at least 50 MB (depending on how many packages you are going to install) up to 200 MB (if you're planning a bit more). The size of your /home partitions depends on the number of users you are going to "install". Swap partition should be about 20 MB of space.
3. Write down the partition data; you're going to need it later on during the installation process.
4. Now a backup of your HD is highly recommended! If you don't have a tape or streamer installed and don't want to backup everything on floppies (which is rather tedious work) at least make a backup of the most important data and files (e. g. `autoexec.bat`, `config.sys`, `*.ini` etc.) Create a boot disk for your OS **and** check whether you can boot with it! There are several programs that might come in handy on your boot disk; those are programs as fdisk, an editor, a format program and of course your backup program.
5. The next step depends on your system configuration

   DOS/Windows 95, one partition on your HD, and no backup

   You have to reduce the size of your partition without deleting data. Move all data to the beginning of your partition using a tool such as **defrag** (DOS/Windows). Defragmentation programs normally don't move hidden files. There might be such hidden files on your HD (e.g. copy protection). You must ensure that those files don't have to be at a certain location on your HD. If you are sure that there aren't any of those hidden files you can change the attributes of those files from `hidden` to `not hidden`. Some defragmentation programs offer the option of deciding whether you want to set this flag or not. Please keep in mind that even the windows-swap file is a hidden file. If it's in your way you have to unselect it under "disable virtual memory".

   Now that you have accomplished the first task of gaining enough space you should move to the directory `\dosutils\fips.exe`. There you will find **fips.exe**. Please study the included documentation about fips carefully since handling partition tables can be rather tricky! The program

fips.exe only runs on DOS, not on Windows; you have to exit windows or (if using Windows 95) bring it into DOS mode (see section S3, page 27). After having run **fips.exe** you will have a second partition which later on will be divided into your Linux partitions.

DOS/Windows 95 and several partitions or you have a complete backup

You can delete your DOS partitions and afterwards install them reduced in size again. Here you are going to lose all data on those partitions. You have to have a complete backup at hand (file backup, no image backup!). Using **fdisk** you delete all partitions and install the new ones. Next you format those partitions, install your OS and copy all data from your backup back to disk (remember that you have your backup program on this boot disk).

OS/2

Here you have the following possibilities:

- You reduce your OS/2 partitions. Therefore you must backup all data on your OS/2 partition, delete those partitions using OS/2's **fdisk** and reinstall them reduced in size. The space now available will be used further on for your Linux partitions.
- You plan to use Linux instead of OS/2. Then you have to do nothing but change the partition type later on to Linux. Or you delete all OS/2 partitions and reinstall them later as Linux partitions.
- You buy a new HD. Formatting and partitioning it will be done later on with YaST.

UNIX/Linux

You already have suitable partitions, meaning that you don't need to bother with repartitioning your disks. Setting the partition table will be done by YaST.

6. Reboot your computer.
7. Check whether your old system is still running! Linux partitions on the freshly created new space will be installed later on using YaST.

Problems

The following problems can occur:

- Partition might not be adequately reduced due to some un removable, see files section P1, page 53.
- Under DOS/Windows your CD-ROM had another HD Identifier; under Windows the computer hangs, see section P13, page 56.

3.6 E1: Can I use the S.u.S.E.- boot-disk?

What's it about

Are you able to use the supplied S.u.S.E.-boot-disk? This mainly depends on your system. The alternative is booting via a "hand-made" bootdisk or via DOS and **loadlin**. You'll find an overview of all schemes on page 20.

Summary

The supplied S.u.S.E.bootdisk covers most existing hardware. All necessary drivers are loaded as modules via a menu. Thus making the S.u.S.E. boot disk a rather flexible installation tool. You cannot use the S.u.S.E. boot disk if your system is supplied with exotic hardware or your CD-ROM is not supported by Linux (here you need a second disk or you will have to copy files to disk. For these possibilities an alternate way is described in the diagram on page 20).

If you're working with DOS or Windows 95 and plan to start Linux via **loadlin** (a rather convenient way) it is recommended *not* to use S.u.S.E. boot disk but to work with **loadlin** instead. This saves you the installation of **loadlin** and the selection of a suitable kernel.

Recommendation

Use S.u.S.E. boot disk if:

- you don't have exotic hardware installed (in case of doubt, just try anyway).
- you are using Linux, Unix, Windows NT, or OS/2.
- for any reason you don't want to use **loadlin** (although this is a recommended method).

3.7 E2: Does Linux support my CD-ROM?

What's it about?

Whether your CD-ROM is supported by Linux is important for the installation. If it is not, you will have to copy all necessary data for your Linux installation onto your HD.

Summary

Linux supports almost every CD-ROM. ATAPI and SCSI CD-ROMs, and even exotic CD-ROMs will work. For further information please see Infoblock section IB3, page 57.

Recommendation

You can assume that Linux supports your CD-ROM unless you definitely know the opposite. If you encounter problems, the most you have to do is to start all over again and choose a different installation procedure.

3.8 S2: Copying packages onto the HD in the case CD-ROM drive not supported

What's it about

In the case that our standard kernels do not support your CD-ROM drive or your CD-ROM drive is not supported by Linux at all, you have to copy Linux packages onto your HD using your current non-Linux OS. This will allow you to install Linux directly from your HD, circumventing the need for the CD-ROM drive.

Recommendation

You are using DOS, Windows or OS/2 and know (perhaps from an failed installation) that Linux does not support your CD-ROM. Moreover you have plenty of space on your DOS, OS/2, or Windows partition.

Furthermore you should have had a look in package descriptions which packages you would like to install. Otherwise we will just install the base system.

Summary

There are two places where you can replace CD-ROM access by DOS-file-access.

- If your ☞*base-Linux* is loaded using **loadlin** the system gathers the contents of the root-disk, which is loaded directly into RAM disk, and the contents of install-disk from CD. If your CD-ROM is not supported by the standard kernels on CD offered by **setup**, data can easily be loaded from a DOS or HPFS file system. Naturally those files have to be installed there first.
- If your base Linux is already running and you want to install the chosen packages using YaST this is normally done via CD. If Linux does not support your CD-ROM you have to copy the files using DOS, Windows, or OS/2 onto your HD. YaST offers you the possibility to choose a directory to install from. You might you have to install the system step by step if there is not enough space to install it all at once.

After having a base system running you might find a kernel (or generate your own) which recognizes and supports your CD-ROM. On CD you'll find three types of files related to kernels. Files without extensions are considered kernels only for installation purposes. These kernels only support hardware that is needed for installation (no streamers or mice). Files with `.ikr` extension are considered normal installable kernels. There is one more file ending in `.inf`. This obviously is a text file containing a number which YaST uses to present information about a given kernel. You should always copy a suitable kernel even if you plan to start via loadlin or floppy. YaST will create a boot disk using this kernel at the end of the installation procedure.

Step by step

Here's how to copy the chosen packages onto your HD.

1. First create a directory you want to copy the files to. The name of this directory is unimportant, as long as you don't forget it. In our example we will refer to it as `\emil`. Additionally you have to create a subdirectory `\emil\suse`
2. Under the `\emil\suse` tree you will need at least three directories. Let's call them `a1`, `images` and `setup`. These are necessary for the base installation.Therefore you should create these directories. Diagram figure 4.3, page 69 shows the complete file tree needed in an overview.
3. Copy all files from `\suse\a1` of CD 1 into `\emil\suse\a1` on your HD.
4. Under `\suse\images` select a kernel which supports your hardware. More detailed information about which kernel supports which hardware can be found in `\suse\images\readme.dos`.
   Copy this kernel to `\emil\suse\images`. A kernel consists of three files, one without an extension and those with `.ikr` and `inf` extensions. If space on your HD is not an issue you can of course copy all kernels into this directory. You can then choose your kernel later on. More information about this can be found under Infoblock section 3.27.1, page 56.
5. Copy `\suse\images\root`
   `\suse\images\initdisk.gz` and
   `\suse\images\TRANS.TBL` to
   `\emil\suse\images`
6. Copy `\suse\setup\loadlin.exe` to `\emil\suse\setup`
7. Furthermore copy file `\suse\setup\inst-img` to `\emil\suse\setup`. This file is relatively big, but it is only needed while doing the first installation. If the base system is up and running, you will always be able to install additional packages from a DOS partition; then the file `inst-img` could be removed.
8. Create the directory `\emil\suse\setup\descr` and copy all files from `\suse\setup\descr` into it.
9. If there is some space left on your HD you can copy `\suse\setup\du` to `\emil\suse\setup\du`, which, of course, you have to create in advance. These files are not really necessary but will help you later on for showing you how much space there is left and how much is already occupied. If you have enough space you might find that this comes in handy.
10. Now you have everything that is absolutely necessary for installing Linux on your HD. All other software is still missing. Since your CD-ROM is not supported by Linux you have to install everything step by step, meaning that you have to copy everything you want onto HD and install it from there using YaST. You don't have to do that right now, but if you already know what you're going to use, you can start now: Just create the corresponding directory under `\emil\suse` and copy everything from CD (meaning the corresponding directory) into this newly created directory. You can find all packages (their contents) in the online documentation or in the package descriptions.

Problems

There should not be any problems so far. If there is not much free space on your DOS, HPFS file system please keep in mind that you have to have root to install the above mentioned base system, otherwise you have to say goodbye (just for some time) to your DOS/Windows/OS/2 software.

3.9 S3: Bringing Windows 95 into DOS mode

What's it about?

You have to bring your computer into real mode (under DOS) to proceed with installation.

Requirements

You work under Windows 95 and have completed the preparations. Installation of Linux can commence.

Summary

The installation program **setup.exe** is an MS-DOS program which is only able to load the Linux kernel of your base Linux into memory if either the CPU runs in real mode or a VCPI server[2] is active. The DOS box of Windows 95 runs in virtual 8086 mode but does not offer an VCPI server. This is why setup does not work here.

Step by step

There are two alternatives: Switching to DOS mode from Windows 95 or booting your computer and selecting command line input (DOS). If Windows 95 is already up and running click on `'Start'-'Shut down'-'Restart the computer in MS-DOS mode'`.

Or you can just press F8 while booting and choose `'command line input'`.

Problems

There should not be any problems so far.

3.10 S4: Invoking setup and first steps with setup

What's it about

Setup.exe starts your ☞*base Linux* which enables you to proceed with real Linux installation.

[2] e. g.provided by **emm386.exe**.

Requirements

You have started MS-DOS or an MS-DOS box (not in protected mode), CD 1 is in your CD-ROM drive and you can access it.

Summary

In this step you will continue until you reach the point where you have to decide between one of two alternatives, namely booting with boot disks or with **loadlin** from CD.

Step by step

Here's how to do your first steps with setup:

1. Start **`setup.exe`** in the root directory of your CD.
2. Choose your favourite language; for `'English'` installation just press ↵ .
3. Enter your CD-ROM drive (e. g.`E:` on DOS). Possibly this has changed due to an additional DOS partition.
4. **Setup.exe** welcomes you; we reply to such a nice gesture by pressing ↵ .
5. The following sections will explain the booting of the base-Linux. section E3, page 28 (below) will help you decide what booting method to choose, and subsequent sections will guide you through the respective process for each method.

Problems

There should not be any problems so far.

3.11 E3: Which way do I want to boot base-Linux?

What is it about?

There are two possibilities to boot your base Linux: either from disks or directly from CD-ROM by means of **loadlin**. Now you will choose the best option.

Summary

Certainly, the easiest way is to start your base Linux directly from CD using **loadlin.exe**. It runs on DOS and loads a kernel image from CD into RAM and starts executing kernel code. Some things must be fixed for this to work properly though. The computer has to run in real mode or in virtual VCPI mode[3]. OS/2 DOS box or Windows NT doesn't work.

Booting via disks is almost always possible but is a little more tricky and requires an extra amount of time. We are talking about the disks created by **setup**. The easiest way is of course to use the S.u.S.E. boot disk (or booting from CD). We will explain these options later, but for let's stick to the decision about loading via **setup** or via **loadlin**.

[3] as provided by, e. g. emm386.exe.

Recommendation

Take the “*loadlin way*” whenever possible as long as your CD-ROM is supported and you are not working under OS/2 or Windows NT. When in doubt, just try it. If this fails you have to start over again at section S4, page 27 and choose the *“floppy-way”*. If you’re installing from your HD, you must chose to boot via the S.u.S.E. boot disk or the disks you’ll create with **setup**.

3.12 S5: Creating boot-disks with setup

What it’s about

Here you create one or two disks for loading your Linux kernel and start up your ☞*base Linux*.

Requirements

You need one or two 3.5”-HD-disks and of course a bootable 3.5” floppy drive. If there is an additional 3.5”-floppy drive installed that functions as boot drive, see section P12, page 55). If you are working with Windows 95 you have not launched **setup** from within a DOS box but in MS-DOS mode instead.

Summary

For booting the base Linux via disks you need at least a disk with a kernel. The S.u.S.E. Linux CD contains several disk images. Such an image can be copied onto disk with **setup** and is thereafter called a boot disk. Included in those images are the linux loader (LILO) and the program **linuxrc**. It is possible to choose a kernel at boot time and to add parameters as needed. By the way, LILO need not be installed on disk but should be your choice if you plan to install more than one operating system and want to select at boot time which to boot. Installation of LILO is not the subject of this chapter but is covered in chapter 8, page 121 in great detail.

linuxrc is your assistant for loading kernel modules adapted to your hardware and to start the installation.

Even the included S.u.S.E. boot disk can be used as boot disk in a case of emergency. You have to create your own boot disk only if you have exotic hardware installed that is not supported by the modular kernel on this disk as explained below

In the latter case selection of a proper kernel is important. It has to support your hardware. There is more information on kernels in section 3.27.1, page 56.

Then you boot either with disks only or with 1 disk and a CD.

Also note the following regarding boot disks on CD. These kernels are not made for being installed permanently on your system. They cover a wide range of hardware and make installation as simple as possible. The kernel should not be too big since there has to be some space left in the RAM disk. If your Linux system is up and running it is recommended you build your own kernel which perfectly suits your system. Thus you gain optimal performance and a minimum amount of required memory. How to build a kernel is described in chapter 13, page 181.

Step by step

Here's how to create a set of disks:

1. In 3.10 you started **setup.exe** and now you're to the point where you must decide to install via **loadlin** or floppy.
2. Select `'floppy'` and press ↵
3. Now you have to select a suitable kernel that supports your SCSI adapter if you have one. **setup** shows the vital part of kernel messages. If you need further information you can look it up in `\disks\readme.dos` or in section 3.27.1, page 56. Remember your kernel's name. You will need it later on. Now press ↵ . If you find out that the chosen kernel was not the best choice you can come back here and select another one.
4. Now you're ready to create the boot disks. Insert the first (DOS formatted) disk into the 3.5" drive and select which disk to create.
   - Start with the boot disk (only recommended, not necessary). Move the cursor onto `'Boot'` and press ↵ .
   - **setup** requests you to confirm that you have inserted a disk. Press ↵ and the disk is written.
   - When this is done press ↵ .
   - Now select `'Done'` to exit this screen and **setup**.

3.13 S6: Installing loadlin and starting your base-Linux

What's it about

This step shows you how to install a Linux kernel, to load a base Linux using **loadlin.exe** and bring up your ☞*base Linux*

Requirements

If you have made it this far and you are ready for your first Linux prompt everything is prepared.

Summary

Which kernel to choose mainly depends on your CD-ROM drive or your HD (meaning the controller). Later on you can easily choose another kernel or, which is recommended, build your own kernel which exactly fits all your needs. More information about this topic is to be found under section 3.27.1, page 56.

Setup now creates a directory `\loadlin` on your DOS partition and copies **setup.exe**, **loadlin.exe**, **linux.bat** and the selected kernel (renaming it to **zImage**). After having finished the installation you can start your Linux by invoking **linux.bat** (all you have to do is to supply the command with the root partition of your Linux; this will be thoroughly explained in section S11, page 37). After this step has been completed the kernel will be loaded and started.

Step by step

Here's how to install **loadlin** and ☞*base Linux*.

1. Select option `'loadlin'` and press ⏎ .
2. Now you'll be presented with the amount of RAM installed on your computer. In most cases this will be correct and you confirm by pressing ⏎ . However, if it does not match you should correct the value.
3. Now you have to decide whether Linux supports your CD-ROM. You should have done this in section 3.7, page 24. Just give exactly the same answer.
   - If Linux supports the CD-ROM just press ⏎ and you are set.
   - If Linux doesn't support your CD-ROM you should have copied the necessary files to your HD in section S2, page 24. Now select the `'HD'` option and press ⏎ . Now you have to specify the complete path to your directory `suse`. In our given example we used `\emil` meaning that you have to fill in `\emil`.
4. Now you should choose a suitable kernel if you have not done so yet. If you decided in section 3.7, page 24 that Linux does not support your CD-ROM, hopefully you copied a kernel onto your HD in section S2, page 24. Now you press ⏎ . Help on your decision about which kernel to choose can be found in Infoblock section 3.27.1, page 56.
5. The next screen requires you to decide whether to install **loadlin**. Answer `'yes'`. Setup now creates `\loadlin` and copies the necessary files into that directory.
6. With the option `'Starting Linux'` we start our ☞*base Linux*. Here you will notice about two pages scrolling down rather quickly. If everything went right you will be welcomed by S.u.S.E.. You can easily have a look at the kernel output by scrolling the text back using Shift + Page ↑ or Shift + Page ↓ , respectively.

Problems

There can be problems at two points. **loadlin** is not able to load the selected kernel or the kernel has problems with the specified hardware.

- **loadlin** doesn't have enough free memory to load the kernel: see section P8, page 55
- **loadlin** cannot load the kernel because the machine runs in virtual 8086 mode but no VCPI server is available: see section P11, page 55
- **loadlin** does not work: see section P9, page 55
- Your SCSI adapter is not recognized.
- The CD is damaged: see section P4, page 53

3.14 S7: Creating boot disks with Unix

What's it about?

You have a Unix already installed and want to switch to Linux.

Requirements

Your CD-ROM is supported by Linux and you have no way of installing Linux via a DOS partition.

Step by step

This is how you create your boot-disk:

1. If you have to format the disk: **`fdformat /dev/fd0h1440`**
2. Mount CD1(e. g. to `/mnt`):
   `cd / ; mkdir mnt`
   `mount -tiso9660 /dev/cdrom /mnt`
3. Change to the `disks` directory on CD:
   `cd /mnt/disks`
4. Now create your boot-disk

   `dd if=<disk> of=/dev/fd0`

   `README` file in `disks` directory and Infoblock section 3.27.1, page 56 contain specific information about the kernels.

3.15 S8: Starting linuxrc

What's it about

linuxrc loads every needed driver as kernel modules. Thereafter YaST starts as well as configuration of your system.

Requirements

For using the S.u.S.E. boot disk (or the creation of your own) together with CD you need a CD-ROM drive supported by Linux. Furthermore you need a bootable 3.5" drive (if you have an additional 5.25" inch drive which serves as boot-drive have a look at section P12, page 55).

If your CD-ROM is not supported you should have created the installation-disk with **setup**.

For booting directly from CD this feature has to be supported by your machine.

Another alternative: with **loadlin** you have already started your Linux.

Summary

The extreme flexibility of the kernel is achieved via kernel modules. Which modules are needed is entered by the user in **linuxrc**. This also applies for the S.u.S.E. boot disk, your own boot disks or the kernel started via **loadlin**. Even drivers loaded as modules might be supplied with parameters. The item kernel parameter is discussed thoroughly in section IB2, page 57 and section 7.3.3, page 111. Furthermore you can use the boot disk for starting the installed system at a later date as well, not only for installation (as an emergency "rescue disk" so to speak).

Step by step

Here's how to start Linux with a boot disk:

1. If you start with a boot disk:
   (a) Insert the boot disk into your drive and boot the machine.
   (b) A welcome screen appears, just press ↵ or wait a while ...
   (c) The message `"Loading Linux..."` appears and after a while the kernel boots.
2. Now **linuxrc** is running. Now you may select a language. ↑ and ↓ moves you up and down ← and → choose responses; here it's simply `'Ok'`. ↵ sends the command.
3. Now select color or monochrome monitor; `'OK'` moves you on.
4. Now the menus get much prettier. `'English'` is already set as default, so just press ↵
5. Now we are in the main menu. Here are the options available:

   Help Select if necessary

   Settings Here you can change language, screen and keyboard.

   System information Here a lot of interesting information about your machine can be obtained.

   Kernel modules Here is where you go to load modules.

   Exception: If your hard disk and your CD-ROM are connected to an (E)IDE controller you do not need to select this item. (E)IDE support is integrated into the kernel. You can skip this section and start installation; how this is done is described below.

   Installation /Start system Here the system boots. More on this is given below.

   A detailed description of **linuxrc** can be found in section 11.3, page 160.
6. Select loading of kernel modules using `'Kernel modules'`.
7. The next submenu asks which modules should (or must) be loaded. Normally these are:
   - If you have a SCSI hard disk or a SCSI CD-ROM: a SCSI module.
   - A CD-ROM module if your CD-ROM is attached neither to a SCSI nor to an (E)IDE controller.

- A networking module if you want to install via network (NFS)
- A PCMCIA module if you want to access the installation system via a PCMCIA card

8. This is how to proceed for loading kernel modules:
   - For selection you press the corresponding submenu and select the appropriate type. After a ⏎ there is an entry to enter parameters. Here typical settings are set as default. More on kernel parameters is in section IB2, page 57 and section 7.3.3, page 111.
   - You can also select `'Automatic loading of modules'`; most often this is sufficient. If unnecessary modules are loaded you can deinstall them using `'Remove loaded modules'`.
   - The result of loading is presented. If it failed, either the driver does not suit the hardware or the kernel parameters are wrong or missing.
   - With `'Show loaded modules'` you can check the result of your activities. You will see all loaded modules and you can identify accidently loaded modules.
   - Accidentally loaded modules can be removed with `'Remove loaded modules'`
9. If you have selected any needed modules, go to `'Installation/Start system'` in main menu. In the submenu popping up you indicate the installation system to take. Of course you have inserted the first CD.

   Now the installation tool YaST starts.

Problems

Problems can occur if:

- Your SCSI adapter is not found. Use a kernel with integrated SCSI driver.
- The ATAPI CD-ROM hangs. See section 3.26.5, page 53

3.16 S9: Invoke YaST

What's it about?

Now you start installing the real Linux. You do this with the installation tool YaST which you start either from CD, installation disk, or a reachable directory (depending on your installation). Then you set some general items on YaST and your Linux system.

Requirements

You have launched the ☞*base Linux* and YaST.

Summary

By providing YaST (YaST stands for *Yet another Setup Tool* which is a play on yacc *yet another compiler compiler*) S.u.S.E. offers you a means of easily installing the Linux system. You don't have to edit configuration files but rather decide what YaST should do for you. In most cases YaST "edits" those configuration files for you. Apart from such basic things as designing a proper keymap, YaST supports you while partitioning your disks, installing software and maintaining your system. Of course you can do everything by hand, but this is not the aim of this chapter. For special questions there are upcoming chapters and, of course, references to further reading. Here we only want to do a "first-time" Linux installation; we don't explore all of YaSTś abilities, but merely a few. If you want to get to know YaST more thoroughly or you have special needs refer to the YaST chapter (chapter 4, page 67).

Step by step

This is what to do to start YaST:

1. You are asked whether you want to install demo mode. Demo mode is based on DOS' filesystem (see section 4.5.2, page 71) and is remarkably slower than a normal installation. We choose `'normal'`, meaning just press ↵ .
2. Now you are back in YaSTś main menu (see section 4.5.3, page 72).

3.17 S10: Designing partitions

What's it about?

This step shows you how to create partitions or to change the existing partitions to Linux partitions.

Requirements

You have started YaST and have completed the previous steps correctly.

Summary

Depending on your HD you will either reduce existing partitions in size or create new smaller partitions for your OS, buy a new HD or decided to use existing partitions for your Linux. Then you will allocate space for Linux. Therefore you create Linux partitions and add several specifications for your Linux file system on your HD. You can find information about partition types in the Infoblocks.

After terminating YaST all partition types will be written to your partition section IB4, page 58 table. When writing partition tables you have to reboot your computer to let the changes take effect. Partition tables are, by the way, independent of the Linux system. There is only one on your HD and every OS installed takes its information from this table. Partition tables can be manipulated by any OS, and that's the reason why you have the system specific **fdisk** program.

If you already have an activated swap partition you should not change it in this step!

How to partition is a very philosophic decision: how many partitions, how big and mounted to which directories etc. For further information please refer to Infoblock section IB5, page 58.

Step by step

This is how you assign your partitions:

1. In YaST choose `'Adjustments of installation'` and herein the item `'Configure hard disk partitions'` (see section 4.5.4, page 72). YaST presents a screen with different sections:
   At the top there are the parameters of your HD.

   Below that there might be warnings or error messages from the **fdisk** program. You can inspect those messages more thoroughly by pressing F6 .
   In the lower part are those partitions that **fdisk** has found on your system. Here you can see, for example, partitions of your former operating system (e. g.MS-DOS). If you already have assigned a swap partition this will be listed here as well.
2. If you want to delete existing partitions you should do that first.

> If you delete a partition all data on this partition will be lost!

3. Go to the partition you want to delete by using ↑ or ↓ respectively. Ensure that you are on the right partition; look at the partition type as well! Then you press F4 and confirm with ↵ .

   If you plan to delete multiple partitions you should do that now.
4. If you want to use existing partitions (another OS) for use with Linux you can do this by just changing the partition type.

> If you change the partition type all data will be lost!

   Go to the partition whose type you want to change using the ↑ and ↓ keys. If you're absolutely certain you don't need this partition, just press F3 .
   You will now see a selection window for the partition types. Choose between *normal* Linux partition or a "swap partition" and confirm by pressing ↵ .
5. If you want to assign new partitions to Linux (this probably is the "normal" case) you should assign them one at a time. So just press F5 . If nothing happens you have no more space left on your device. You should remove unnecessary partitions first. A selection window appears where you can decide what type of partition to create. Please select `'primary partition'`, `'extended partition'`, or `'logical partition'`; confirm with ↵ . Please remember: you can assign no more than 4 primary partitions. If you need more than 4 partitions, the

last partition has to be assigned an extended partition; within this extended partition you can assign logical partitions. You'll find more information is to be found under section IB4, page 58.

If you have assigned a primary or logical partition you have to specify the device name of the partition. YaST offers you all available free device-names. Just select the uppermost name (e. g.`'/dev/hda2'`) and press ⟵.

The next step allows you to set the size of your new partition. YaST offers the first free cylinder as start cylinder. Normally you just accept this offer by pressing ⟵. Now you enter the end of the partition. Here there are three possibilities: the number of the end cylinder (e. g.`976`) number of cylinders for this partition (e. g.`+66`) or size in megabytes (e. g.`+100MB`). Pressing ⟵ leads you to `'more'`; please confirm with ⟵.

Now there is the freshly created partition in the lower window. If this is not as you wanted it you can delete it easily, just press F4.

6. One of your partitions has to be assigned a swap partition. If you have not done so already you can choose a suitable partition using ↑ and ↓ (a reasonably good size for a swap partition is approximately 20 MB); please have a look at section IB4, page 58 for further information. Press F3 and choose `'Linux-Swap-Partition'` and confirm with ⟵.
7. If all partitions have been set (your HD should be completely used; hope you have not forgotten the swap partition) you move the cursor to `'More'` by pressing TAB. Pressing ⟵ leads you back to the main menu.

3.18 S11: Assigning filesystems for partitions

What's it about

You have created all necessary partitions and these partitions have been added to the partition table (but they are still not written to disk!). There are some vital parameters still missing which will be given in this step.

Requirements

You have created the necessary partitions.

Summary

In this step you enter information for your Linux partitions (even for OS/2 partitions). This information is going to be written to `/etc/fstab` (this is the filesystem table). This file contains all constant values concerning your file system, such as device name, mount point in your Linux tree, or just the type of the partition.

YaST needs this data for creating Linux filesystems on your freshly formatted partitions. Swap partitions remain untouched in this step since they have already been configured in section S10, page 35 and only have one degree of freedom. As opposed to the partition table, the steps performed in this section are just concerning Linux and do not concern other operating systems located on other partitions.

If you want to know more about these mechanisms see the various resources in the bibliography

Also noteworthy:

- In Linux all filesystems are linked together on one tree (look at figure C.1, page 397). You must declare each filesystem a branch of the whole tree; this is going to be its ☞*Mountpoint*. Even DOS or HPFS filesystems can be linked into your Linux tree.
- Memory of filesystems is administrated under Linux by usage of ☞*inodes*. An inode just points to the corresponding data in your files. The number of inodes is assigned while creating a filesystem. If you want to create lots of small files you will need many inodes (which of course need some space for their own as well); for filesystems containing many large files, less inodes are necessary. More about this in chapter section 4.6.2, page 76.

Step by step

This is how you proceed to assign filesystems to your newly created partitions Go to the submenu, item `'Set target partitions / filesystems'` under `'Adjustments of installation'`:

1. First some information about what is to be done:
   - For your DOS HPFS filesystems (in DOS HPFS partitions) you *can* create a ☞*Mountpoint*.
   - For each of your Linux partitions:
     - you *must* select a mount point (if none is entered).
     - you *can* change the suggested ☞*inodesize*.
     - you *can* change the suggested formatting mode.
   - Functions `'setting type'` and `'reading fstab'` are not needed normally not needed.
2. If you want to access a DOS or HPFS partition under Linux you have to select it and press F4 . Now a window pops up and you can enter a directory. Under this directory you will later on find your DOS HPFS file system. e. g.enter `/dosc` (don't forget the slash!). Confirm this action by pressing ↵ .

   If you have selected a DOS partition another window appears; here you can choose which way to take for accessing your DOS under Linux (see section 4.6.1, page 76). Generally you want to use your DOS independently but just want to open a "window" under Linux for exchanging files. Please choose `'DOS'`. UNIX-file attributes and long filenames as offered by the ☞*UMSDOS* file system are only necessary for demo-mode.
3. Next you enter the ☞*Mountpoint* of your Linux partitions. Just select your Linux partitions one after the other and press F4 . A window will pop up letting you decide where to mount this partition on your filesystem tree. Give in the complete path beginning with the "slash" (/).

   Which partitions you give as mount point depends on how you plan to organize your partitions. In any case, you *need* a root-directory `/` which is the “root” to the whole filesystem. All other filesystems and their mount points are more or less due to your preferences. You should create a mount point for each Linux filesystem. You will not be able to access unmounted filesystems.

> Directories which are needed at boot-time must be directly under `/`; at this stage the separate branches are not yet put together (mounted). That’s the reason why you are not allowed to assign `/bin`, `/dev`, `/lib`, `/etc`, and `/sbin` to other partitions!

4. Now you set the i-node density for your Linux partitions. YaST sets a default value depending on the partition size. Generally: take an ☞*i-node* density of 4096 bytes per i-node with two exceptions:
   You have lots of little files (or you want to create them later on) or you want to use the live file system. In both the cases 1024 or 2048 are a better choice. Information on the Live-filesystem is in Infoblock section IB6, page 59.

   Now go ahead and choose those Linux file systems whose i-node density you want to change and press F5 . A window with all possible alternatives appears. Choose the one you want and press ↵ .
5. Now you should decide whether (and how) you want to partition your Linux partitions. Since we created those partitions newly in section S10, page 35 they have to be formatted. If you own a modern HD `‘normal formatting’` must do; if your storage media is not that new perhaps you should choose `‘Formatting and checking’`.

   Select the Linux partitions one after another press F6 to select the convenient formatting.

   Your partitions are not formatted instantly as your partitions were just designed. In this case YaST “remembers” this until the next boot. Then it will format your partitions accordingly, and the file systems will be created.
6. Select `‘More’` to go back to YaST’s main menu.

3.19 S12: Choosing base selection of software

What’s it about

Your HD is now prepared. Now you can decide which software to install. First you will chose the software you want and then the actual installation will occur later, after rebooting.

Requirements

You have made it so far and have an empty disk on hand.

If Linux doesn’t support your CD-ROM you have already selected software packages (in section S2, page 24) and copied them onto your DOS HPFS partition. Furthermore you have selected section 3.16, page 34 in `‘Installation via HD partition’` as installation source.

Summary

Let's summarize: The partition table is changed but has not yet been written to disk and you have decided how your future file system is going to look. You cannot yet install software but it is quite convenient to select it now.

You might find (while selecting packages) that your partitions are too small, too big or the relation is not the best. If this happened just go back to YaST(section S10, page 35 and section S11, page 37) and fix your settings until everything is to your satisfaction.

The software you choose for installation has to be saved somewhere. Since there is no HD already formatted and installed this is not possible. Therefore you need a disk for saving your settings. This disk is going to be formatted and used for saving your settings.

Now you have a whole CD full of compressed Linux software! What you choose is almost up to you, but there are certain packages that you need (series a). Series series `base` (a)is the "'base" of your Linux system. Other packages can be installed and removed as you like.

Step by step

Here's how to proceed to select your of base software installation:

1. In YaST select the item `'Choose and install packages'` and confirm with ⟵ .
2. YaST now asks for a disk. Insert a 3.5" disk into floppy drive A:, which, of course, should not contain any important data. If you don't insert a disk you are not able to save your settings, thus having to repeat the selection of packages later on once again.

   If the disk is inserted confirm with ⟵ . YaST now creates a filesystem on this disk; if necessary the disk is formatted first.

   Now you've got to be patient. Data has to be read from CD (or from HD if you've chosen this installation media). Afterwards there will be the installation menu (section 4.6.5, page 77).
3. Now you select `'Change /create configuration'`.
4. Now you are in series selection (section 4.6.8, page 78). In the uppermost window all series are listed with a short description. On the right hand side there is the amount in megabytes that you've already selected.
5. Now you can go through your series and packages, choose packages you would like to have, but have an eye on your free disk space. Don't forget: you *urgently* need every package from series a. How selection of series and packages work is described below.
6. What can I do in series selection?
   - Pressing ⟵ leads you to package selection within the chosen series.
   - In the lower window you can view how your Linux partition gets filled.
   - Using F3 YaST shows you your partition in a large window.

- F5 lets YaST check for dependencies of the selected packages. Dependencies take into consideration that certain packages require other packages to be installed, and sometimes packages conflict with each other.
- F10 brings you back to the installation menu.

7. Now choose a series that you are interested in. Press ↵ and the package selection list of that series appears (section 4.6.8, page 78). In the upper left corner of the screen there is a list of the packages, in the upper right corner there is the free space left on device and below a description of the actual selected package.
8. What can I do in package selection?
   - ↵ selects or unselects the highlighted package respectively. `[X]` means that this package is going to be installed.
   - F2 shows a detailed description of the respective package, ↵ leads you back to package selection.
   - F3 gives a detailed view of your partition's free space.
   - F10 leads you back to package selection.
   - In the upper righthand corner you can check your partition's space.
9. If you're done with series and packages and everything is selected as you like (and everything fits on your HD) then you can switch back to series selection with F10 and, by pressing it a second time you will be brought back to the installation menu.
10. The menu entry '`What if`' shows you what is going to happen (which packages are going to be installed). You don't have to select this menu entry, it's just for informational purpose. This item is very useful if you plan (later on) to remove packages to make sure nothing unwanted is about to be removed.
11. Now your selection has to be written to disk. Therefore you just select '`Save configuration`'. In the box that appears you can now enter a name for your configuration. The entry below serves that as a simple description entry; here you can enter a short explanation of your configuration. This might come in handy if you plan to use this configuration later on.

    Go to '`Save configuration`' using TAB and press ↵ . Your selection is now written to disk.
12. Next you go back into YaST's main menu by choosing '`Main menu`' and pressing ↵ . If you hit '`Start installation`' you will be shown an error box since this is not possible yet.

3.20 S13: Quitting YaST and rebooting

What's it about

Anything that could be done without a usable hard disk has been done. Now you have to shut down and reboot your Linux system for making the Linux partitions available.

Requirements

If you have made it that far all requirements are fulfilled.

Summary

We have already mentioned that your HD is not ready to be used yet. By terminating YaST you write this new partition table to HD. Now you have to reboot *immediately*! You don't have to hurry but you should not start any other programs before having rebooted. You shut down Linux with the following command: **`shutdown -r now`**. *Don't just switch off your machine!*

Now your machine should boot and you have to proceed exactly as you did before (booting via CD-ROM section S6, page 30 or boot disks section 3.12, page 29)

After starting from CD lilo is already installed but you cannot use it right now (nor the batch file **linux.bat**); you have to start via **setup** on CD 1 again.

Step by step

That's how to proceed to terminate YaST and to reboot:

1. Choose `'Finish installation and exit YaST'` and confirm with ↵ . Now there is a message that your partition table will be written; Eventually YaST reminds you of rebooting your machine immediately. Confirm this with ↵ . Now you should see the system prompt.
2. Enter **`shutdown -r now`** and press ↵ . After a couple of seconds booting commences.
3. You have to boot just like you did at the first time. If you boot via disk insert the boot disk (just like first-time: either S.u.S.E. boot disk or your self created boot disk). If you boot using two disks you don't need to create installation-disk since you already have it.

   If you have booted via **loadlin** remove your disk from the floppy drive. After booting has finished start the program **setup** from CD. You have to go through the whole program again; the only thing that differs is that you don't have to install loadlin again since this has already been done in section S6, page 30. Here is a quick reference of what you have to enter in **setup**. If you encounter problems please go back to section S4, page 27 and section S6, page 30

   - Choose `'English'` (or German if you prefer...)
   - Enter your CD-ROM drive (letter).
   - Confirm the welcome screen with ENTER
   - Choose `'loadlin'`.
   - Now select `'CD'` or `'HD'` just like in section S6, page 30.
   - Select a suitable kernel like in section S6, page 30.
   - Select your CD-ROM drive just as in section S6, page 30.
   - Select `'Loading Linux'`.
   - Now you have a Linux prompt again; ready for logging in.

3.21 S14: Starting YaST again and installing software

What's it about

Now we are up to formatting the partitions and assigning file systems. Thereafter Linux software will be installed.

Requirements

You have booted and have got a Linux prompt on your screen, plus you have your disk at hand on which your software selections have been saved.

Summary

Now your partition table has been written but all freshly assigned partitions have not been formatted yet. They don't even contain any filesystems yet. If you login "RAM Linux" runs again.

After starting YaST first your partitions are formatted, then filesystems assigned. Next your swap partition is activated. Next you can start installing your selected software packages.

Step by step

Here's how to start YaST and to install software packages:

1. You now have a Linux prompt; log in using user name `'root'`.
2. Now there is the "welcome screen" with lots of good hints and tricks; since you've already read them previously you can skip it.
3. Enter **`yast`** and YaST starts.
4. Now you have to answer some questions you already know. Here is a quick overview:
   - Choose `'English'` (or German if you prefer).
   - Now you have a disk containing your software selection (you'll remember we created it in section S12, page 39). Insert this disk and confirm with `'Yes'`.
   - Now YaST takes care of the freshly created partitions. Select `'Formatting'` and watch YaST formatting and assigning filesystems on your drives.
   - Confirm your swap partition by entering [↵]. Then your swap partition is activated.
   - Choose `'Normal'` if the system asks you whether to install in demo mode or normal mode.
5. Now you are taken back to YaST's main menu. You can now finally install your selected software. You have saved your software selection to disk. Select `'Setting up installation/starting'` and wait a while until all descriptions have been read. Then you will see the installation menu.

6. Now you choose `'Load configuration'`; a window with a couple of configurations appears. Most of them have been predefined by S.u.S.E.. Your own configuration is located at the end of the list. Select it (or do you want to try one of S.u.S.E.'s configurations?) and confirm loading. After a while you will see the installation menu again. Please notify that your configuration is now set at the top line.
7. If you like you can now adapt your software configuration to your personal needs. This is achieved easily by moving to `'Changing/creating configuration'` and confirming it using ↵. This changed configuration of course can be saved to disk using `'Saving configuration'`. This task has already been described in section S12, page 39.
8. Now you start installation with `'Starting installation'`. YaST presents you with a list of packages that are to be installed and which packages already have been installed.
9. Go back to main menu by selecting `'Main menu'`.
10. Select `'Finishing installation and leaving YaST'`
11. Select a kernel.
12. Now you have to confirm to create a boot disk. Since your disk containing the software installation is still in your drive YaST takes this and turns it into a boot-disk.
13. Next you'll see the question whether you want to configure lilo; this question is to be answered with `'No'` since this is explained in system management later on.
14. YaST now informs you that you have to boot your machine afterwards; you should confirm this.
15. Now you have the Linux prompt back. Enter **`shutdown -r now`** for starting your system properly.

Problems

There is one problem that can occur:

- The error message `"mke2fs failed"` or similar messages might be displayed on the screen; see section P10, page 55

3.22 S15: Starting YaST for the last time and finishing up installation

What's it about

This step is for customizing your Linux system.

Requirements

You have installed the base software packages.

Summary

The machine does not reboot, but continues with loading your installed Linux. YaST starts automatically.

YaST now asks you a couple of questions for configuring additional devices as mice etc. You do not have to configure these things right now since you can invoke any time and configure your system. If you are unsure just skip this item and do it later on.

Since access to the CD now is possible YaST may install even software that is not on CD 1.

Step by step

Here's how to customize your Linux system:

1. YaST now requests a CD change and automatically installs the selected software that has not been installed yet.
2. YaST now asks whether you want to test some screen fonts. If you do so, just answer `'yes'`. Choose the font you want from the list and select `'test'` for testing a font, or `'cancel'` if you want to stop testing.
3. If you have selected `'test'` the screen switches to the new font. Check if all special keys (such as German umlauts if you need them) are accessible! If you do not like the font just select `'Other font'` and the game starts all over again. If you decide to keep it, select `'Standardfont'` to keep it.
4. YaST now enquires if you want to configure your modem. If there is a modem installed you can do so right now or come back here later on. If you say `'yes'` you are asked for the serial device of the modem.
5. YaST now wants to know if you want to configure your mouse, If you want to do so right now select `'yes'`. Select your mouse-type. If it is a serial mouse YaST needs to know the serial interface where it is connected.
6. Please confirm the start of **gpm** at boot-up. If you encounter problems with it just come back here and remove it.
7. Now enter a time-zone. Here again there are yards of entries.
8. YaST now creates a login for you since you should not work with the system as `'root'` but with your own login. Think of a nice little nickname (without spaces). A password can be entered later on when you log in for the first time. This also applies to `'root'`.
9. Now you are back to YaSTś main menu. You have done it!
10. Now you can select any menu item. Most of them should already look familiar to you. You may as well leave YaST and log in using your new user name. By the way there are far more consoles than one. Press Alt + F2 to Alt + F6 for switching consoles.
11. If you have logged in using your new user account you should enter a password by giving **`passwd`**. Then you are asked for a password which has to be entered twice.
12. Now you may work. You can enter (e. g.) **`ls -a`** for listing all files in your home directory.

3.23 E4: How do you plan to start your Linux in future?

What's it about

Your Linux system is now almost completely installed. Now you can decide how to start your Linux on a regular basis.

Summary

In principle there are three ways of starting your Linux:

1. You start Linux via **loadlin**. This is the most comfortable method but requires certain steps to be taken.
   - Your machine must run in real-mode under MS-DOS, or in virtual 8086-mode[4]with an VCPI server available (in other words: under Unix, OS/2 or DOS-box of Windows 95 this does not work). It works fine under DOS-prompt or in Windows 95 DOS-mode.
   - You must have enough memory below 640 kB. Here you can encounter problems if too many drivers are loaded.
2. You start Linux via boot-disk. This always works and needs no work to be done, as we have created this boot-disk in section S14, page 42; of course you can use S.u.S.E. boot disk as well. Boot disk also is a rather good alternative if you have problems using the other two methods.
3. The most advanced method, of course, is use of the boot-loader LILO. LILO lets you decide which system to boot (before the system actually boots). Configuration of lilo is a little tricky. It has to be installed in the boot sector of your HD, which means a couple of problems can occur. Therefore configuration and installation of lilo is discussed in a chapter of its own and not here in the installation section.

Recommendation

The best thing to do if you're working with DOS, Windows 3.x, or Windows 95 is the "loadlin" method. This will work in almost any case. Together with OS/2 or other operating systems working in protected-mode, or if **loadlin** doesn't work use the boot-disk.

3.24 S16: Starting via loadlin

What's it about

If you plan to use loadlin for starting Linux there are a couple of things you have to prepare in advance. Depending on your system there are some files which have to be adapted.

Requirements

You want and are able to use loadlin (at least you hope so.;)).

[4] offered e. g. by **emm386.exe**

Summary

In principle there are two ways of activating loadlin. One is to switch between various systems via boot menu or to start Linux via **loadlin** from another running system.

Both methods have advantages and disadvantages:

- A boot menu saves you the detour via another operating system.
- You can add other configurations to your boot menu for creating an universal starting mechanism.
- A boot menu is the more flexible way and you will be able to solve problems if they occur.
- You have to modify start files to build a boot menu which you will have to do by trial & error.
- Changing to Linux from the DOS prompt is very simple.
- Linux start can be very nicely integrated into a Windows 95 session. Double clicking an icon will start Linux (Windows 95 contains DOS 7.0).

Recommendation

You should use a boot menu from DOS or Windows 3.x. If using Windows 95 you can start from the running system.

Start menus under Windows 95 are a very complex matter so please understand that we are only able to give you some hints on them.

Necessary steps for all loadlin users

This is what you have to do whether you decide to use a boot menu or if you start from a running system (under DOS, Windows 3.x, or Windows 95):

1. Probably you already installed **loadlin** (this was done in section S6, page 30); if not, you have to do so via **setup**.
2. Change to `c:\loadlin` (under MS-DOS). There you will find a file called `linux.par`. Create a file named `startlin.bat` (or any other name). Now insert the following line as described in file contents 3.24.1, page 47:

```
c:\loadlin\loadlin @@c:\loadlin\linux.par
```

File contents 3.24.1: Example of a batch file for starting Linux

Now, please edit the file `linux.par` (file contents 3.24.2, page 48).

Instead of `xxx` you have to enter your root partition's device name (you wrote down this name in section S11, page 37). `startlin.bat` starts your Linux. The file `linux.par` is used by `startlin.bat` as well as by `config.sys` and contains essential parameters. Later on, after getting more familiar with Linux you can of course add/replace parameters in here. If you have built a kernel of your own just copy it to `c:\loadlin\vmlinuz` and it will be booted from now on.

```
c:\loadlin\vmlinuz  # first value must be
                    # the filename of the Linux-kernel

root=/dev/xxx       # the device which gets mounted as root FS

ro                  # mount root read-only
```

File contents 3.24.2: Example of the file `linux.par` with customized values

Using a boot menu if you have DOS or Windows 3.x

Here's how you configure a boot menu under DOS or Windows 3.x respectively:

1. First you have to define a boot menu in `c:\config.sys`. Enter something like file contents 3.24.3, page 48

```
[Menu]
menuitem=Win, starting Windows,...
menuitem=DOS, starting DOS, MS-DOS...
menuitem=Linux, starting Linux...
menucolor=15,1
menudefault=Win,5
```

File contents 3.24.3: Example for `config.sys` (first part) for the Linux start

   Under the label `[Menu]` you define entries in the boot menu, its color and after how many seconds which item is started automatically.
2. Below you enter the labels [Common], [Win], [DOS], and [Linux]. Things written in [Common] are always executed, all other entries are system specific. Use the lines that are in your current `config.sys` as our example certainly does not suit everyone's (Diagram file contents 3.24.4, page 49).

   Now save the file.
3. Now open `c:\autoexec.bat`. Here you have to put exactly the same labels and assign entries to labels. Notation differs slightly. *Linux case* is mentioned first. Which label has been selected is to be found in variable %config%. Perhaps you write something similar to diagram file contents 3.24.5, page 50.
4. If you boot your machine now the boot-menu appears and you have five seconds to choose an operating system. After 5 seconds Windows starts automatically. If you select `'Linux'` Linux starts and awaits your login.

Starting Linux from within Windows 95

Here's how you create a start icon for Linux for booting Linux from within a running Windows 95 session:

```
[Common]
device=c:\dos\himem.sys /testmem:off
device=c:\dos\emm386.exe noems I=E000-F4FF
dos=high,umb
files=30
buffers=10
shell=c:\dos\command.com

[Win]
devicehigh=c:\dos\dblspace.sys /move
devicehigh=c:\cd\slcd.sys /D:SONY_000 /B:340 /M:P /V /C

[DOS]
devicehigh=c:\dos\dblspace.sys /move
devicehigh=c:\cd\slcd.sys /D:SONY_000 /B:340 /M:P /V /C

[Linux]
shell=c:\loadlin\loadlin.exe @@c:\loadlin\linux.par

[Common]
rem Remains blank
```

File contents 3.24.4: Example for `config.sys` (second part) for the Linux start

1. Open the **explorer** change to `c:\loadlin`, drag the file `startlin.bat` onto your desktop.
2. Go to a file or onto the desktop (depending on where you want your Linux icon) and press the right mouse-button and select `'Create link'`.
3. Mark the link just entered, press the right mouse-button and select `'Properties'`. Go to register `'programs'` click on the button `'extended'`. Cross entry `'MS-mode'`. Confirm with `'OK'`.
4. Via `'Change icon'` you can select a nice icon; give the link a suitable name; Voila!
5. Double clicking this new symbol should bring up a mask telling you that Windows 95 is about to switch to DOS-mode. If this mask bothers you: turn it off in the properties menu.

An alternative: a boot menu with Windows 95

Here's how to proceed to install a boot menu for Windows 95:

1. First you have to edit `c:\msdos.sys` Make the file visible by entering:
   C:> `attrib -R -S -H c:\msdos.sys`
   This is a text file where you have to enter some lines for deactivating Windows 95' start menu. Label [Options] should resemble file contents 3.24.6, page 50
   The parameter `Logo=0` is optional and avoids switching to graphics mode before Windows 95 is started; booting is much faster and you avoid lots of trouble if you plan to use the DOS emulator on Linux later on.

```
@@echo off

rem Entries for all Configurations
switches= /f
set comspec=c:\dos\command.com
prompt $p$g
loadhigh c:\dos\keyb gr,,c:\dos\keyboard.sys
loadhigh c:\dos\doskey
set temp=c:\temp
loadhigh c:\dos\mscdex.exe /D:SONY_000 /E /V /L:H
c:\logimaus\mouse.exe

goto %config%

:Win
c:\dos\smartdrv.exe a- b- c+ 2048 1024
path c:.;d:.;c:\windows;c:\dos;c:\util;
win :
c:\dos\smartdrv /C
goto ende

:DOS
path c:.;d:.;c:\dos;c:\util;
goto ende

:ende
echo * Goodbye *
```

File contents 3.24.5: Example for `autoexec.bat` for Linux start

```
[Options]
BootGUI=0
BootDelay=0
BootMenu=0
Logo=0
```

File contents 3.24.6: `msdos.sys` for Linux start

The parameter `BootGUI=0` is for booting Windows 95 directly into DOS mode. For starting Windows out of this you have to enter:

C:> `win` but this is already done by our example `c:\autoexec.bat` if you have selected Win95 from our menu.

2. Then you have to define your own boot menu in `c:\config.sys`. Therefore you should enter (diagram file contents 3.24.7, page 51):

```
[Menu]
menuitem=Win95, start Windows 95...
menuitem=DOS, start MS-DOS...
menuitem=Linux, start Linux...
menudefault=Win95,5
```

File contents 3.24.7: Example `config.sys` (part one) for starting Linux under Windows 95

Under the Label `[Menu]` you define entries of the boot menu, its color and the delay for starting up automatically.

3. Further down you enter labels `[Win95]`, `[DOS]`, `[Linux]`, and `[Common]`. `Common` is for entries that apply every time (this will very rarely be the case under Windows 95); all other labels are for the corresponding system. Use those lines that are already written in your `config.sys`; the example in our diagram file contents 3.24.8, page 51 should only be regarded as a hint.

```
[Win95]
dos=high,umb
device=c:\windows\himem.sys /testmem:off

[DOS] device=c:\plugplay\drivers\dos\dwcfgmg.sys
dos=high,umb
device=c:\windows\himem.sys /testmem:off
device=c:\windows\emm386.exe noems I=B000-B7FF
devicehigh=c:\cdrom\torisan.sys /D:TSYCD3 /P:SM

[Linux]
shell=c:\loadlin\loadlin.exe @@c:\loadlin\linux.par

[Common]
accdate=C+ D+ H+
switches= /F buffers=20
```

File contents 3.24.8: Example `config.sys` (part two) for starting Linux-Start under Windows 95

Now save this file.

4. Now open `c:\autoexec.bat`. Here you have to enter exactly the same labels. Notation slightly differs. Which label has been selected is written

to variable %config%. Perhaps you should write something similar to file contents 3.24.9, page 52

```
@@echo off
loadhigh keyb gr,,c:\windows\command\keyboard.sys
goto %config%

:Win95
win
goto ende

:DOS
path c:.;d:.;c:\windows\command;c:\util;
loadhigh c:\windows\command\mscdex.exe /D:TSYCD3 /L:x
loadhigh c:\windows\command\doskey
c:\windows\command\mouse.exe
goto ende

:ende
echo * And now? *
```

File contents 3.24.9: Example `autoexec.bat` for starting Linux under Windows 95

5. If you now boot your machine there should be the Windows 95 boot menu should appear, giving you two seconds to select an entry. If you choose *Command line* your own boot menu will be displayed. Now you have five seconds to select a system. After this delay Windows 95 starts automatically. If you select `'Linux'`, Linux starts and awaits your login.

3.25 After your system is installed

Congratulations! We have achieved a lot! You have a working Linux with lots of software installed on your HD. Don't be afraid, it's not going to be dull now. As the system administrator there still are lots of tasks to perform. Most of it is explained in our S.u.S.E. Linux; more help is available on the CD-ROMs.

Here are some forthcoming topics:

- Installing the X-Window System
- Installing LILO (if you want to boot via LILO)
- Building a kernel of your own
- Installing aps filter (it prints almost everything)
- Connecting to a LAN
- Mail
- Connecting to the internet
- Configuring the DOS-emulator

3.26 Problem description

3.26.1 P1: Files cannot be moved

Files with the `system` or `hidden` attribute set can't be moved by defragmentation programs. If you want to find out where the problem lies just type:

```
attrib \*.* /s > <file>
```

Now you can see a whole list of your HD in <file>. Here you can identify the files that cause problems and change them using:

```
attrib -S -H <file name>
```

Here you should proceed with care not to delete copy-protection files, permanent swap files, or other system specific files. After defragmentation you should undo your changes of attributes.

If this doesn't work, then you *really* have to repartition your HD. This means that you have to save all your important files for restoring them after you have finished repartitioning your drive. The alternative is, of course, buying a new HD. Since prices are going down steadily this might be a good solution.

3.26.2 P2: No English keymaps in DOS-mode

In DOS-mode you enter the following line:

```
oloadhigh keyb gr,,c:\windows\command\keyboard.sys
```

or you simply add this line to your `autoexec.bat`. If your Windows is located elsewhere then of course you have to adjust the path accordingly.

3.26.3 P3: No CD-ROM driver in DOS-mode

In MS-DOS mode only those drivers are available that have been loaded in `config.sys` and `autoexec.bat`. It makes sense not to put your CD-ROM driver into those files since Windows 95 has its own drivers. For using those drivers in DOS mode you have to create a link to an MS-DOS command line and insert properties `'program'` under `'extended'` to create own startup files. There you have to enter your drivers for using those drivers in a DOS box.

3.26.4 P4: CD is damaged

This possibility is extremely unusual but who knows, it can happen.

3.26.5 P5: ATAPI-CD-ROM hangs while reading

If your ☞*ATAPI* CD-ROM is not recognized or it hangs while reading this mostly is due to incorrectly connected hardware. Generally all devices should be connected to the EIDE controller in the correct order. The first device is master on the first controller, the second device is slave on first controller. The third device should be master on the second controller and so forth.

Now it often occurs that there is only a CD-ROM besides the first device. The CD-ROM drive sometimes is connected as master to the second controller (secondary IDE controller). This is wrong and can cause Linux not to know what to do with this “gap”. You can try to fix this by passing the appropriate parameter to the kernel (`hdc=cdrom`) (see section 7.3.2, page 102). Nevertheless connecting both devices to the same controller is much better.

Sometimes one of the devices is just “mis-jumpered”. This means it is jumpered as slave but is connected as master or vice versa. In case of doubt just check your hardware settings and correct them.

Finally there exist a couple of faulty EIDE-chipsets, most of which have been recognized and the code of the kernel has been built to handle them. There is a special kernel for these cases. The parameters are described more thoroughly in section 7.3.2, page 102 and chapter 13, page 181.

3.26.6 P6: Mitsumi driver announces: mcd: init failed

This does not be have to be anything serious, provided you have a Mitsumi CD-ROM drive and it is not an ☞*ATAPI* device.

- There is no Mitsumi CD-ROM drive in your machine but your kernel contains the Mitsumi driver. This is alright.
- You have a Mitsumi ATAPI drive installed (e. g.FX001DE, FX200, FX300, FX400). You should have used a kernel with EIDE drivers not a kernel with the mcd driver.
- You really have a Mitsumi CD-ROM that’s not an ATAPI CD-ROM (e. g.Lu005, Lu005S, FX001, FX001S, FX001D). Then your drive is not set to the default values 0x300, 10. You should give the correct parameters to the kernel at boot time. You can assist your driver using the following line:

  `mcd=<addr>,<IRQ>[,<mcdwait>]`

  <addr> is the hexadecimal port address (e. g.0x300) <IRQ> stands for the used interrupt. The last parameter can be omitted first (see section P7, page 54).

  More information is to be found in section 7.3.2, page 102.

3.26.7 P7: “Unable to mount root” using Mitsumi CD-ROM-drives (Not ATAPI-drives)

This problem may occur while installing via a Mitsumi (non-ATAPI) drive. The kernel recognizes the drive but claims due to timing problems: `"No Disk in Drive"`, although the CD is inserted. Most of the times, but unfortunately not always it helps setting the last parameter to *5*.

`mcd=<addr>,<IRQ>,5`.

<addr> stands for hexadecimal port address (e. g.0x300), <IRQ> stands for used interrupt

More information about kernel parameters is to be found under section IB2, page 57 and section 3.26.6, page 54.

If this all doesn’t work you have to install via disks. You’ll have to go back to section S4, page 27 and select *disks* in section E3, page 28.

Our experience has shown that this generally works. Obviously the DOS CD-ROM driver puts the drive into a wrong state, thus making it impossible for Linux to recognize the drive correctly. Therefore you should reset your machine before trying to install via disks or switch it off for a moment.

3.26.8 P8: Loadlin doesn't have enough memory to load the kernel

You don't have enough free memory below 640 kB. Try to remove drivers from your startup files or shift them to high memory.

If you use compressed drives under Windows 95 and shifting the driver to high memory doesn't work you have to decompress those drives.

3.26.9 P9: Loadlin doesn't work

If you encounter any problems using **setup** or **loadlin** you should start **setup** like this:

`setup -DLOADLIN_TEST`

Now use the button `'Run loadlin -t'` for creating an diagnostic file: `c:\loadlin\susesave.dbg`.
Please send this file to technical support.

3.26.10 P10: Error with mke2fs

This annoying problem rarely occurs. Unfortunately it is only to be solved by moving the end of the partition forwards or backwards one cylinder. Of course the beginning of the next partition has to be adjusted accordingly.

3.26.11 P11: DOS runs in protected mode

loadlin can only boot the kernel if the machine isn't running in protected mode, or if a VCPI server is running. If you're working under Windows 95 you have to switch to MS-DOS-mode.

- Either you go via the `'Start'`-Button to `'Shut down'`, then `'Restart computer in MS-DOS-mode'` or
- you create a link to the MS-DOS command line and change the properties accordingly: In register `'program'` you select `'extended'` and mark with an "X" on `'MS-DOS-mode'`. If you now restart your machine it will be placed in MS-DOS mode.

3.26.12 P12: The 3,5 floppy drive is connected as B: and not bootable

PCs can only be booted from the first physical floppy drive (this is A: under MS-DOS). Moreover this drive has to be marked active in BIOS otherwise you can only boot via HD.

If you have accessed your 3.5 floppy as "B:" you have to change the connections of those drives (5.25 and 3.5):

- Turn off your machine and open the case.

- Look for the broad data cable which connects floppy drives to the controller.
- There normally are two pairs of plugs (one for each floppy). Only one plug from each pair is used (3.5 and 5.25 have different plugs). You now have to exchange both connections using the appropriate plugs. If this doesn't work due to short cables you have to swap the drives' positions as well.
- If your cable only has two single plugs (no pairs) you have to buy a new data cable.
- Close the case again and go to the setup-menu after switching it on. How you can access setup depends on your BIOS. A "nice" BIOS gives instructions on screen. If not look int your manual or try the Del key.
- Change position of your floppy drives in the setup menu:
  Drive A: 1.44 MB, 3,5-floppy
  Drive B: 1.2 MB, 5,25-floppy.
- Now mark this floppy drive as active. Here you have to enter submenu `'Advanced C-MOS setup'` (or similar). You should see a line `'System boot-up sequence'`; here you designate A:as the first drive in the boot sequence.
- Save these settings and leave setup.

3.26.13 P13: Label of CD-ROM drive has changed

If you have created an additional partition using **fips** this is a DOS-partition. Therefore all other drives are shifted back one step and your CD-ROM is no longer `D:` but `E:`.

After changing partition types with YaST this changes again; your CD-ROM drive should now be `D:`.

If you are not able to open the **Windows explorer** or **desktop** under Windows 95 this means that it still tries to access `'E:'` (using the above example). You have to use the system management tool to change your CD-ROM label to `'D:'`.

3.27 Infoblocks

3.27.1 IB1: Selecting a kernel

As installation proceeds you have to select a kernel for driving your system during installation and perhaps afterwards as well. The Linux kernel contains drivers for certain drives (networks, HDs, CD-ROM drives). The kernel must be the right one for your machine.

On our CD there are lots of different kernels either as disk images for boot disks or as files for loadlin. Those kernels are temporary since they must support a lot of hardware but can't be too large. Later on you should build your own kernel, specifically adapted to your system.

In most cases you can use our supplied boot disk for installation. Only if this kernel supplied with our boot disk doesn't recognize your hardware properly will you have to try another kernel. You can get more information about kernel configuration online via **setup** or in files `/disks` or `/suse/images` on CD 1, respectively.

For covering almost any hardware configuration there is a modular kernel installed on your system which contains only the absolutely essential drivers necessary for booting. So there are two kernels for EIDE-machines and several for SCSI-machines. All other drivers are loaded by the kernel daemon as modules.

If you encounter problems make sure your hardware is correctly specified in `/etc/conf.modules`. The other option is to build a monolithic kernel which includes all drivers (some driver still cause problems if loaded as modules).

or you build a monolithic kernel (some drivers still cause problems if loaded as modules).

3.27.2 IB2: Kernel parameters

If you have chosen a suitable kernel you have to decide which kernel parameters are needed. Kernel parameters are your way of "talking" to the kernels drivers. Using kernel parameters you can achieve several things:

- You tell the kernel which hardware you use and how it is accessed (address, interrupt etc.). Sometimes the driver can find out by itself but it is much safer and faster if you do this manually.
- You can disable drivers. This might be helpful if there are several CD-ROM drivers provided in the kernel and each of them tests its hardware in a very complicated manner (auto-probing). This only is relevant if you use a monolithic kernel which includes all CD-ROM drivers.
- In case of problems you can probably achieve a running Linux by using specific kernel parameters.

You can enter kernel parameters either if you boot via **loadlin** or if you boot via **lilo**. Even with S.u.S.E.boot disk which supplies the relevant drivers as modules you can enter kernel parameters. In this case the notation of kernel parameters differs slightly. You have to enter those parameters every time you boot. The kernel doesn't remember.

After installation you can put kernel parameters into `/etc/lilo.conf` or `/etc/conf.modules` (if you use a modularized kernel) respectively where they are read automatically by **lilo** or **modprobe**, respectively.

More about parameters, which other parameters exist and how you have to enter them is explained in chapter section 7.3.2, page 102.

3.27.3 IB3: Does Linux support my CD-ROM?

Generally it can said that almost every CD-ROM drive is now supported by Linux.

- Using ☞*ATAPI*-drives (those drives connected to an EIDE-controller) there should be no problems at all.

- Using SCSI CD-ROM drives it is only important whether the SCSI controller is supported by Linux (supported SCSI controllers are listed in an appendix). If your SCSI controller is not supported and there is a HD connected to it you will have a problem anyway.
- Lots of proprietary CD-ROM drives are supported under Linux (see Appendix). Here it is most likely that problems occur. If your drive is not mentioned you could try with a similar type.

3.27.4 IB4: Partition types on PC

Each HD contains a partition table which has space for four entries. Each entry can either be a primary partition or an extended partition where only *one* can be an extended partition.

Primary partitions are very simple; they are a continuous range of cylinders which are assigned to an OS. Using primary partitions you can only assign four partitions per HD. More won't fit into the partition table.

Here the extended partition principle comes into use. The extended partition is one that can be subdivided into several *logical* drives which don't need a separate entry in the partition table. The extended partition is more or less a container which contains all logical partitions.

If you need more than four partitions you can simply make the last partition an extended partition and give it the remaining free cylinders (space). Here you can now create as many logical partitions as you like.

Linux doesn't care where it is installed (primary and/or logical).

3.27.5 IB5: Partitions for the Linux system

The philosophy of Unix file systems (therefore Linux as well) differs from those used by MS-DOS file systems. There are no different drives but only one directory tree. Its branches can be on different drives but this is usually not seen by the user.

Division into several partitions includes advantages and disadvantages. Because partitions are independent of each other, possible problems are restricted to the respective partition itself. But since the partition sizes are fixed, your system can't "breathe". If the partition is full you need to repartition or to add another partition to your tree.

How many partitions should you create? Minimum should be three partitions:

- A root partition which contains the system and all other commonly used files. This is the root of your tree and is designated by `/`.
- A home partition which contains all user-specific data. This one is added (mounted) as `/home`.
- A swap partition which is not included in the file system tree. This is used by the system to enlarge available RAM.

Of course you can divide up your system into more partitions. It is recommended that you make the root partition rather small for minimizing access to this HD and thus improving system security.

In case of doubt or if you are unsure of what to do and you had a DOS or Windows 95 partition you should add three primary partitions.

3.27.6 IB6: The live file system

Our feature *live file system* integrates programs from CD (labeled `live system` into an existing Linux directory tree. YaST creates symbolic links which point to programs on CD. So you can use all programs from CD without having to install them and thus minimizing necessary HD space. The advantage of the live file system is its minimal space requirements. But there are disadvantages as well:

- Accessing a live file system takes longer since accessing CD-ROMs is much much slower than accessing HDs.
- Since many symbolic links have to be created and each of them uses one inode you must have enough free ☞*Inodes* available. The amount of inodes can be defined while creating file systems; you can only change them afterwards by repartitioning your HDs.
- For integrating the live file system your partition has to be at least 50 MB in size.

3.27.7 IB7: Demo mode

Since a lot of first-time users want to "play" with Linux before repartitioning the whole disk we have included the possibility of installing a "demo mode" onto an existing MS-DOS file system. A minimal Linux system will be installed in a directory in your existing MS-DOS file system which needs approximately 20 MB. Access to the 600 MB live file system is reached via symbolic links which point to certain programs on CD.

Dynamic integration of the live file system is not implemented in this mode since about 40000 links have to be created. As the current implementation of the ☞*UMSDOS*-filesystem with HDs of 8k-clusters would use up to 300 MB this, of course, would make the term "demo-mode" absurd. Nevertheless you are free to perform a "normal" Linux installation on top of a DOS partition. But you should realize that performance and system security are less optimal.

> On the DOS partition where you want to install the "minimal system" there should not be any file named `linux`. This special directory will take up your base-system!

For booting your demo system later on there are two options. Either you start it via our boot disk or you can use **loadlin** which enables you to start Linux from a running DOS system. This tool was used during first-time installation provided that you have DOS installed and you have a DOS-boot configuration that runs your machine in real mode not in protected mode.

3.28 Partitioning for novices

Is Linux and its filesystem completely new to you? You probably ask yourself questions like: How much space should I assign to Linux? What's the minimum I need? What's best for my needs? How should I divide up the available space?

Let's start with the minimum: 80 MB. This only works if you use the machine for a simple purpose: You only work on a text console (no X-Window system). If you want to peek into X and start a couple of applications: 200 MB. Both values include swap.

What's an extensive installation? 500 MB. In this world of giga-byte hard disks this is rather modest. Swap included and there is no upper limit.

What's the best for your needs? That depends on what you want to do:

- X and modern applications like **Applix*ware*** or **Netscape**: 700 MB to 1 GB.
- For creating your own little applications on X you also need 700 MB to 1 GB.
- Compile your own X servers, write your own CDs and the items mentioned above: 4 GB.
- Set up an Internet/FTP server: 400 MB minimum.

How should you divide up the hard disk? This simple question cannot be answered easily. If you do not want to partition these simple rules apply:

- up to approx. 550 MB: swap partition and a root (`/`).
- more than approx. 600 MB: swap partition, root (100 MB) and the rest for programs (`/usr`).
- more than 750 MB: swap, root (100 MB), home (`/home`) app. 100 MB and the rest for programs (`/usr`).

> If you plan to start Linux directly from the hard drive, you will need a Linux partition below the "1024 cylinder limit" as a boot partition (cf. section 8.3.4, page 125 and section 8.7.3, page 138). This doesn't concern you if you will start Linux from DOS/Windows with **loadlin**. Most of the time this boot partition will be the same as the root partition.

Furthermore you should consider that some programs (mostly commercial programs) install their data under `/opt`; just in case, please, either provide for `/opt` an own partition or make the root partition big enough. Examples:

- **KDE** (30 MB)
- **Wabi** (10 MB)
- **Netscape** (35 MB)
- **Applix*ware*** (300 MB)
- **Staroffice** (130 MB)

If you don't want a separate partition `/opt`, you may use this "work around":

There's enough space anywhere else in your system, e. g. under `/usr`; create a directory (e. g. `opt`) and make a symbolic link:

```
root@earth: # cd /
root@earth:/ # mkdir /usr/opt
root@earth:/ # ln -s /usr/opt /opt
```

If you plan a bigger configuration than 1 GB it is recommended that you read further. For swap partitions please see section 3.29.1.

3.29 Partitioning for experts

Partitioning has already been outlined in section S1 and section S10. This section should provide more detailed information for tailoring a system that best suits your needs. This section mainly is important for those of you who want an optimized system as far as security and performance are concerned.

It is necessary to have a profound knowledge of the functions of a Unix filesystem. The topics ☞*mountpoint*, physical, extended and logical partitions should not sound Chinese to you.

First: There is no golden mean for all but lots of tiny golden means for anybody. Do not panic we will give you some real numbers as hint.

First gather the following information:

- What is the purpose of the machine? (Fileserver, compute-server, stand-alone machine)
- How many persons are going to work with this machine? (simultaneous logins).
- How many hard disks are installed? How big are they and which system? (EIDE, SCSI or even RAID controllers).

3.29.1 Size of swap partition

Quite often you will read:

"☞*Swap* should at least be quite as large as physical RAM". This is a relic of times when 8 MB was regarded as a lot of RAM. Well, times sure have changed. Those who purchase machines with less than 16 MB of RAM have not been counselled well. Lets come back to the statement mentioned above. It was our aim that the machine will have app. 30 to 40 MB of virtual ☞*memory*.

Using modern applications that need quite an amount of memory these values have shifted up a bit. Generally 64 MB of virtual swap should be sufficient. Do not be stingy. If you compile a kernel under X and want to have a look at the manual pages using **Netscape** and have an **emacs** running will already take up all your 64 MB.

For being safe, opt for at least 96 MB virtual memory. One thing you should never do is not to assign swap space at all! Even on a machine with 256 MB RAM there should be a swap partition. The reasons are described in section 3.29.3

You plan to run extensive simulations and need some giga-bytes of memory(!). In case of doubt as to whether Linux suits your needs please read section 3.29.2 (Compute-Server).

3.29.2 Appointment of the machine as stand-alone machine

The most common use for a Linux machine is a stand-alone machine. In order to make decisions for you as easy as possible we now provide you with some concrete figures which you can use at home or at your company. In table 3.1 there is a overview of different installation volumes for a Linux system.

Installation	Needed disk space
minimum	80 MB up to 200 MB
small	200 MB up to 500 MB
medium	500 MB up to 1 GB
big	1 GB up to 3 GB

Table 3.1: Examples for installation

Of course these values increase as you want to save additional data volumes.

Example: Printer server/router

Let's assume that you do not want to throw your old 386 SX 20 with is 80 MB hard disk away. Separate the disk into a 16 MB swap partition and the rest (64 MB) for root `/`. This little machine may serve as a firewall or as a gateway to the internet. Of course it should only be used by root.

Example: stand-alone machine (small)

You have app. 500 MB spare hard disk that should be used for Linux.: 100 MB for root `/`, a 32 - 40 MB swap partition and the rest on user `/usr`.

Example: stand-alone machine (average)

There is 1 GB available for Linux. As above 100 MB for root `/`, 64 MB for swap,100 MB for `/home` and the rest for `/usr`. Note, that the RPM database under `/var` occupies several MBs (cf. section 5.3.2, page 96)!

Example: stand-alone machine (luxury)

If you have more than 1 GB available there is no overall solution. Please read section 3.29.3.

Example: File server

Here hard disk performance *really* is crucial. You should use SCSI devices if possible. Please also keep in mind the performance of the disk and the controller.

A file server is used for centrally saving data. These might be ☞*home directories*, a data base or other archives. The advantage is an easy administration.

If the file server should serve a huge net (from 2o users onwards) optimizing hard disk access is essential.

Supposed you want to provide a file server for 25 users (their home directories). You know that every user will have 80 MB of personal space. If not all of them compile the whole day a 2 GB disk which is mounted under `home` will do.

If there are 40 users mathematically a 4 GB disk is necessary. Anyway it is better to split `home` onto 2 2 GB disks as now they share the load (and access!).

As a compute server:

A compute server generally is a powerful machine that carries out extensive calculations over the net. Normally such a machine is equipped with extensive main memory (from 256 RAM onwards). The only point where fast disks are needed is for the swap space. Same applies here: separate the swap partitions onto separate disks. Furthermore keep in mind that Linux can only handle 128 MB swap partitions but anyway can handle 8 of those[5]

3.29.3 Optimizations

Normally the disks are the bottle-neck. For avoiding this there are two possibilities which should be used together:

- Separate the load on multiple disks
- Equip your file server with enough memory (at least 64 MB)

Parallelising multiple disks

This needs some further discussion. The total amount of time needed for transferring data can be separated into four parts:

- Time elapsed until the request reaches the controller
- Time elapsed until this request is send to the disk
- Time elapsed until the hard disks manages to set its head
- Time elapsed until the media has turned to the right sector
- Time elapsed for transferring data

Item 1 depends on the network connection and has to be regulated elsewhere. We do not to cover this right here. Item 2 can can be ignored; this depends on the controller. Item 3 is the vital part. The time is counted in milli seconds. Relative to the access time of main memory (that is measured in ns) this is

[5] slight modifications provided even 64

a factor of 1 million! Item 4 depends on the disk rotational speed, item 5 depends on this speed and the number of heads and of the actual position (inside or outside).

For optimized performance one should consider item 3. Here the SCSI features “disconnect” comes into action. Lets have a look at what happens:

The controller sends the command (in this case to the hard disk) “Go to track x, sector y” to the device. Now the mechanic of the disk has to start-up. If this is an intelligent disk (if it supports disconnect) and the driver itself also is able of doing disconnect, the controller sends a disconnect and the disk separates itself from the SCSI bus. Now other SCSI devices can do their task. After a while (depending on the strategy or load on the SCSI bus respectively) a connection to the disk is reestablished. Normally the device now has reached the demanded track.

On a multitasking-multiuser system like Linux there are lots of optimizations that can be done here. Let’s have a look at an output of the command **df** (comp. screen output 3.29.1).

```
Filesystem         1024-blocks  Used Available Capacity Mounted on
/dev/sda2              45835   27063    16152     63%   /
/dev/sdb1             992994  749694   192000     80%   /usr
/dev/sdc1             695076  530926   133412     80%   /usr/lib
```

Screen output 3.29.1: Example of a partitioning: Output using command **df**.

So, what benefits does parallelizing bring us? Provided we enter in `/usr/src`:

```
root@earth:/usr/src/ > tar xzf paket.tgz -C /usr/lib
```

Here **paket.tgz** should be installed onto `/usr/lib/paket`. For doing so the shell launches **tar** und **gzip** (these are located in **/bin** thus on `/dev/sda`), then `paket.tgz` in `/usr/src` is read (is on `/dev/sdb`). At last the extracted data is written to `/usr/lib` (under `/dev/sdc`). Now, positioning as well as read/write of the disks internal buffers can be activated at the same time.

This is only one example; there are many more. As a rule of thumb, if there are many disks (with the same speed), `/usr` and `/usr/lib` should physically be placed on different disks. Here `/usr/lib` should have approximately 70% of the capacity of `/usr`. `/` due to its access it should be placed on the disk containing `/usr/lib`.

From a certain number of SCSI disks onwards (4-5) one should consider buying a RAID controller. Thus operations on the disks are not only quasi-parallel but parallel. Fault tolerance is one of its famous side-products.

Processing speed and size of main memory

We want to point out that size of main memory is more important under Linux than the processor itself. One reason[6] is Linux’ ability of dynamically creating buffers of hard disk data. Here Linux uses lots of tricks as “read ahead” (gets sectors in advance) and “delayed write” (saves writes until there

[6] if not the main reason

is a bundle to write). The latter is the reason why you should not switch off your Linux machine. Both items are the reason why Linux is so fast and why the memory seems to fill rapidly. Linux works with shared libraries, that is lots of programs and applications share the same library. This leeds to the fact that a library call has only to be allocated once. Of course this also means that your memory fills up rather quickly. So if you do not know whether to purchase another main board or some more memory we recommend to do the latter as it increases the speed of Linux.

```
             total      used      free    shared   buffers    cached
Mem:         63304     62312       992     15920     38692      4200
-/+ buffers:           19420     43884
Swap:       199508     14548    184960
```

Screen output 3.29.2: Output of **free**

If you want to find out what memory is used up to now just enter:

root@earth:/root > `free`

This will give you an overview of used memory and buffers. The screen output 3.29.2 shows that some 38 MB are saved in buffers. If you want to access data that already is buffered this data almost immediately is available.

Chapter 4

YaST – Yet another Setup Tool

YaST is a main part of S.u.S.E. Linux. It helps you to install and maintain your software (installation, deinstallation) and to administrate your system.

This chapter will cover YaST's main features to make the installation as easy as possible.

> There is a relative new YaST on S.u.S.E. Linux 5.2. Thus the screenshots in this book may differ from those you see. Furthermore some of YaST´s new functions are not mentioned.

4.1 Control and keyboard mapping

Control of YaST is mainly achieved by the cursor and tab keys. You can move throughout the list using the cursor-keys as well as Page ↑ and Page ↓ and accept your selection with Enter or ↩ respectively. When encountering *Yes/No* or entry-fields you can jump to and fro using ↦ . Esc always leaves the current menu. If you want to keep the settings you made you need to press F10 .

On some rare occasions (e. g. when starting YaST from a non-Linux terminal) you are not able to use the function keys since the codes are misinterpreted by the terminal. In this case you should use Strg + f <number> for simulating the function keys F<number> . F10 can be used instead of Ctrl + f 0 . F1 and F12 are not used under YaST.

4.2 Installation media

This menu (figure 4.1) lets you choose the installation media.

Normally you would select `'Installation via CD-ROM'`. `'Installing from HD'` allows you to install even if your CD-ROM is not recognized by Linux. (section 4.4) `'Installation via NFS'` enables you to install Linux over a network, meaning your computer doesn't have a CD-ROM of its own but is connected to a machine (via ☞*ethernet*) which has a supported CD-ROM. Please refer to section 4.5 for further information.

4.3 Installation from CD-ROM

If you plan to install via your CD-ROM please specify your CD-ROM.

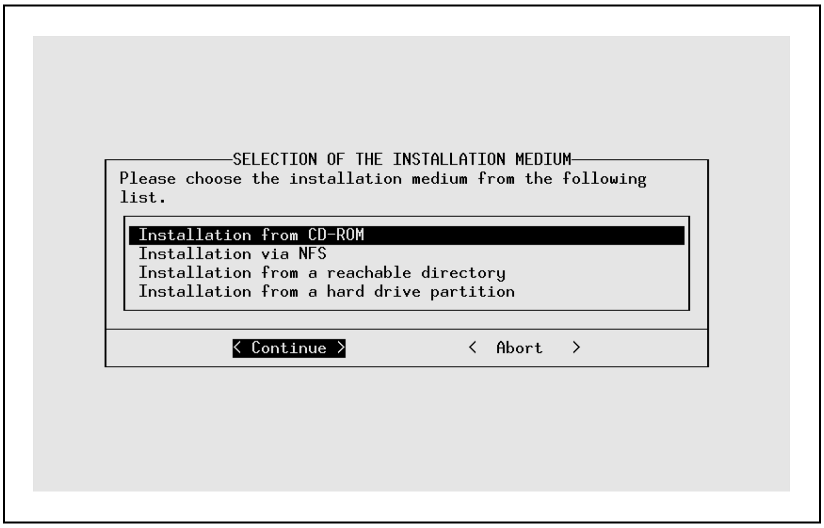


Figure 4.1: Selection of installation media in YaST

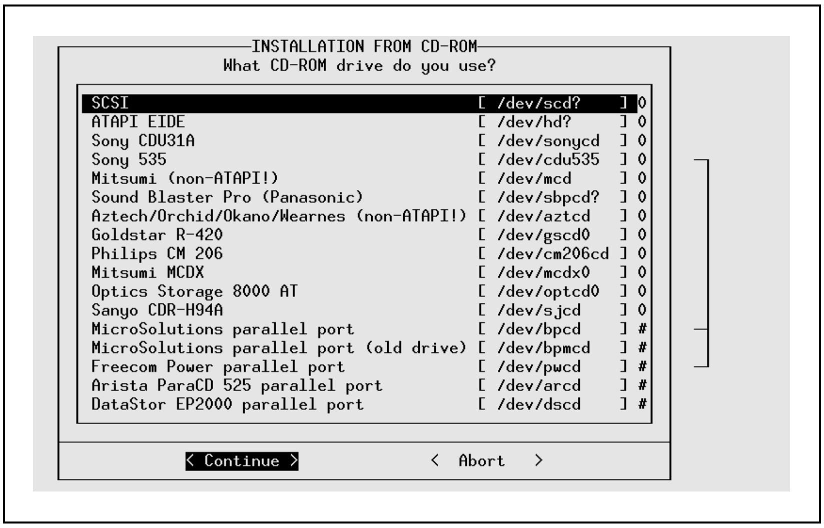


Figure 4.2: Selecting an CD-ROM drive

There are different drivers for Mitsumi drives! Those mentioned in the above list as “mitsumi drivers” are designed for the “old” Mitsumi CD-ROMs which are connected to a separate controller (e.g. LU-500 or FX-001). New Mitsumi drives (e.g. FX-400) are ATAPI drives and therefore you must select ATAPI EIDE. Same applies to Sony and Aztech drives.

Mitsumis mcdx driver differs in so far as it is able to read multisession CDs. Therefore it is rather meaningless for installation which one you choose. We have decided upon putting both for selection because there might be cases where one of them works and the other doesn’t. So now you can choose either of the two.

4.4 Installation via HD partition

If your CD-ROM is not supported by Linux, you still can install Linux on your computer by choosing another installation media. YaST expects a certain file structure which you have to "mirror" on your MS-DOS - OS/2 (can be *HPFS* (☞*filesystem*)) drive. Since you almost certainly don't have enough space to mirror the whole CD you have to decide which packages you want to install in each step. You can install as many packages as fit on your HD temporarily. After installation of your base system you can add other packages, one after the other.

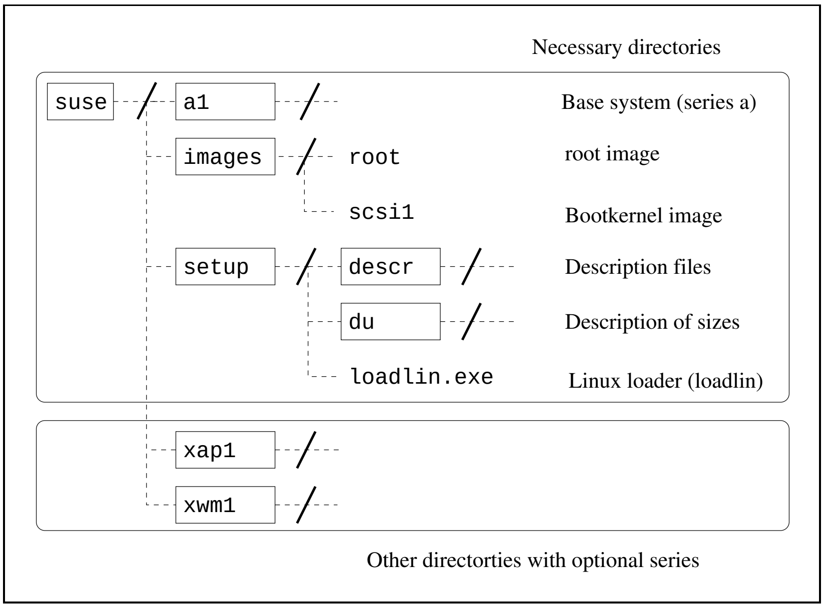


Figure 4.3: Directory tree for installation

Our example shows you how to install a base system via an MS-DOS HD partition. Here directories and file structures are given and cannot be changed.

All directories belonging to series A (`a1`) are needed as well as directory `setup` including all subdirectories. Directory `images` has to contain the kernel that you plan to use (double! one with the `.ikr` extension and one the kernel itself; file with the .inf extension contains a text number enabling YaST to show a description of this specified kernel and should be copied along with the other files. Configuration of a specific kernel can be found in the "README" file which resides in the same directory).

If you want to start from MS-DOS you additionally need the `root` file under the subdirectory `images`.

If you plan to start your installation system directly from MS-DOS, the directory `/suse/` is obligatory. Assuming that you have created a directory `suse` under the directory `C:\emil`, then you have to fill in the path `C:\emil` in

setup (without the `'\'` at the end!). Later on when YaST asks you what the source directory is you type `/emil/suse`.

4.5 Installation via NFS

Installation via network provides a means of easily installing multiple computers for experienced Linux users, even if only one of them is equipped with a CD-ROM or the sources are reachable over an ☞*NFS* mounted harddisk. You should be familiar with configuring an NFS server before you try installing via NFS. Even installation of a notebook which is equipped with a PCMCIA networking card is possible with this feature.

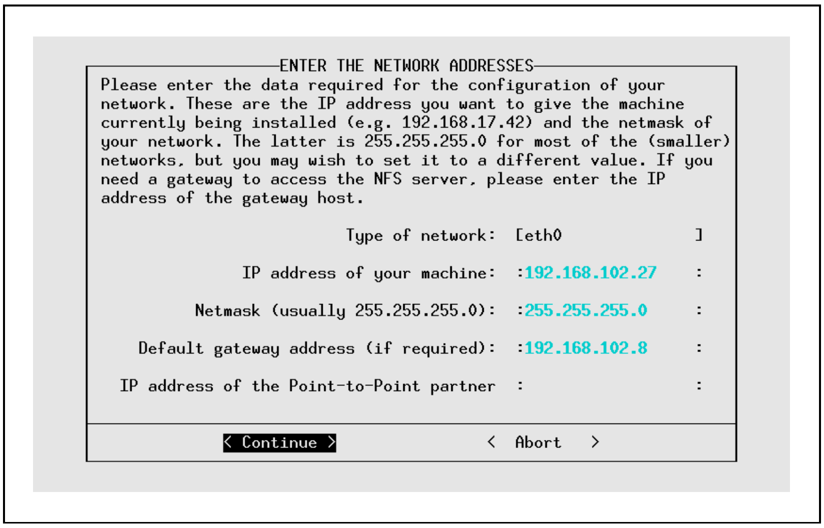


Figure 4.4: Entering network data for NFS installation

For installation via NFS computers need not to be linked via ethernet, but with others (e. g., parallel interface) as well. This feature mainly is interesting for laptops and notebooks. If you plan such an installation you should, of course, select a kernel with PLIP support, and you have to configure your PLIP interface before being able to start installation.

If you're directly connected to the NFS server the address being used matches the address of the PLIP partner and is exactly the same of your NFS server. The PLIP interface in most cases is just plip1; even the hardware parameters should not differ from standard for most cases.

Please be aware that while using your parallel interface as a PLIP interface you're not able to connect a printer to the same parallel interface. In most cases a printer connected to this interface will cause a permanent reset as soon as the driver starts!

The next mask asks you to give in the IP-address of your NFS server and the source directory. Of course the server has to export the named directory!

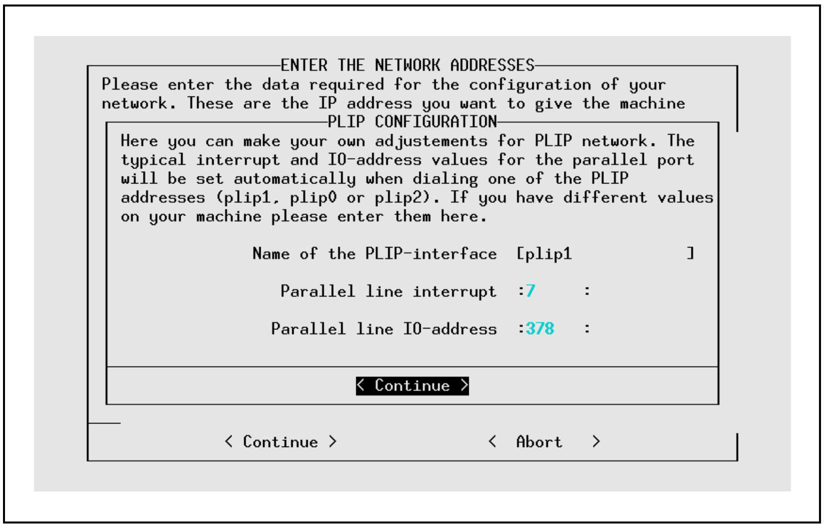


Figure 4.5: PLIP configuration

4.5.1 Installation via reachable directory

This option is generally used to install software once you have your Linux system running. In addition, it lets you install from drives for which no drivers existed when this CD was produced.

Now, to access this drive you start YaST. After having done so, change to another console (e. g.using Alt + F2) and login as `'root'`. Now you have to ☞*mount* your CD-ROM *by hand* to a directory, this can be done by (e. g.):

```
root@earth: # mount -tiso9660 /dev/cdrom /cdrom
```

See section 19.12.

Now you can perform a normal installation; just fill in (figure 4.6) the directory of the sources, which is the path where the CD-ROM is mounted and add `/suse` to it.

So, if you have mounted your CD-ROM (or your HD partition) to `/cdrom` you enter (as in figure 4.6) `/cdrom/suse`.

```
/cdrom/suse
```

4.5.2 Select the installation mode

Here you can choose between either “demo” or “normal” Linux installation. If you use “demo” mode Linux will be installed on top of an existing MS-DOS partition; of course this is only possible if there is at least one MS-DOS partition on your system!

Use of demo-mode is only available if there is at least one DOS partition on your system!

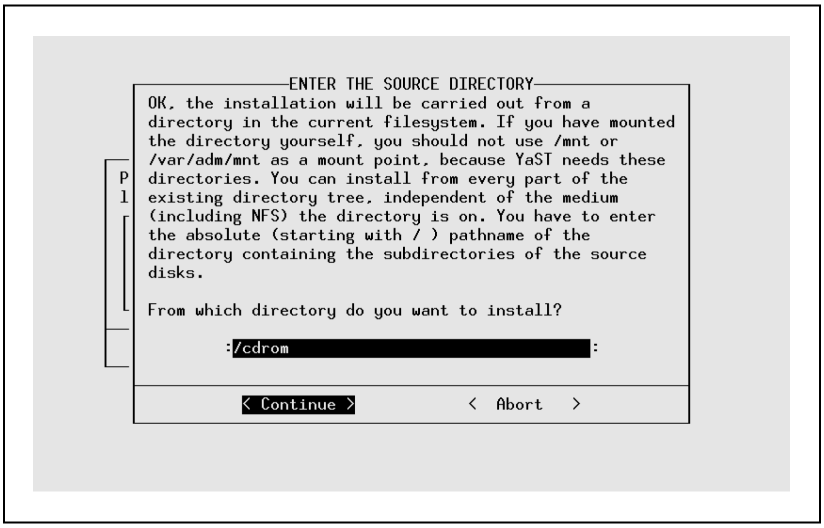


Figure 4.6: Entering the source media

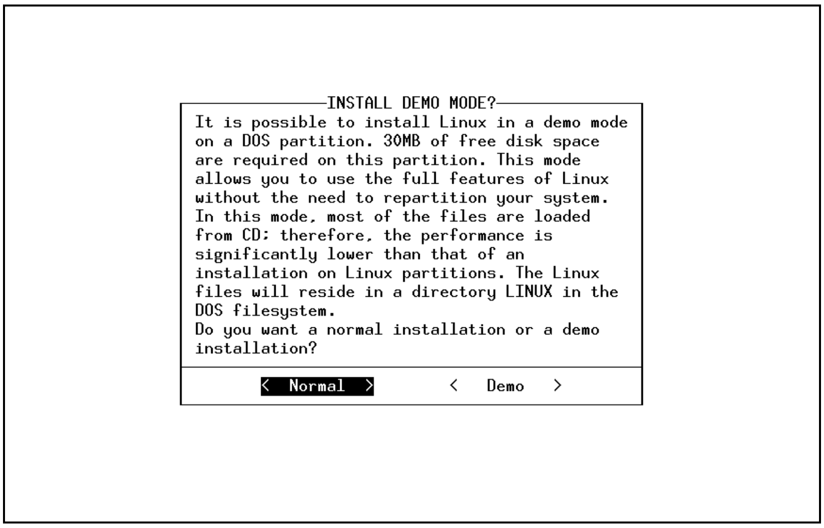


Figure 4.7: Select installation mode in YaST

4.5.3 The main menu

After having entered all the data absolutely necessary for the installation, you will find yourself in YaST's main menu. All previous settings can be changed at any time.

4.5.4 Settings

All installation relevant information is done in '`Setting up installation`'.

Installation can only proceed if you tell YaST where to mount your partitions. If your computer has very little RAM it is recommended to disable '`Reading DU-information`' (as it consumes rather a big amount of memory). This

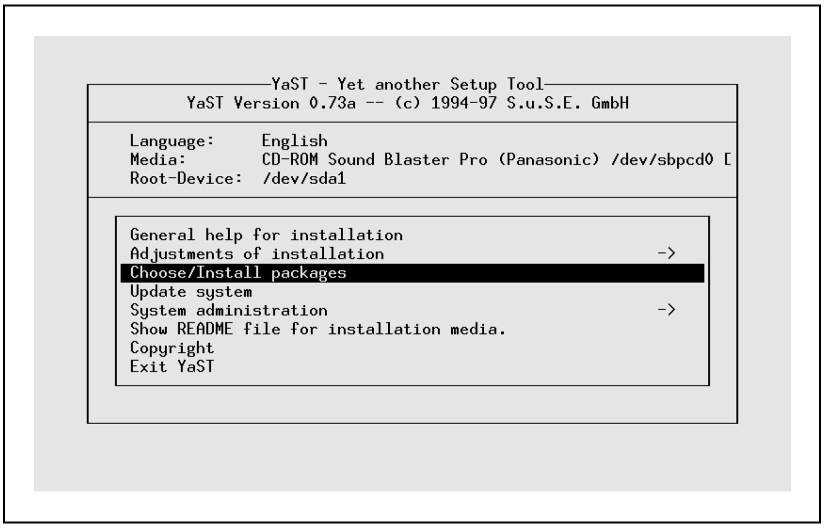


Figure 4.8: YaST's main menu

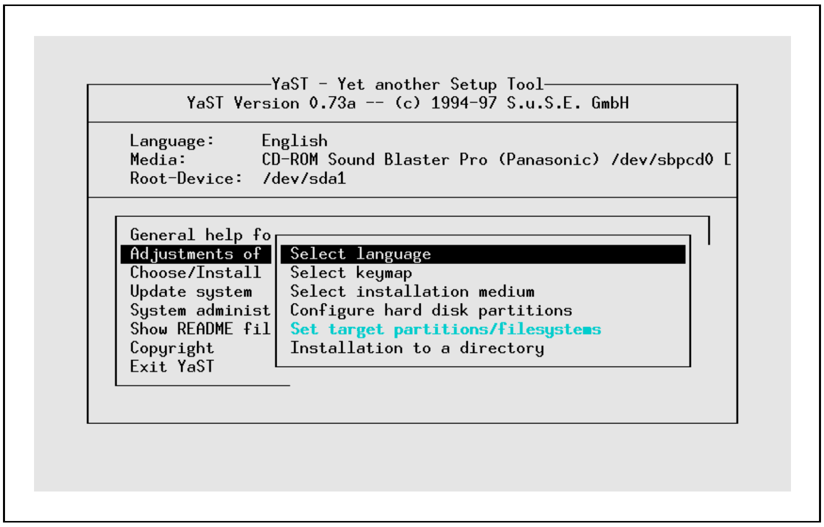


Figure 4.9: Menu '`Setting up installation`'

also is a good choice if your computer is supplied with 8 MB RAM but you have not activated a swap partition yet.

4.5.5 Partitioning your HD

The most critical and error prone point while installing a new OS is partitioning your HD(s). Normally each OS requires you to use a partition of its own. Linux offers you the possibility of installing a *demo mode* on top of an existing MS-DOS filesystem but you should only take this into consideration if you plan to "experiment" with Linux. The performance of demo mode is much less efficient compared to a *normal* installation. Furthermore the system is not as stable as if given a partition of its own since there are no filecheck programs under MS-DOS and your Linux can be influenced by MS-DOS during runtime.

So if you want to give Linux some partitions of its own you should split your HD into a couple of partitions. Generally a relatively small partition is used as *root* partition thus reducing write access and resulting in a more stable and not so error prone system and quite large partitions for containing the rest of the system. So `/usr` on most Unix systems is a partition of its own. It can be convenient to use a separate partition for `/var` or `/tmp`. After all partitioning is more a philosophic question so there is no “ideal solution”.

There is one partition that you should create; this is a *swap partition* to increase your amount of *virtual memory* (☞*memory*).

It is possible to create a swap file but since every access to this file is done via the filesystem this reduces performance; this is important if you have little memory installed. Here a swap file is no alternative at all to a swap partition!

If there is more than one HD installed you will be asked which HD to partition leading to a menu where all your partitions (of this HD) are presented.

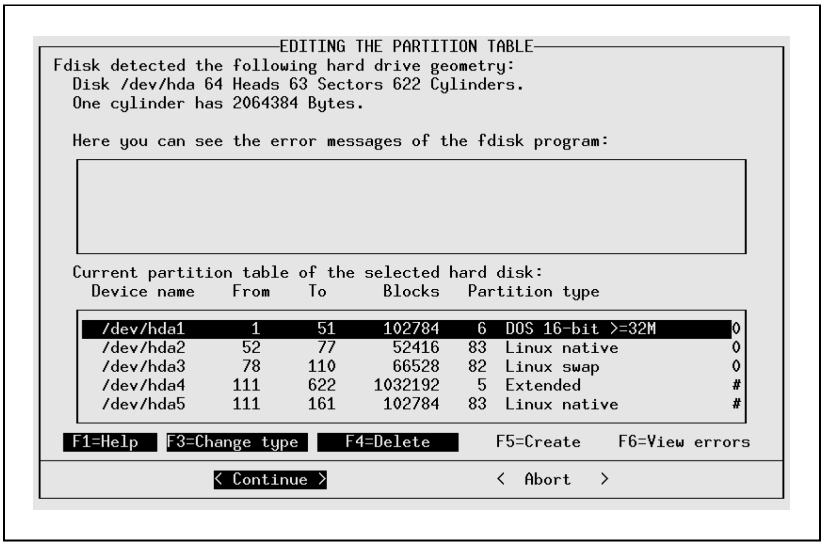


Figure 4.10: Partitioning with YaST

Using the cursor (↑ and ↓) keys you can move up and down through the partition list. F3 changes partition type, F4 deletes a partition and F5 creates a new partition.

Changing of partition type is necessary for creating a swap partition. Perhaps no partitioning is required if you have decided to use partitions momentarily used by another OS for Linux. In this case just move the cursor on to the desired partition and change the type by pressing F3 .

Be aware that Linux has no restrictions of where to put your system. It is possible to install Linux on a *primary* partition as well as on a *logical* within an *extended partition*.

Logical partitions have been introduced because in the partition table there is only room for four partitions; if you need more than four partitions at least the last partition has to be assigned an *extended* partition. Now you can create as many logical partitions in this extended partition as you like.[1]

After partitioning your computer has not to be booted at once but you can set the amount of software you want to install later on. Suppose you find out that your freshly created partitions are not big enough to hold everything you decided to install, you can go back and change the partition table until it is to your satisfaction. Only after quitting YaST the partition table will be written. Then you really have to reboot your computer immediately!

4.6 Assigning file systems

After partitioning your HD you have to link your freshly created partitions to directories in your Linux tree.
Select '`Set up your target partitions/filesystems`' under the topic *Installation settings*.

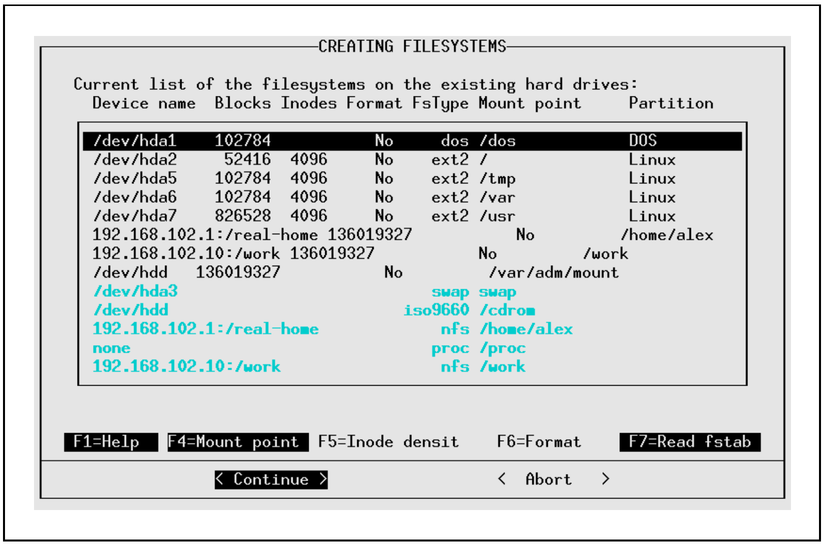


Figure 4.11: Assigning file systems

In our example (figure 4.11) you see the partitions of a system containing one HD. You can now decide for each partition if and how you want it to be formatted and of course the mount point in your Linux tree.

You have to assign one partition as *root* partition! This is called *root* as it is the root of all directories. That's why it is assigned the mount point (directory) `/`.

Manipulating your filesystems can be achieved by using the corresponding function keys. Needless to say that you can only manipulate Linux partitions. If the cursor is positioned on another filesystem than Linux all functions except mounting point are disabled.

[1] Extended as well as logical partitions are known to DOS as *extended DOS partition* or *logical drive* although this concept applies to any operating system

4.6.1 Mountpoint

F4 lets you choose where to mount your partitions.

You *have* to assign *one* partition as root-partition ((/)). You can mount your MS-DOS partitions to directories where you can find them later on easily (e. g./dosc for the first MS-DOS partition or /dosd for your second MS-DOS partition.

> Be aware that all mount points have to be given using absolute pathnames and that special keys for those directories are not allowed. You may not use /etc, /bin, /sbin, /lib, and /dev for your own partitions since those directories are reserved for system commands, system files, and contain the programs needed to mount other partitions!

There is one special thing about MS-DOS partitions! You can mount MS-DOS partitions in two different ways.

- One of them is the "normal" **MS-DOS** partition where all MS-DOS restrictions apply and
- the other is a **UMSDOS** partition. UMSDOS allows you, e. g.usage of long filenames, furthermore all Linux-specific properties such as owner, permissions and creating of links are possible under UMSDOS. Since MS-DOS doesn't provide you with the use of long filenames those are saved in special files which are located in each directory. Those files are named "–linux-" and should not be deleted at all, since otherwise Linux is not able to access its files! While running linux you will not see those files since they only contain internal information about the UMSDOS filesystem.

Since the DOS filesystem does not allow setting such special information they are located in special files which are located in each subdirectory. These files are called `--linux-.---` and should not be deleted under DOS, since in that case Linux will not be able to access them any more! Under Linux these files are invisible since they only contain internal information of the UMSDOS filesystem.

4.6.2 Inodes density

☞*Inode*density gives the average size of a file on a particular partition.

Number of inodes determines how many files can be created on a certain partition. If this number is too small it can happen that a partition seems to be full although there are still some blocks left. If you, e. g., take 4096 bytes per inode it means that on average each file has a size of 4 KB; now imagine you put only small files on this partition (about 1 kB) you can only use a quarter of your HD since your partition seems to be "full". 4 kB per file has proven to be an adequate size (more inodes per partition of course mean that there is less "netto-data-space" since inode tables also have to be written on this partition). If you plan to use one partition as spool directory , e. g. for netnews, a size of 2048 is enough since those articles normally are rather small. Another reason for choosing a large number of inodes is our **Live-File-System**. When using our live- filesystem approximately 40.000 files will be incorporated which use one inode per file.

4.6.3 Formatting

Pressing F6 you can tell if and how a partition should be formatted. With modern SCSI HDs checking for bad sectors is not really necessary but if you want to be sure you can select this option. Formatting and checking for errors of course takes quite an amount longer.

If you didn't change the partition table in this YaST session you can format your HDs immediately. Otherwise YaST saves your selections and formats after them (after you have rebooted). Partitioning takes some time depending on the size of the partition to be formatted.

4.6.4 Reading `fstab`

If you want to read an existing fstab you have to press F7 . You will be shown even those entries which belong to the HD's filesystem (`swap`, `proc`, `nfs-mount` etc.). Those entries are greyed out and can't be changed. Of course they will be kept if you decide to save your changed `fstab` partitions. This feature is needed if you do an update of your existing Linux system, since YaST needs to know where the system resides. It is possible to have several versions of Linux running parallel on your computer.

4.6.5 Size of installation

After you have finished configuration of your filesystem please go to '`Start / setup/installation`' to choose the amount of packages you are going to install. YaST provides a means of saving your own installation-profile or to load an existing one.

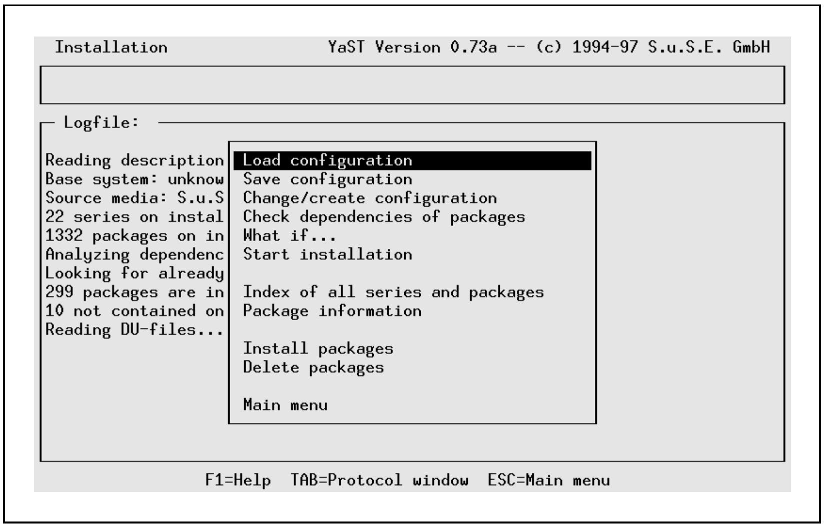


Figure 4.12: Assigning amount of installation

Furthermore you can start installation or make sure what would happen if you install the actual configuration.

4.6.6 Loading an existing configuration

There are several predefined configurations under `'Load configuration'`; you can choose any of them. Under several others there is a base system which you should use if you start YaST from disk. A minimal system gives you the opportunity of starting Linux from HD and to install more packages afterwards. Since YaST isn't loaded from disk, handling and performance is much more convenient. This applies mainly to computers with little RAM.

If you have designed (and saved) your own profile you can, of course, find it here and load it. This might come in handy if you're installing an identical configuration on several different computers.

If you have already installed a system you should be careful when you load a different configuration since all packages not belonging to this particular configuration will be marked for deletion. If you want to keep those preinstalled packages you merely answer `no` if presented the question whether you want to delete or not. Otherwise you have to deselect all packages marked with `'[D]'`; afterwards there should be an `'[i]'`.

4.6.7 Saving your configuration

Here you can save your own configuration. If YaST is started from disk this disk is used for saving your configuration; if you have booted directly from CD you will be asked to insert a pre-formatted disk; YaST will write your configuration to this disk.

4.6.8 Changing your configuration

If you select this option you will be presented a list; all packages on our CD are listed here. You can decide which package to install plus you can deselect previous installed packages and thus let them be deleted.

You can move up and down by using the cursor-keys or ↓ and ↑ , as well as Page ↑ and Page ↓ ;. ↵ opens the series for further details on packages belonging to this series.

In the lower window there is a description of the package under the actual cursor position; the right window shows you the actual percentage of your partitions being used.

You can leave this mask by pressing F10 ; then you'll be back in the configuration menu.

If you press ↵ on a series you'll be put into package selection of this series. If you have loaded a configuration before, the packages belonging to this configuration are marked with a cross.

Here you can see a short description of the package in the window underneath of the actual package (the one under the cursor). The right window shows you the percentage of your partitions.

Those values are being updated while selecting/deselecting a package. In some cases there is not enough room to show all necessary information. By

```
 Series selection           YaST Version 0.73a -- (c) 1994-97 S.u.S.E. GmbH

 a      1.CD  Linux Base System (You need it!)                       [ 47.6] 0
 ap     1.CD  Programs that don't need X                             [ 11.3] 0
 aplx   3.CD  ApplixWare Office                                      [  0.0] 0
 d      1.CD  Development (C, C++, Lisp, Perl, etc.)                 [ 31.6] 0
 doc    2.CD  Documentation                                          [  0.0] 0
 e      1.CD  Emacs                                                  [  0.0] #
 emu    3.CD  Emulators                                              [  0.0] #
 fun    1.CD  Games and more                                         [  0.2] #
 gra    1.CD  All about graphics                                     [  2.2] #
 n      1.CD  Network-Support (TCP/IP, UUCP, Mail, News)             [ 10.0] #
 pay    3.CD  Commercial Software                                    [  0.0] #
 snd    1.CD  Sound related stuff                                    [  0.0] #

device-name partition  total    used   free(MB) free%  mount-point

/dev/hda2     Linux       49.6     18.6     31.0   62%  /
/dev/hda1     DOS        100.2      0.0    100.2  100%  /dos
192.168.102.1:
/real-home    Linux      360.7      0.0    360.7  100%  /home/alex
/dev/hda5     Linux       97.2      5.0     92.1   94%  /tmp

                    F1=Help   F3=Partitions   F10=Esc=Exit
```

Figure 4.13: Selecting series in YaST

```
 Package selection  -  Series YaST Version 0.73a -- (c) 1994-97 S.u.S.E. GmbH

 [X] aaa_base  S.u.S.E. Linux base package                 0  Mount point
 [X] aaa_dir   S.u.S.E. Linux directory structure          0      Free (MB)
 [X] base      Collection of important GNU packages        0  /
 [X] bash      The GNU Bourne-Again Shell                  #           28.5
 [X] bash1     Old version of bash                         #  /dos
 [X] bdflush   Linux update daemon                         #          100.2
 [X] compress  standard un*x compression program           #  /home/alex
 [X] cpio      The GNU cpio backup and archiving utility   #          360.7
 [X] cron      cron daemon                                 #  /tmp
 [X] db        BSD database                                #           92.1
 [X] devs      Device files                                #  /usr
 [X] diff      GNU diffutils                               #          599.2
 [ ] eazy      Compatibility links                         #  /var
 [X] elvis     A vi-clone called elvis                     #           87.7

Version:        97.6.29
Peculiarity:    Base system
Package Size:   installed 436 K (compressed 143 K)
This package installs a few important configuration files. Central scripts
like SuSEconfig are also in this package.

        F1=Help   F2=Description   F3=Partitions   F10=Ok   Esc=Abort
```

Figure 4.14: Selection of packages, series `a1`base system

pressing F2 you can enlarge the lower window, by pressing F3 you can enlarge the right window.

Each package name is preceeded by it's current state:

'[ ]' indicates that this package is not yet installed

'[X]' indicates that this package will be installed

'[i]' indicates that this package is installed

'[D]' indicates that this package will be deleted

'[R]' indicates that this package will be updated (replaced)

By pressing ⟵ you can switch either between '[ ]' and '[X]' or between '[i]' '[R]' and '[D]'. Pressing F10 leads you back to the series selection menu; pressing ESC leaves without changes. If you, by any chance encounter problems with space of your partitions you can easily switch back to the main menu and repartition your disk. The actual partitioning will be written if you leave YaST. If you're running YaST from disk, only install a minimal configuration, boot your Linux and install the remaining packages. Of course you can try by selecting your "dream"-configuration to see how much space is needed.

4.6.9 What if..

YaST can also be used for deinstallation, therefore we created this menu so that you can see what will happen if you start installation. Here you can make sure that you didn't mark packages for deletion which are necessary for your Linux system to work properly.

4.6.10 Installation

This is the real installation. All marked packages will be read from CD, uncompressed and copied to their directories. If you have changed partition tables during this YaST session, installation is not yet possible since after changing partition tables your computer needs to be rebooted. Furthermore this modified data has not yet been written to partition table, this only happens after leaving YaST. During installation status messages will be shown in the lower window. The upper window informs you about the actual package to install. After installation you could jump back to the protocol window by pressing ⟼. That's for making sure everything went right. Here you can see if problems occurred.

4.6.11 Deleting packages

Selecting these item gives you an overview of the packages that are actually installed. This list includes "foreign" packages, these are packages not included on the S.u.S.E. CD. YaST can neither update these packages nor check for any dependencies.

The easiest way is to replace these packages by packages on your S.u.S.E. CD. Just select the packages you need to replace. A short description is displayed if you press F2 (as there is no complete documentation available). F10 deletes the marked packages.

Thereafter you may install the corresponding packages from CD.

Check package dependencies

YaST checks the dependencies of packages already installed and packages that are to be installed, minus the packages that are to be removed; or phrased differently, the packages that would have been installed if you had already selected the 'Start installation' option.

The following types of dependencies exist:

The package dependencies which were found are then displayed in a list.

`AND`	Means that if this packages is installed, all listed packages should be installed too. For example, if you install the compiler, you'll also need the include and lib files.
`OR`	Means that if this package is installed, at least one of the listed packages should be installed as well.
`EXCL`	Means that if this package is installed, then none of the listed packages should be installed.

Table 4.1: Package dependencies

Index of all series and packages

This option simply provides a list of all packages included on the CD. Those packages marked with an `'*'` are either already installed or are selected for installation. This option is helpful and allows you to get a quick overview.

4.6.12 Search packages

Sometimes you might wonder where the file is that you are looking for. You try to find out where it is located on your S.u.S.E. Linux CD.

There is a "package information" that makes it quite a snap to look for packages and/or files. You may either search through the installation medium or through the entire system (or both). YaST looks for this file and presents you a list of hits.

4.6.13 Install packages

This item is for installing packages. These might be either new or bugfgixed versions which may be downloaded from our FTP site `ftp.suse.com`. Well, you may even install packages you downloaded from the Internet or built yourself. A couple of formats are supported: Tar archives (`.tgz`) and RPM packages (`.rpm`, `.spm` and `.src.rpm`) as well as special designed patch files (`.pat`) that are available (if needed) on our FTP site.

This installation procedure is divided into three steps. (a complete instruction may be get by pressing F1):

- Selection of installation medium
- Selection of the packages offered
- Installation of the given packages

You may now select the package medium by entering `'Source:'` and pressing ⏎ . Here the following items are available: `'Directory'`, `'FTP'`, `'Installation medium'` as well as `'Floppy'`. You may need to change the default path (for directory and/or FTP). YaST will connect to the source medium as soon as you hit ⏎ . Thereafter you will get a list of the available packages. This makes it possible to install packages via the Internet. Just select `'FTP'`.

```
 Install patches.            YaST Version 0.83b -- (c) 1994-98 S.u.S.E. GmbH
Source: (FTP)            ftp.suse.com:/pub/SuSE-Linux/suse_update
[FTP to /tmp/ftp22412]

 dir    1.0K  Feb 10 03:35  S.u.S.E.-4.4.1                                   #
 dir    1.0K  Feb 10 02:24  S.u.S.E.-5.0                                     #
 dir    1.0K  Feb 27 12:15  S.u.S.E.-5.1                                     #
 dir    1.0K  Feb 10 03:24  XFree86-3.3.1-SuSE                               #
 dir    1.0K  Feb 18 12:17  XSuSE                                            #
 dir    1.0K  Feb 19 23:27  kernel                                           #
 dir    1.0K  Mar  5 03:05  misc                                             0
 dir    2.0K  Mar  3 19:40  special                                          0

 Logfile:
Connecting to ftp.suse.com[209.0.51.1]
Closed connection to ftp.suse.com[209.0.51.1]

          F1=Help   F2=Description   F10=Install       Esc=Escape
```

Figure 4.15: Install packages via FTP

. The default address is: `ftp.suse.com:/pub/SuSE-Linux/suse_update` (vgl. figure 4.15, page 82).

> If you get an error message as "`530 User ftp access denied.`" this means that there are too many users currently online. Just try later

You need to work your way through to the package you want to install. Now you can install this package as usual by marking it using the Space key. F10 accepts your selection. The package itself is copied to `/tmp/ftp`<`processID`>. Thus if something goes wrong you may install it manually. (see section 5.3.1, page 93).

4.7 Administration

Besides pure installational work YaST also helps you maintaining your system now that you are a brand new ☞*system administrator*. After installation (installing every package) has been done, there are still a couple of things to be done for adjusting your system to your computer and your personal needs. You can reach system administration via '`System administration`'.

4.7.1 Integrating hardware into your system

This is designed for exactly specifying your hardware. In most cases a *symbolic link* (☞*Link*) from a standard device to your personal hardware will be created allowing you to access your hardware whenever you want without having to know the device's name.

Adjustment of mice and CD-ROMs is quite easy, whereas configuring your printer can be rather tricky.

Configuring a printer is much more demanding.

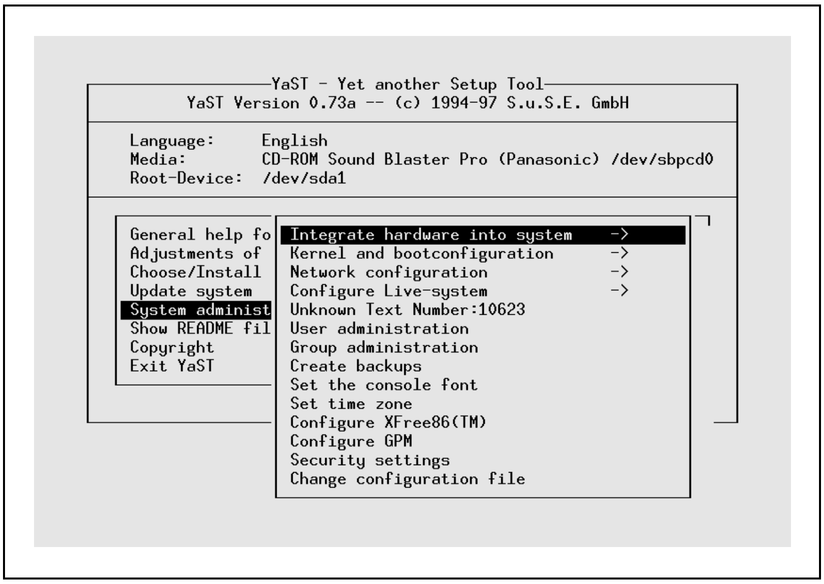


Figure 4.16: Administering your system

Configuring your printer

Accessing a printer under Linux is not as trivial as it seems to be. Luckily with **apsfilter** there is a powerful filter which automatically detects the file type, converts it and sends it to the printer.

General (like all other Unix) PostScript plays a central role under Linux. Printing a PostScript file on a PostScript printer is rather easy but since those printers are quite expensive only a few users call one of them their own. That's what we use **GhostScript** (`gs`) for, a free software that interprets a PostScript document and converts it into a form readable by the printer.

apsfilter even converts ☞*ASCII* files into *ghostscript* files for printing them directly on a PostScript printer or converting them into a format readable by the printer (programmGhostScript).

YaST offers the possibility of easily installing and configuring **apsfilter**.

If your printer is capable of printing colours you just activate the corresponding flag. In `'Printer type'` you type in the type of your printer and whether it is a PostScript printer or not.

`'Printer name'` shows you a list of the most common (and supported) printers. You can scroll through the list and choose a printer.

If your printer is not mentioned here you can try a similar one or the ancestor model (e. g.if you have LaserJet 5L just take LaserJet 4).

`'Paper size'` usually is `A4` . If your printer is connected to a serial ☞ *interface* you can adjust the baud-rate as well.

In general printers are connected to the parallel port, meaning that you have to enter the user interface. For most cases this will be `/dev/lp1`; if your printer is connected to the first parallel interface. `/dev/lp2` is only for

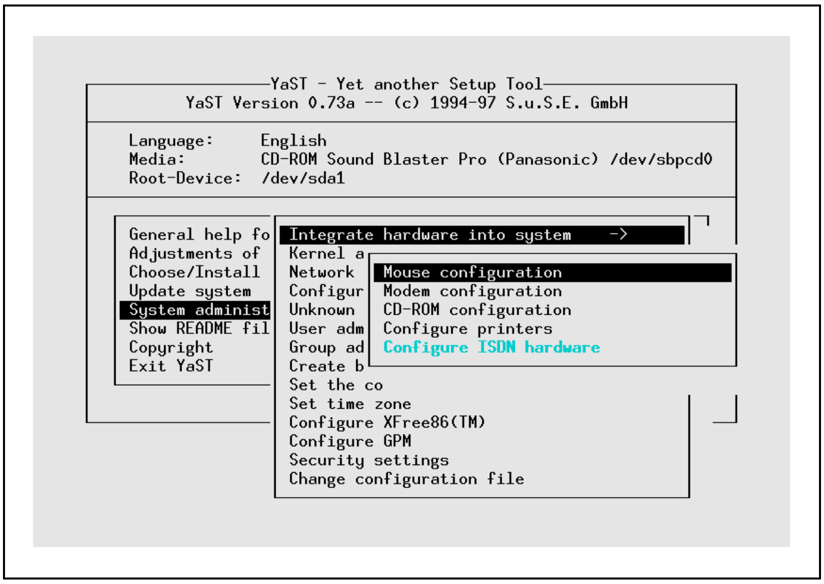


Figure 4.17: Hardware configuration

printers on the second parallel port. `/dev/lp0` is the parallel port on a **Hercules graphic device**.

4.7.2 Kernel and boot configuration

Here you can adjust and configure your system's boot setup.

You can install either of the precompiled kernels (from S.u.S.E. CD), if you notice your first choice wasn't optimal.

Anyway it is recommended to build a customized kernel which is perfectly adapted to your system; this kernel (see chapter 13)will be much smaller and faster.

Moreover you will be given the possibility of creating a boot-disk for your system if you haven't done so already. This is highly recommended since you will be able to boot your entire system even if something goes wrong while installing LILO or you can't boot your system for another reason.

> If you have Linux already installed and want now to install Windows 95 you really *must* create a Linux boot-disk! Windows 95 assumes that it has the computer to itself and therefore overwrites the **Master Boot Record** *without* any hints or warnings!

4.7.3 Configuring LILO

YaST also serves as a frontend for correctly configuring LILO. A detailed description of all entries and options is to be found in chapter 8. We recommend reserving an additional entry in `/etc/lilo.conf` (e. g.linux.old) and to activate the entry **`kernel optional`**. If you recompile your kernel (see

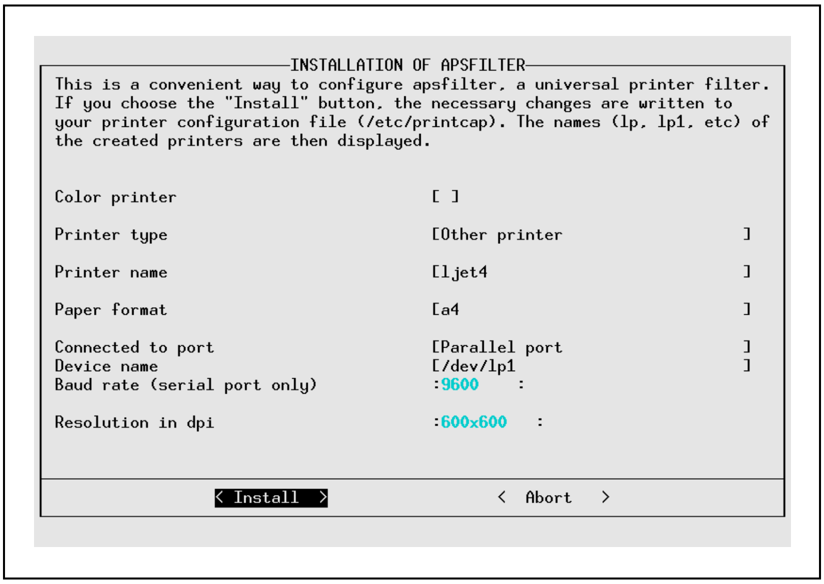


Figure 4.18: Setup **apsfilter** with YaST

chapter 13) a backup of your old kernel will be saved allowing you to always start the older kernel if anything should happen to the new one.

4.7.4 Configuring your network

General networking configurations can be done by YaST. Do configure your network in any case since there are a lot of programs and applications that require you to have a correctly working network; even if you're not connected to a real network at all.

General configuration of a network mainly consists of assigning an ☞*IP-Address* and a name; furthermore it is possible to add one or more name-servers and to install a **sendmail** configuration file which should be adequate for most cases.

During configuration you will be asked whether you only want **loopback** installed or a complete TCP/IP network, If you don't have an ethernet card installed you should answer '`yes`'; this also spares you further unnecessary configuration.

More information about configuring networks is found in chapter 16.

4.7.5 Integrating / Releasing CD-Live-File-System

If you integrate Live-File-System links will be created for every package not already installed on your system. These programs can be easily started from CD without using disk-space. You should keep in mind since your CD-ROM will be mounted every time you boot, you *have* to leave the CD in your CD-ROM drive (mount point is `/S.u.S.E.`). If you want to mount another CD first boot with our live-filesystem and then unmount it using the command:

```
root@earth: # umount /S.u.S.E.
```

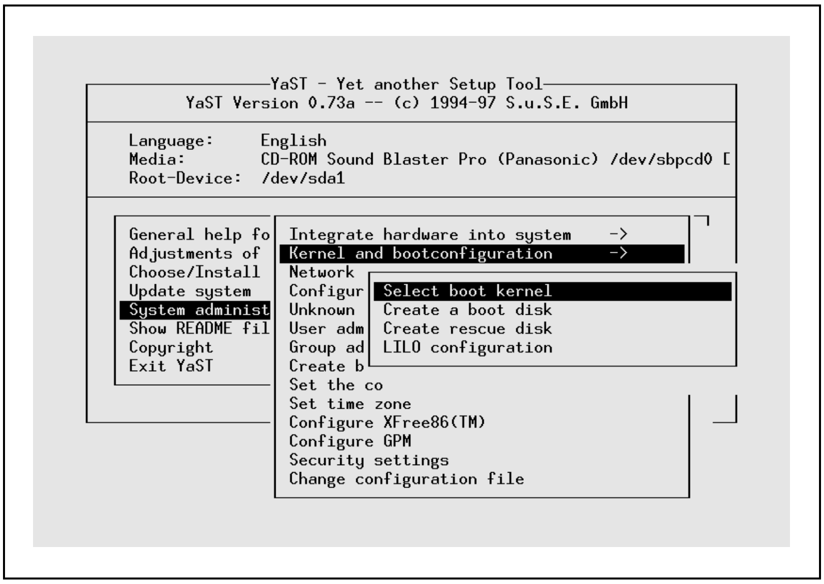


Figure 4.19: Kernel and boot configuration

and mount the other CD afterwards wherever you want to.

In some cases, even base programs are integrated from CD. In this special case those programs are constantly running and therefore you are not able to unmount the CD. Our base system is designed so that it is able to run without CD so just boot your computer with the CD removed. Linux will display a couple of error messages which you can ignore. All programs for accessing a CD are on your system.

If you receive an error message `"no space left on device"` this could be because the ☞*inode* density on your system is too small. With very small partitions this is the most common error. Unfortunately this can only be solved by reformatting and decreasing inode size (e. g.1 kB /inode instead of 4 kB /inode- for further information look at section 4.6.2).

If your HD is used up to extend (no space is left) you can't even start YaST for disintegrating live-filesystem as YaST can't create the files it needs for starting. Just delete some of the symbolic links pointing to `/S.u.S.E.`, hereafter starting YaST again.

4.7.6 User management

YaST provides a comfortable means of maintaining, adding, and deleting your users.

If you are an experienced Unix user it might be too much work to start YaST for a simple **useradd**; needless to say that you can use the standard programs for achieving the same result, such as **useradd** and **userdel**.

Figure 4.20: User management with YaST

4.7.7 Changing configuration file

S.u.S.E. Linux is maintained by a central configuration file (`/etc/rc.config`). This file is read at boot time by the bootscripts which configure your system accordingly.

There is an editor supplied with YaST to edit and change all these entries by hand, without having to know the exact syntax and what all these entries mean.

Move the cursor onto the entry that you want to change, press ⏎ or F3 to change its value. If you have edited this file by hand (without using YaST) you should run the script **SuSEconfig** to update the changes. This script reads `/etc/rc.config` and adjusts all program specific changes in its configuration files. A detailed description of all configuration files is to be found in section 12.6.

4.7.8 Creating backups

It can be quite helpful to backup all modified and new files and packages to a file or to a tape- these in most cases are configuration files. That's what this function has been designed for.

This dialog consists of three parts:

1. Adjusting the amount of files being backuped
   Here you can tell YaST which directories should not(!) be updated (should be excluded). Predefined are `/tmp` `/dev` and `/proc`; you should add mounted CD-ROMS or NFS-mounted filesystems to this list. The less you want to be backuped the faster it runs since unnecessary comparisons with package lists are omitted. Using + and - you can add new directories or remove them. Pressing F10 leads to the next step.

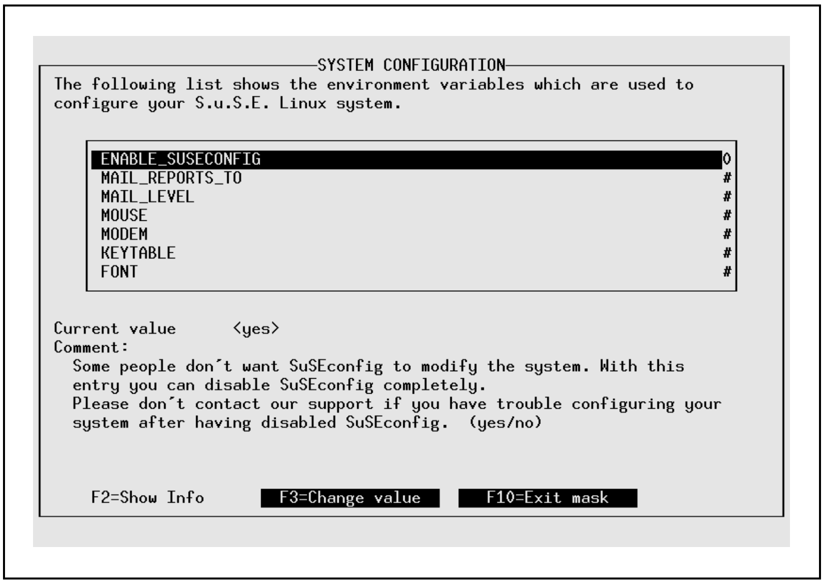


Figure 4.21: Editing the configuration file with YaST

2. Searching

   Now YaST searches for files which should be backuped. The amount and size of packages found are updated while searching. After this has been done there will be a list with all the files that have been found. Here you still can deselect files by using the SPACE key.
3. Entering the command Here you decide how those files are going to be saved. You can give archive names, options and more.

Of course this backing-up mechanism can only work if the dates of the files have not been changed otherwise. Furthermore this function requires quite an amount of RAM. Filenames of an ordinary CD take up to 6 MB of RAM; of course you need enough free disk space to save the archive. Compressing the archive will lead to a file reduced in size (approximately half of the original). The best way to do backups is a tape anyway.

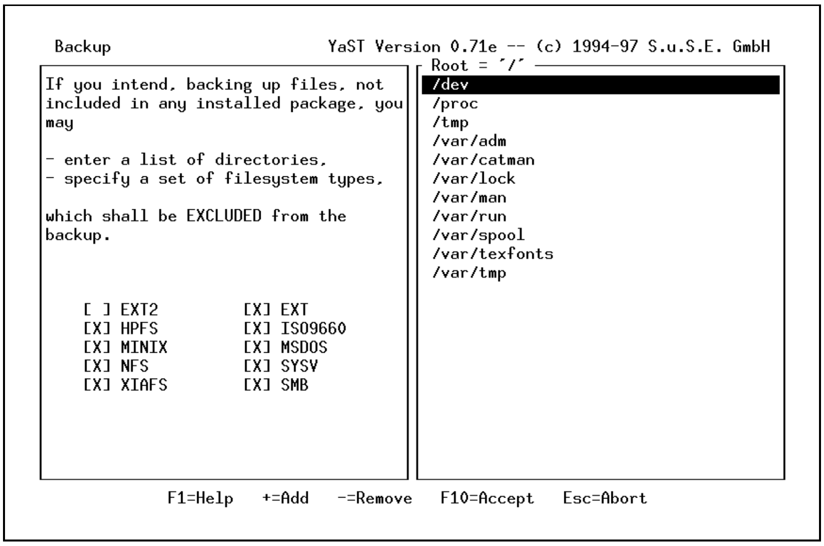


Figure 4.22: Backup with YaST

Chapter 5

Updating the system and package management

5.1 Updating S.u.S.E. Linux

S.u.S.E. Linux offers you the option of updating an existing system without having to reinstall everything. Here you have to distinguish between an update of one or more packages or of the whole system (perhaps because your older system still is **a.out**).

5.1.1 Updating base system

Since updating changes all central parts (e. g., libraries) of the system this can't be done while the system is up and running. That in particular means that you have to start an update like a first-time installation (see section 3.15).

Under `'Installation settings'` you choose `'Assigning partitions / filesystems'` and specify your root partition; YaST reads the existing `/etc/fstab` and mounts your filesystems.

After going back to the main menu you should choose the item `'Updating your system'` (section section 4.5.3). Now your old system is analyzed by YaST. If it still is an **a.out** system YaST automatically converts it into ☞*ELF*.

After that all central configuration files of the system are updated; of course YaST creates backup files of those files that have been changed since the last installation.

If the base system has been updated you are lead to YaST's special update-mode where you can decide which packages to update.

YaST offers two lists; first, those packages YaST has recognized and decided that an update could be useful. The second list shows you a list of packages where this is not as simple (perhaps the old package did work without any information about their version).

In those two lists you can freely decide which package should be updated. If you start the update all selected packages are replaced by the new ones saving all files that haven been changed since the last installation.

After having completed this task you should proceed as with a normal installation; perhaps you should install a newer kernel as the older one might not recognize the ELF format.

5.1.2 Update configuration

S.u.S.E. Linux uses an enhanced concept for configuring and updating your system (see chapter 12 und section 12.5); this is far more complex (than **Slackware**) but extremely powerful and makes future updates really easy to handle.

As automatically converting old configuration files is not possible, you have do some adjustments after having updated. You will be rewarded for doing so with a very flexible configuration system which enables you to change system wide administrations without having to reboot your machine.

Furthermore a base for future updates is generated, as all settings relevant to the system are put into one single file (`/etc/rc.config`).

5.1.3 Updating of single packages

With S.u.S.E. Linux you can update single packages whenever you want. In YaST's package list (see section 4.6.8) you can move around as you please. If you select a vital package which is necessary for the system to run you will be warned by YaST; such packages should be replaced only in update-mode. For example, lots of packages contain *shared libraries* which might be in use when you want to run an update. Updating a running system will inevitably lead to a malfunctioning system.

5.2 From version to version

In the following sections problematic details are listed which have been changed from release to release. In this overview changes concerning configuration files will appear or whether well know programs will show a surprising behavior. But only those things are listed which might arise problems for the administrator or for the user at their daily work. This list is probably incomplete; please, also consult the SDB – to be found in package `sdb`, series `doc` (cf. section 2.1.3, page 12).

5.2.1 From an ealier version to 4.x

The **init** scripts in `/etc/rc.d` which used to bring up the system have been replaced by a system that is able to organize the different runlevels in accordance to **System V**. Since the file system standard does not allow executables under `/etc`, the **init**-scripts have been moved to `/sbin/init.d`. Automatic conversion of old configuration files isn't possible in all case; therefore you probably will have to edit some files by hand. The goal ist, that from now on system relevant configurations could be done without booting.

Since all vital settings will be stored in a system wide config file (`/etc/rc.config`) coming updates will be easy to perform.

5.2.2 From 4.x to 5.0

Problems and special issues:

- Package management changed from `TGZ` to `RPM` (cf. section 5.3).
- News **Bash**.
- **`startx`** is not started in the background any more.
- To start `Samba` set the variable `START\_SMB=yes` in `/etc/rc.config`.
- System relevant cron jobs are listed in `/etc/crontab` now.
- New group `'dialout'`; put all users in this group, who are allowed so execute "dialout" programs (**minicom**, **pppd**, etc.).
- The file system of the rescue system is modelled like usual layout of the running system.

5.2.3 From 5.0 to 5.1

Problems and special issues:
Nothing known right now.

- You will find the package description on the first CD in directory `/docu`.

5.3 RPM – The package manager

With S.u.S.E. Linux 5.0 **RPM** (**`rpm`**) ("Red Hat Package Manager") was introduced. This way life will become easier for all involved people: for the user, for the system administrator, and last, but not least for the package builders. With the mighty RPM database detailed information about the installed software packages is comfortably available.

`rpm` mainly operates in three different modes: compile installable packages from socalled pristine sources, install those binary packages resp. de-install or update them in a clean manner, query the RPM database or an RPM package. In the following sections the second and the third mode are described only[1].

Installable RPM archives are packed in a special binary format; these archives consist of (program) files to install and certain meta-information, which will be used by **`rpm`** to configure the software package or which will be stored in the database for documentation purposes. RPM archive normally have the extension `.rpm`.

5.3.1 Managing packages: install, update and de-install

Normally, installing an RPM archive is incredibly easy:

```
root@earth: # rpm -i <package>.rpm
```

With this standard command, the package will be installed only, if its dependency requirements are met and if it does not conflict with another package. With an error message **`rpm`** "cries" for packages which you also have to install to fulfill the dependencies. In the background the database takes care that no conflict will arise: a file can belong to one package only. With different options you can force the RPM system to ignore these defaults—but please, make sure that you know what you do: otherwise you will risc the integrity of the system and you might loose the feature to update the system straightforwardly!

[1] Concernig the first issue (build mode), cf. [Bailey, 1997] or manpage of `rpm` (`man rpm`).

Use the option `-U` resp. `--upgrade` to update a package. This way **rpm** removes the files of the old version and immediately installs the new files; at the same time it tries to handle the configuration files with care—slightly simplified:

- If a configuration file was *not* changed by the system administrator, **rpm** will replace the file silently; no action by the system administrator is necessary.
- If a configuration file was changed by the system administrator *before* the update, **rpm** will save the changed file with the extension `.rpmorig` or `.rpmsave` (backup file) and install the version from the new package; but only then, when the originally installed file and the new version are different. If this happens, it's probably necessary that you have to compare the backup file (`.rpmorig` or `.rpmsave`) with the newly installed file and to make your changed again. After doing so, please delete all `.rpmorig` and `.rpmsave` files, to avoid touble if the next update will take place in the future ... [2]

Obviously, the switch `-U` is more than just de-installing (`-e`) and installing (`-i`); please, use `-U` as often as possible.

> After every update you have to check all backup files created by **rpm**; these are your old configuration files. If necessary take over your customizations from the old `.rpmorig` or `.rpmsave` files to the new configuration files; after this process delete the files with the extension `.rpmorig` resp. `.rpmsave`.

To remove a package enter the command:

```
root@earth: # rpm -e <package>
```

rpm will only erase the package if there is no open dependency any more; this way e. g., it theoretically isn't possible to de-install an old `libc` as long as another program still needs it to work properly—the RPM database watches over it.

5.3.2 RPM queries

With option `-q` ("query") **rpm** initiates queries. With this option it is possible to inspect RPM archive (add the option `-p`) and to query the installed database. Several switches are available to specify the wanted information (cf. table 5.1).

`-i`	Summary information

Table 5.1: to be continued...

[2] `rpm` will choose `.rpmorig`, if the file was unknown for the RPM database until now—otherwise `.rpmsave`; in other words: `.rpmorig` files will be created while updating from a foreign format to RPM and `.rpmsave` while updating one RPM package with another RPM package.

`-l`	File list
`-f <FILE>+`	Query a package owning <FILE> (specify <FILE> with full path!)
`-s`	File list with state information (implies `-l`)
`-d`	Documentation files (implies `-l`)
`-c`	Configuration files (implies `-l`)
`--dump`	File list with complete details (to be used with `-l`, `-c` or `-d` benutzen!)
`--provides`	Capabilities the package provides
`--requires`, `-R`	Capabilities the package requires
`--scripts`	(De-)installation scripts (pre/post install/uninstall)

Table 5.1: The most important query options (`-q` [`-p`] ... <package>)

For example, the command

```
root@earth: # rpm -q -i rpm
```

displays the following information:

```
Name        : rpm                        Distribution: S.u.S.E Linux
Version     : 2.4.1                            Vendor: S.u.S.E. GmbH
Release     : 1                            Build Date: Wed Jun 18 14:46:53 1997
Install date: Sat Jun 21 12:01:21 1997     Build Host: Fibonacci.suse.de
Group       :                              Source RPM: rpm-2.4.1-1.src.rpm
Size        : 1365662
Packager    : feedback@suse.de
URL         : (none)
Summary     : Red Hat Package Manager
Description :
RPM is a powerful package manager, which can be used to build, install,
query, verify, update, and uninstall individual software packages. A
package consists of an archive of files, and package information,
including name, version, and description.
```

Option `-f` does only work, if you specify the complete file name (full path); specify as many file name as you want; e. g.:

```
rpm -q -f /bin/rpm /usr/bin/wget

rpm-2.4.1-1
wget-1.4.5-2
```

If you do know only a part of the file name, you will have to use a short shell script like this:

```
#! /bin/sh

for i in `rpm -q -a -l | grep  $1 `; do
    echo \"$i\" is in package:
    rpm -q -f $i
    echo ""
done
```

With the help of the database you can do verify checks; these checks will be initiated with the option `-V` (the same as `-y` or `--verify`). With this option

rpm will show all files of a package which are changed since the first installation. **rpm** knows eight character symbols to give some hints concerning the kind of the change:

`5`	MD5 check sum
`S`	File size
`L`	Symbolic link
`T`	Modification Time
`D`	Major and minor device numbers
`U`	Owner
`G`	Group
`M`	Mode (permissions and file type)

Table 5.2: Verify options

In the case of configuration files the character `c` will be printed; example, if you have changed `/etc/wgetrc` from the package `wget`:

```
root@earth: # rpm -V wget
S.5....T c /etc/wgetrc
```

The files of the RPM database are placed under `/var/lib/rpm`. If the partition `/usr` has a size of 500 MB, the database can occupy nearly 20 MB; especially after a complete update. If the database is much bigger than expected, it should help to rebuild the database with the option `--rebuilddb`; before rebuilding make a backup copy of the old database.

Furthermore the **`cron`** script **`cron.daily`** makes gzipped copies of the database and stores them under `/var/adm/backup/rpmdb`; the number of copies is controlled by the variable <MAX_RPMDB_BACKUPS> (default `5`) in `/etc/rc.config`. The size of one backup is approximately 2 MB (this value is valid for a 500 MB large `/usr` partition). Please, take this “waste” of space into account, if you make your decision concerning the size of the root partition; if you have an own `/var` partition, you don't have to worry about all these things.

5.3.3 Tools for RPM archives

The **Midnight Commander** (`mc`) is able to “browse” RPM archives and to operate on parts of them. This tool works on an RPM archive as if the archive would be a regular file system. This way you can view `HEADER` information with F3 and you can copy parts of an archive with F5. – By the way, right now **Emacs** doesn't have the facility to do similar operations `;-)`

xrpm is the name of a new graphical RPM manager; this tool is realized in the Python programming language. **xrpm** supports actions via FTP.

KDE comprehends the tool **krpm**, a graphical interface under the X Window System to deal with RPM. Looks nice, but at this moment is in the shape of an early development stage.

With **Alien** (**`alien`**) it is possible to convert or install an "alien" binary package. One can try to convert "old" TGZ archives to RPM before installing, this way the RPM database can keep track about such an package while installing. But attention: **`alien`** is a perl script and still is "alpha" software, as its author says, nevertheless it has reached version 5.17 (at the moment).

Last, not least, there is YaST . . .

Chapter **6**

Notebooks and PCMCIA cards

There are several PCMCIA adapters supported by Linux which makes it possible to use PCMCIA adapters on notebooks running Linux.

All common cards are supported, such as **Intel**, **Cirrus**, **Vadem**, **VLSI**, **Ricoh** and **Databook** chips. Even those special adapters found in **IBM** and **Toshiba** laptops and PCMCIA adapters found in desktops work.

For installation PCMCIA cards are only relevant if

- you install via ☞*NFS* using the PCMCIA networking card.
- you install via CD-ROM and the CD-ROM is connected via PCMCIA.
- you install onto a SCSI disk which is connected to a SCSI-PCMCIA adapter.

6.1 The concept

The PCMCIA support is a kernel-independent subsystem and therefore has not yet been included in the kernel itself. Further development aims at this; most probably PCMCIA support will be included in kernel version 2.2.

PCMCIA support is only available via modules (see section 13.2). For maintaining these modules there is a special program, **Card-Manager** (**`cardmgr`**), whose task it is to supervise the PCMCIA sockets and to load the appropriate modules on demand, or removing it respectively. From this point of view the Card-Manager can be considered a second kernel daemon (**kerneld**, see section 13.2.2). This concepts also enables you to change cards in a running system without any special precautions.

But it also has its disadvantages. Since it is not possible to build monolithic kernels with PCMCIA support there are some restrictions as far as hardware is concerned. Well, it is not possible to install Linux entirely on a SCSI disk which is connected to a net laptop via such an SCSI controller. To be precise: Installation is possible but booting the system is rather tedious since the SCSI driver's adapter can only be loaded after booting.

Use of **JAZ-Drives** on *parallel* port is not such a problem, since this driver can be compiled into the kernel. Well, you will not be able to use the parallel port for (e. g.) a printer (it's our aim to make this possible in future versions of S.u.S.E. Linux).

The only solution at the moment is a special form of RAM disk which enables the system to launch programs before booting the system (see section 13.4.5).

For addressing PCMCIA hardware you can load the appropriate modules before the "real" booting.

Realisation is a rather complex matter and can not be discussed in greater details in this chapter[1].

The special S.u.S.E.boot-disk uses this kind of ramdisk; so if you want to install the system onto a SCSI disk connected via an PCMCIA-SCSI adapter booting with this disk is the only choice you have. Since this is rather disturbing and system tasks like upgrading a kernel are an immense waste of time we highly recommend to install at least the root partition of an existing (E)IDE disk; the amount of space needed is approximately 20 MB.

6.2 Installation

as described above the S.u.S.E. boot disk (from S.u.S.E. Linux 4.4.1 or later) is able to load the Card-Manager in advance and to load the required modules. Hereby automatically recognizing the PCMCIA chipset.

For installing Linux on a laptop, this should be your first choice. In **linuxrc** you select '`Kernel modules (hardware driver)`' and then '`Load PCMCIA modules`'. **linuxrc** now tries to recognize the PCMCIA chipset, loads the PCMCIA kernel modules and starts the Card-Manager. The software now takes some seconds for analyzing the hardware and for loading the necessary modules (please be aware that for memory reasons only those modules are on disk that really are relevant for an installation; modules for modem cards are not included).

As soon as **linuxrc** announces that it has successfully launched the Card-Manager you can load the installation system directly from CD and commence "as usual" by starting YaST (as described in section 3.16). On the other consoles you can have a look whether all cards haven been installed correctly (e. g. using Alt + F3 or Alt + F3).

[1] If you are interested: Information can be found at `ramdisk.txt` and `initrd.txt` in the directory: `/usr/src/linux/Documentation`.

Chapter 7

Kernel parameters

7.1 Drivers in the kernel

There is a wide variety of PC hardware. Now, to access this hardware you will need the appropriate driver. The Linux kernel has to cooperate with this driver and your hardware. Generally there are two ways of integrating drivers into your system:

- These drivers can be compiled directly into the kernel. These kernels are referred to as *monolithic* kernels. Monolithic kernels e. g. are on our CD for creating boot-disks for rather exotic hardware. Some drivers have to be compiled directly into the kernel for accessing them at all.
- Drivers can be loaded into the system at runtime (on demand). This should be used since it reduces the amount of code being loaded at a given time; thus it saves a lot of memory. Our S.u.S.E. boot-disk kernel makes use of modules, thus being able to serve most hardware configurations.

  Some drivers do not exist as modules yet. Those are all drivers for EIDE-controllers, which therefore are compiled into our kernels on S.u.S.E.-boot-disk.

Nevertheless it can happen that a hardware component is not recognized. In such a case you have the advantage of passing parameters to the kernel for "helping" it.

With monolithic kernels parameters have to be passed by LILO (or **loadlin**); that's why they are normally referred to as LILO parameters. Modular drivers get their parameters via **insmod** resp. **modprobe** and start simultaneously.

Unfortunately the format of LILO parameters differs to **loadlin** parameters. That's why they are parted in two groups and shown separately. Some modules have generalized parameters (CD-ROM drives) where you can use the same parameters either with LILO or with **loadlin**

7.2 Some hints

Before we present you the list of parameters some hints on recognizing hardware of the driver, passing parameters and booting using the S.u.S.E.boot-disk:

- Most drivers can do an *autoprobing*, i. e. they test different standard addresses where this hardware usually can be found. By doing so it can happen that the driver hits an address that is not his but initializes it anyway. Then most certainly the machine will hang.

- Some modules can even be loaded successfully if their hardware is not installed at all (this mainly applies to 3Com networking card drivers.

  Anyway, you should try autoprobing first. Unused drivers can be removed easily. Hardware that is not automatically recognized must then be started using the appropriate parameters.
- Some driver are not available as modules. These are (e. g.) all EIDE drivers. So, if you only have EIDE components (hard disk or CD-ROM) you do not need to load modules when booting with S.u.S.E. boot-disk.
- At last there are some hardware components that do not only have one driver but plenty (**NCR 53C810**, **Ultrastor**). As far as we know there is no significant difference between these two Ultrastor drivers. The BSD driver for NCR53C810 also supports other NCR53C8xx-products (e. g. 53C875), whereas the old NCR driver is the only one that recognizes CD writers. Just try which one suits your needs.

7.3 The parameters

7.3.1 Notation and meaning

Below you will see an alphabetical list of kernel parameters and the corresponding devices. The following parameters always have the same meaning:

<addr>	hexadecimal portaddress, (e. g.0x300)
<irq>	Interrupt to access the device (e. g.7)
<dma>	DMA channel of the device (e. g.1)
<Start address>,	
<End address>	hexadecimal memory allocation for *shared memory*

Table 7.1: Frequently used variable names for kernel parameters

Now we mainly describe the parameters that are necessary for a successful installation. But there are other parameters for special aims and purposes. Please keep in mind that upper cases are significant!

A complete introduction to possible kernel parameters is to be found (after installation) under `BootPrompt-HOWTO` in `/usr/doc/howto`.

7.3.2 LILO-parameters

These listed parameters can only be entered at the LILO prompt (or via **loadlin**) to the kernel. If you plan to install this driver as a module please have a look at parameters in the next section (section 7.3.3).

> All parameters have to be entered directly one after another, separated by commas! There should not be a blank (whitespace) between the parameters!

- *Adaptec AHA-1520 / 1522 / 1510 / 1515 / 1505 SCSI-Host adapter*
  `aha152x=<addr>,<irq>,<id>[,<rec>[,<par>]]`

Variable	Values / Meaning
<**id**> (SCSI-ID of the host adapter)	`0, 1`
<**rec**> (reconnect)	`0, 1`
<**par**> (parity)	`0, 1`

This driver serves for running many low-cost SCSI controllers. All soundcards with a SCSI controller (except for **Pro Audio Spectrum**) have an adaptec chip installed and thus can be accessed.

For all not-original 152x the fourth parameter (**`RECONNECT`**) seems to be necessary. For most types it has to be set to `'0'`; only the AHA2825 needs `'1'`.

Example: **`aha152x=0x300,10,7`**

- *Adaptec AHA-1540 / 1542 SCSI-Host adapter*
  `aha1542=<addr>[,<buson>,<busoff>[,<DMA speed>]]`

Variable	Values / Meaning
<**buson**>	`2..15`
<**busoff**>	`1..64`
<**DMA speed**>	`5,6,7,8,10`

Example: **`aha1542=0x300`**

- *Adaptec AHA-274x / 284x / 294x Host adapter*
  `aic7xxx=<translation>,<busreset>`

Variable	Values / Meaning
<**translation**>	**`extended`** activates translation mode
<**busreset**>	**`no_reset`** disables resetting (engl. *reset*) the SCSI-bus while initializing the hostadapter

Example: **`aic7xxx=no_reset`**,
If the machine hangs at SCSI reset.

The SCSI hostadapter Adaptec 2920 is supported by the Future Domain driver (section 7.3.2, page 107).

- *AdvanSys SCSI-Host adapter*
  `advansys=<addr1>,<addr2>,...,<addrN>`

Example: **`advansys=0x110,0x210`**

This example tells the kernel to search for the AdvanSys host adapter at the given addresses.

- *AM53/79C974 SCSI-Host adapter*

  `AM53C974=<host-id>,<target-id>,<rate>,<offset>`

Variable	Values / Meaning
<**host-id**>	the host adapters SCSI-ID (often `7`)
<**target-id**>	the devices SCSI-ID (often `0..7`)
<**rate**>	`3,5,10` MHz/s max. transfer rate
<**offset**>	Transfer mode; `0` = asynchron

  If the host adapter seems to "swallow" some packages, reduce the maximum transfer rate for this device (e. g.the first CD-ROM SCSI drive `/dev/scd0` with ID 5) on the SCSI bus with:

  Example: **`AM53C974=7,5,3,0`**

- *Amount of SCSI devices per ID*

  `max_scsi_luns=<anzahl>`

Variable	Values / Meaning
<**anzahl**>	`1..8`

  Example:

  If the first LUN (logical unit number) should be used explicitly you have to set the parameter **`max_scsi_lun=1`**

- *ATAPI-CD-ROM*am (E)IDE-Controller

  `hd<x>=cdrom`

  `hd<x>=serialize`

Variable	Values / Meaning
<**x**>	a, b, c, d

  where:

Variable	Values / Meaning
`a`	Master on 1. IDE controller
`b`	Slave on 1. IDE controller
`c`	Master on 2. IDE controller
`d`	Slave on 2. IDE controller

  Example:

  An ATAPI CD-ROM as master on secondary IDE controller is set by **`hdc=cdrom`**.

- *Aztech CDA268-01 CD-ROM*

  `aztcd=<addr>[,0x79]`

  The value `0x79` only has to be set with unknown Firmware-version.

  Example: **`aztcd=0x320`**

  Since probing this driver takes extremely long it is recommended to pass **`aztcd=0`** if it is not needed. With our installation kernels disabling also can be done by **`aztcd=off`**.

- *BusLogic SCSI-Host adapter*

  BusLogic=<addr>

  Example: **BusLogic=0x300**

  This host adapter can be configured by far more parameters but these only serve for fine tuning the device and are described in `/usr/src/linux/drivers/scsi/BusLogic.c`.

- *EIDE controller chipsets*

  Some EIDE controllers do have faulty chipsets or cause problems if the secondary controller should be used.

  In the meantime there exists support for many of those chipsets in the kernel (section 13.4.5, page 188) which additional have to be activated by a kernel parameter.

  The following chipsets can be configured:

CMD 640	This chipset can be found on many mother boards. Since it contains many bugs the kernel offers a special support which recognizes the chip and bypasses the problems. Moreover, in some cases use of the secondary controller is only able by using this special code. In PCI systems this chip automatically is recognized; for VLB systems the following parameter is needed: **ide0=cmd640_vlb**
RZ 1000	This chip is used on many mother boards that use the Neptun chipset and it is buggy. If support for this chip is activated the system works a little slower but reliable. An additional activation via kernel parameter is not necessary.
DTC-2278	Only activating this driver via **ide0=dtc2278** makes it possible to use the secondary controller.
Holtek HT6560B	For activating the secondary controller the following parameter is needed: **ide0=ht6560b**
QDI QD6580	If this driver is activated it enables a higher speed: **ide0=qd6580**
UMC 8672	For activating the secondary controller the following parameter is needed: **ide0=umc8672**
ALI M1439/M1445	For activating the secondary controller the following parameter is needed: **ide0=ali14xx**
PROMISE DC4030	For activating the secondary controller the following parameter is needed: **ide0=dc4030**. CD-ROM and tapes on secondary controller are not supported yet.

Table 7.2: Special EIDE chipsets

- *Ethernet networking cards*

ether=<irq>,<addr>[,<par1>[,<par2>...<par8>]],<Name>

The different parameters for <**par1**> to <**par8**> do have different meanings for several drivers. Mostly only two parameters are needed, where the first is the start address and the second is the end addressed of *shared memory*. The first non-numerical argument is treated as the name.

<**irq**>	Used interrupt; `0` for autoprobing
<**addr**>	Port address; `0` for autoprobing
<**start**>	Start address for shared memory; some drivers use the 4 lowest bits for the debug level; the **Lance** uses them for its DMA channel.
<**end**>	End address for shared memory; the **3COM 3c503** driver uses this parameter to differ between internal and external transceivers. **Cabletron E21XX** card uses the lowest four bits, for selecting a media.
<**Name**>	The interface's name (normally **eth0**)

Table 7.3: Variable names for ethernet networking cards

The main reason for passing this parameter is to let the kernel recognize more than one networking card since only the first card is searched by default. This can easily be done by:

```
ether=0,0,eth1
```

Please notice that by passing `0` for IRQ and address the driver is explicitly told to launch *autoprobing* which means probing several addresses independently.

- *Floppy disk drives*

floppy=<drive>,<typee>,cmos

Variable	Values / Meaning
<**drive**>	0, 1, 2, 3
<**type**>	`0` - unknown or not recognized `1` - 5 1/4" DD, 360 KB `2` - 5 1/4" HD, 1,2 MB `3` - 3 1/2" DD, 720 KB `4` - 3 1/2" HD, 1,44 MB `5` - 3 1/2" ED, 2,88 MB `6` - 3 1/2" ED, 2,88 MB

floppy=<value>

For <**value**> the following values may be set (table 7.4):

all_drives	more than two floppy disk drives
asus_pci	denies access to 3. and 4. floppy disk drive
daring	only with reliable controllers – enhances performance
0,daring	Opposite of **daring**
<**addr**>**,two_fdc**	If the value <**addr**> is omitted while using a secondary floppy controller a port address of `0x370` is set.
thinkpad	**IBM Thinkpad**-machines
0,thinkpad	no Thinkpad-machine
unexpected_interrupts	Show warnings if something unexpected happens.
no_unexpected_interrupts	and...
L40SX	This value reaches the opposite of **unexpected_interrupts**

Table 7.4: Kernel parameter values for **floppy**

- *Future Domain TMC-16x0-SCSI-Host adapter*
  fdomain=<**addr**>**,**<**irq**>**[,**<**id**>**]**

Variable	Values / Meaning
<id>	The host adapters SCSI-ID `0..7`

This driver also servers the **Adaptec 2920** host adapter.
Example: **fdomain=0x140,11,7**

- *Future Domain TMC-885/950-Hostadapter*
  tmc8xx=<**addr**>**,**<**irq**>
  Example: **tmc8xx=0xca000,5**

- *Goldstar R420-CD-ROM drive*
  gscd=<**addr**>

- *Iomega ZIP Drive on parallel port*
  ppa=<**addr**>**[,**<**high**>**[,**<**low**>**[,**<**nybble**>**]]]**

Variable	Values / Meaning
<**addr**>	Port address
<**high**>	Waiting time while transferring data (micro seconds)
<**slow**>	Waiting time for other operations
<**nybble**>	if `1` use 4 bit mode

- *Logitech bus mouse*
  bmouse=<**irq**>

- *Mitsumi-CD-ROM drive*
  mcd=<**addr**>**,**<**irq**>**[,**<**wait**>**]**

Variable	Values / Meaning
<**wait**>	Value for waiting time while starting

  You can vary the parameter <**wait**> between `0` and `10` if the CD-ROM drive does not react fast enough on system requests (`"timeout"`) and therefore does not find the root image while installing.
  Example: **mcd=0x300,10,5**

- *Mitsumi-CD-ROM drive (Multisession)*
  mcdx=<**addr**>**,**<**irq**>
  Example: **mcd=0x300,10**

- *Mozart interface*
  isp16=<**addr**>**,**<**irq**>**,**<**dma**>**,**<**type**>

Variable	Values / Meaning
<**type**>	Sanyo, Panasonic, Sony, Mitsumi

  This driver is responsible for CD-ROM drives connected to either of **ISP16**, **MAD16** or **Mozart**. The value of <**type**> follows the interface plug where the CD-ROM is connected to the sound card.
  Example: **isp16=0x340,10,3,Sony**

- *NCR 5380 SCSI host adapter family*
  ncr5380=<**addr**>**,**<**irq**>**,**<**dma**>
  ncr5380=0x340,10,3

- *NCR 53c406a SCSI host adapter family*
  ncr53c406a=<**addr**>**[,**<**irq**>**[,**<**fastpio**>**]]**

Variable	Values / Meaning
<**fastpio**>	0, if no fast PIO mode is wanted

  Example: **ncr53c406a=0x340,10,0**

- *Optics Storage 8000 AT CD-ROM drive*
  optcd=<**addr**>
  Example: **optcd=0x340**

- *Parallel-Port CD-ROM drives*
  bpcd=<**addr**>
  Example: **bpcd=0x37f**

- *Philips CM206 CD-ROM drives*

  `cm206=<addr>,<irq>`

  Example: **`cm206=0x340,10`**

- *Pro Audio Spectrum 16 - SCSI host adapter*

  `pas16=<addr>,<irq>`

  On **Pro Audio Spectrum 16** there is an host adapter whose settings can propagated to the kernel by this parameter.

  Example: **`pas16=0x340,10`**

- *reboot mode*(at leaving Linux)

  `reboot=<modus>`

  here you have the following values for <**`modus`**>;

Variable	Values / Meaning
`warm`	Warm reboot (no memory check)
`cold`	Cold reboot (with memory check)
`bios`	BIOS reboot
`hard`	CPU crash reboot (triple fault)

  Example: **`reboot=cold`**
  boots the machine after shutdown as if the reset button has been pressed.

- *Protecting memory segments*(Reserve)

  `reserve=<start1>,<range1>,...,<startN>,<rangeN>`

  By means of this parameter you can reserve memory space for hardware which does not like being autoprobed and (e. g.) reacts with a system crash.

  Example: A “sensitive” networking card can be protected from being autoprobed and initialized by entering:

  `reserve=0x330,32 ether=5,0x330,eth0`

  In this example the networking card has a 32 bit data bus which starts at address `0x330` and uses interrupt `5`.

  For the full description of networking card parameters see section 7.3.2, page 106.

- *Passing root partitions*

  `root=<partition>`

Variable	Values / Meaning
<**`partition`**>	e. g. `/dev/hda1`, `/dev/sdb5`

  Example: **`root=/dev/hda5`**
  boots the kernel and tries to load the root partition from the first logical drive in the extended partition on the first (E)IDE hard disk.

- *Sanyo CD-ROM drive*

  sjcd=<addr>

  Example: **sjcd=0x340**

- *SCSI streamer*(Streamer)

  st=<puffer>,<threshold>[,<max>]

Variable	Values / Meaning
<**puffer**>	Size of buffer (Amount of blocks of 1 KB)
<**threshold**>	Write threshold (Amount of blocks of 1 KB)
<**max**>	Maximal amount of buffers; optional (e. g.2)

  Example:

- *Seagate ST01/02 SCSI host adapter*

  st0x=<addr>,<irq>

  Example: **st0x=0xc8000,5**

- *Sony CDU 31/33 A*

  cdu31a=<addr>,<irq>

  Autoprobing of this driver has been removed from the kernel, thus you have to set any parameters explicitly

  Example: **cdu31a=0x340,5**

  If there is no interrupt assigned to the drive and if you only can access it via polling mode you have to enter `0` for the IRQ.

  Example: **cdu31a=0x340,0**

  If this drive is (e. g.) connected to a Pro Audio Spectrum card this could resemble the following line:

  Example: **cdu31a=0x1f88,0,PAS**

- *Sony CDU 535*

  sonycd535=<addr>,<irq>

  Example: **sonycd535=0x340,10**

- *Size of main memory* (RAM)

  mem=<size>

  Size of memory can be entered in three different ways; either in bytes, kilobytes or megabytes. These examples show the different ways using a size of 96 MB.

  Examples:

  mem=96M
  mem=96000k
  mem=96000000

- *Soundblaster Pro 16 MultiCD*

  sbpcd=<addr>,<type>

Variable	Values / Meaning
<**type**>	`LaserMate`, `SPEA`, `SoundBlaster`

  Example: **sbpcd=0x340,10**

  Since probing this driver takes extremely long, it is recommended to disable it using the command:

  sbpcd=0

  On S.u.S.E. boot-disks this is also possible using

  sbpcd=off

- *Trantor T128/128F/228 SCSI host adapter*

  t128=<addr>,<irq>

  Example: **t128=0x340,10**

- *Trantor T130B SCSI host adapter*

  ncr53c400=<addr>,<irq>

  Example: **ncr53c400=0x340,10**

- *XT*h

  rd disk controllerparm.xt

  xd=<type>,<irq>,<addr>,<dma>

7.3.3 `insmod`-Parameter

This section now describes those parameters that can be loaded as modules. If you encounter difficulties while loading a driver (although you have entered the parameters) or if there is no section describing the parameter you have to integrate this driver into a monolithic kernel.

Some drivers do not exist as a module yet; some of them recognize your hardware properly only if they are compiled into the kernel. Anyway, it is recommended to try the "module variant" first.

If a driver is loaded as a module each and every variable used can be overwritten on command line. There is e. g.the variable **io** in **NE2000** driver which specifies the used I/O range. Thus, the correct command for loading this module is (cf. the kernel chapter, section 13.2, page 181):

```
root@earth:/ # insmod ne io=0x300 irq=10
```

or better with **modprobe**:

```
root@earth:/ # modprobe ne io=0x300 irq=10
```

Please be aware that there should be no spaces before or after the "equal". Moreover, hexadecimal values have to be set in the given form (leading '`0x`').

> If you want to enter more than one parameter they have to be separated by blanks! This is the main difference to entering parameters at the LILO prompt where no blanks should be used within parameters for one driver!

The parameters you enter here can additionally be integrated in `/etc/conf.modules`. Here many parameters can be assigned to a certain module. This is done in one line per module. The line should resemble the following:

```
options <module name> <parm1>=<wert1> ...
```

where:

Variable	Values / Meaning
<module name>	Name of the module without extension `.o`
<parm1>	Parameter 1
<wert1>	Value that is assigned to parameter 1

An entry for the example NE2000 card might look like:

```
options ne io=0x300 irq=10
```

Now, we want to list the most important parameters for most modules used.

- *3Com 3c501 / 3c503 / 3c505 / 3c507 Networking cards*
  Name of module: `3c501.o`, `3c503.o`, `3c505.o`, `3c507.o`

Parameter	Value
io	**<addr>**
irq	**<irq>**

  Example: **modprobe 3c509 io=0x300 irq=10**

- *3Com 3c509 / 3c579 Networking cards*
  Name of module: `3c509.o`, `3c579.o`

Parameter	Value
io	**<addr>**
irq	**<irq>**
xcvr	0: intern; 1: extern

  Example: **modprobe 3c509 io=0x300 irq=10 xcvr=0**

- *Adaptec AHA-1520 / 1522 / 1510 / 1515 / 1505 SCSI host adapter*
  Name of module: `aha152x.o`
  aha152x=<addr>,<irq>,<id>[,<rec>[,<par>]]

Variable	Values / Meaning
<id>	SCSI-ID of the hostadapter; 0, 1
<rec>	reconnect; 0, 1
<par>	parity; 0, 1

  Example: **modprobe aha152x aha152x=0x300,10,7,1,1**

- *AdvanSys SCSI host adapter*
  Name of module: `at1700.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe at1700 io=0x300 irq=10`**

- *Aztech CDA268-01 CD-ROM drive*
  Name of module: `aztcd.o`

Parameter	Value
aztcd	<**addr**>

Example: **`modprobe aztcd aztcd=0x300`**

- *Cabletron E21xx Networking card*
  Name of module: `e2100.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	<**addr**>
xcvr	0: intern; 1: extern

Example: **`modprobe e2100 io=0x300 irq=10 mem=0xd000`**

- *Digital DE425 / 434 / 435 / 450 / 500 Networking cards*
  Name of module: `de4x5.o`
  io=0x<**bus**><**device-id**>

Parameter	Value
bus	Number of the PCI bus, normally 0
device-id	Number of the PCI device

This data is shown with new PCI BIOSes at boot time, or you can view them under Linux respectively using:

```
root@earth: # cat /proc/pci
```

Example: **`modprobe de4x5 io=0x007`**

- *DECchip Tulip (dc21x4x) Networking cards*
  Name of module: `tulip.o`

Parameter	Value
io	<**addr**>
if_port	<**medium**>

where <**medium**> can be one of the following:

Variable	Values / Meaning
-1	auto
0	10TP
1	100TP
2	BNC

Example: **`modprobe tulip io=0x300 if_port=-1`**

- *Digital DEPCA / DE10x / DE20(012) / DE42, EtherWORKS Networking cards*
  Name of module: `depca.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe depca io=0x300 irq=10`**

- *D-Link DE620 pocket adaptor Networking card*
  Name of module: `de620.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
bnc	1 if BNC I/O
utp	1 if UTP I/O

Example: **`modprobe de620 io=0x300 irq=10 bnc=1 utp=0`**

- *EtherWORKS 3 (DE203, DE204, DE205) Networking card*
  Name of module: `ewrk3.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe ewrk3 io=0x300 irq=10`**

- *Intel EtherExpress 16 Networking card*
  Name of module: `eexpress.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe eexpress io=0x300 irq=10`**

- *Intel EtherExpressPro Networking card*
  Name of module: `eepro.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
mem	<**addr**>

Example: **`modprobe eepro io=0x300 irq=10 mem=0xd000`**

- *Fujitsu FMV-181/182/183/184 Networking card*
  Name of module: `fmv18x.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe fmv18x io=0x300 irq=10`**

- *Future Domain TMC-16x0 controller*
  Name of module: `fdomain.o`

Parameter	Value
setup_called	1
port_base	<**addr**>

Example: **`modprobe fdomain setup_called=1 port_base=0x300`**

- *Goldstar R420-CD-ROM drive*
  Name of module: `gscd.o`

Parameter	Value
gscd	<**addr**>

Example: **`modprobe gscd gscd=0x300`**

- *HP PCLAN+ (27247B and 27252A) Networking card*
  Name of module: `hp-plus.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **`modprobe hp-plus io=0x300 irq=10`**

- *HP PCLAN (27245 / 27xxx) Networking card*
  Name of module: `hp.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **modprobe hp io=0x300 irq=10**

- *HP 10/100 VG-AnyLAN (ISA, EISA, PCI) Networking cards*
  Name of module: `hp100.o`

Parameter	Value
hp100_port	<**addr**>

Example: **modprobe hp100 hp100_port=0x300**

- *IBM Tropic chipset Token Ring Networking card*
  Name of module: `ibmtr.o`

Parameter	Value
io	<**addr**>

Example: **modprobe ibmtr io=0x300**

- *ICL EtherTeam 16i / 32 Networking cards*
  Name of module: `eth16i.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **modprobe eth16i io=0x300 irq=10**

- *Iomega ZIP Drive on parallel port*
  Name of module: `ppa.o`

Parameter	Value
ppa_base	<**addr**>

Example: **modprobe ppa ppa_base=0x300**

- *Mitsumi CD-ROM drive*
  Name of module: `mcd.o`
  mcd=<addr>,<irq>[,<wait>]

Variable	Values / Meaning
<**wait**>	Delay at startup time

You can vary the <**wait**> from `0` to `10` if the CD-ROM drive does not react fast enough (`"timeout"`) thus probably not finding the root image during installation.

Example: **modprobe mcd mcd=0x300,10,5**

- *Mitsumi-CD-ROM drive (Multisession)*

  Name of module: `mcdx.o`

  modprobe mcdx mcdx=<addr>,<irq>

- *Mozart-Soundcard with CD-ROM drive interface*

  Name of module: `isp16.o`

  isp16=<addr>,<irq>,<dma>,<type>

Variable	Values / Meaning
<type>	Sanyo, Panasonic, Sony, Mitsumi

  This driver is responsible for CD-ROM drives connected to either of **ISP16**, **MAD16** or **Mozart**. The value of **<type>** follows the interface plug where the CD-ROM is connected to the sound card.

  Example: **modprobe isp16 isp16=0x300,10,1,sony**

- *Novell NE2000 / NE1000 Networking cards*

  Name of module: `ne.o`

Parameter	Value
io	**<addr>**
irq	**<irq>**

  Example: **modprobe ne io=0x300 irq=10**

- *NI6510 (AM7990 "lance" Chip) Networking card*

  Name of module: `ni6510.o`

Parameter	Value
io	**<addr>**
irq	**<irq>**
memstart	**<addr>**
memend	**<addr>**

  Example: **modprobe ni6510 io=0x300 irq=10**

- *Optics Storage 8000 AT CD-ROM drive*

  Name of module: `optcd.o`

Parameter	Value
optcd	**<addr>**

  Example: **modprobe optcd optcd=0x300**

- *Parallel-Port CD-ROM drive*

  Name of module: `bpcd.o`

Parameter	Value
bp_base	<addr>
bp_nybble	1

Example: **modprobe bpcd bp_base=0x300**

- *Parallel Port IP (PLIP)*
  Name of module: `plip.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **modprobe plip io=0x300 irq=10**

- *Philips CM206 CD-ROM drive*
  Name of module: `cm206.o`
  cm206=<**addr**>**,**<**irq**>
  Example: **modprobe cm206 cm206=0x300,irq=10**

- *Sanyo CD-ROM drive*
  Name of module: `sjcd.o`

Parameter	Value
sjcd	<**addr**>

Example: **modprobe sjcd sjcd=0x300**

- *SMC Ultra Networking card*
  Name of module: `smc-ultra.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>

Example: **modprobe smc-ultra io=0x300 irq=10**

- *SMC 9194 Networking card*
  Name of module: `smc9194.o`

Parameter	Value
io	<**addr**>
irq	<**irq**>
if_port	<**medium**>

where <**medium**> can be one of the following:

Variable	Values / Meaning
0	auto
1	TP
2	AUI, 10base2

Example: **`modprobe smc9194 io=0x300 irq=10 if_port=2`**

- *Sony CDU 31/33 A*
  Name of module: `cdu31a.o`

Parameter	Value
`cdu31a_port`	<**`addr`**>
`cdu31a_irq`	<**`irq`**>

Example: **`modprobe cdu31a cdu31a_port=0x300 cdu31a_irq=10`**

- *Sony CDU 535*
  Name of module: `sonycd535.o`

Parameter	Value
`sonycd535`	<**`addr`**>

Example: **`modprobe sonycd535 sonycd535=0x300`**

- *Soundblaster Pro 16 MultiCD*
  Name of module: `sbpcd.o`
  `sbpcd=<addr>,<type>`
  where <**`type`**> can be one of the following:

Variable	Values / Meaning
0	`LaserMate`
1	`SoundBlaster`
2	`SoundScape`
3	`Teac16bit`

Example: **`modprobe sbpcd sbpcd=0x300,1`**

- *Western Digital WD80x3 Networking card*
  Name of module: `wd.o`

Parameter	Value
`io`	<**`addr`**>
`irq`	<**`irq`**>

Example: **`modprobe wd io=0x300 irq=10`**

Chapter 8

LILO – what it does, how it works

One of the packages that are part of your Linux system is LILO, the *LI*nux *LO*ader (bootmanager). Former DOS users at first glance might think this is just one more of the all too many "tools and goodies" of the system; but upon closer inspection you will find that LILO plays a crucial role in making working with Linux comfortable.

Besides providing the ability to pass potentially needed additional information about your hardware to the kernel at boot time, the bootloader allows you to have multiple operating systems installed on a single machine and lets you selectively boot one of them from the hard disk.

This chapter describes how a bootloader works in general and how to install LILO on your system.

8.1 Booting a PC

After turning on your computer, the first thing that happens is that the BIOS (Basic Input Output System) takes control, initializes the screen and keyboard, and tests the main memory. Until this task is completed, no external devices or external storage media are known to the system.

Once the basic system has finished its internal setup it starts to verify the hardware around it. Date, time, and information about some of the most important external devices is read from the CMOS (**CMOS setup**) settings. After reading the CMOS, the BIOS should recognize the first hard disk (including details such as its geometry). It can then start to load the operating system (OS) from there.

MBR

To load the OS, the system loads a 512-byte data segment from the first hard disk into main memory and executes the code stored at the beginning of this segment. The instructions contained there determine the rest of the boot process. This is the reason why the first 512 bytes of the hard disk are often called **Master Boot Record** (MBR).

Even though the whole process is rather complicated and this description is fairly simplistic, it should be clear that up to this point (loading of the MBR), the boot sequence is independent of the installed operating system.

8.1.1 Master Boot Record

The layout of the MBR has been certified by an independent convention. The first 446 byte are reserved for program code[1]. The next 64 byte offer space for a partition table for up to four partitions[2]. The last 2 bytes have to contain a special “magic number” (AA55); any MBR which replaces this number by a different number is rejected.

The layout of a boot sector follows a certain standard. The first 446 bytes contain executable code, the next data section that contains space for a partition table.[3] The last two bytes of the 512 byte segment are used to mark a sector as valid boot sector.

8.1.2 Boot sectors

Boot sectors are the first sectors on a hard-disk partition[4]. They offer 512 bytes of space and are designed to contain code which is able to launch an operating system on this partition. Boot sectors of formatted DOS, Windows, or OS/2 partitions do so (and furthermore contain more basic instructions of the filesystem). On the contrary Linux boot partitions are empty at the very start. A Linux partition can not be started although it might contain a kernel and a valid root filesystem.

A valid boot sector follows the traditions and enters the “magic number” of the MBR into the last two bytes.

8.1.3 Booting DOS or Windows 95

The `MBR` contains information that determines which partition of a hard disk is “active”, i. e., which partition should be searched for the operating system to be booted. [5] The executable code in the MBR (first stage of the boot loader) tests whether the marked partition contains a valid boot sector.

If this is the case, the second stage of the boot loader can be started from there. DOS system programs can now be loaded and you will see the usual DOS prompt.

Under DOS, only primary partitions can be marked active. Therefore you can not use logical partitions inside an extended partition as bootable DOS partitions.

8.1.4 LILO –Here you go...

The Linux boot loader is usually installed in the `MBR`. (Details in section 8.3.4 nd section 8.4). When started, LILO already has access to both real-mode hard disks and due to its installation is able to write all data to the raw-device[6] without information on partitioning. Because of this, operating systems can

[1] The code itself — and its capabilities — do rely on the system that created the MBR
[2] Without this partition table no filesystem (MS-DOS: drives) can exist; i. e.you cannot use the hard disk
[3] You cannot use a hard disk without a partition table in its MBR under DOS.
[4] except for the extended partition which serves as a “container” for other partitions
[5] This implies that DOS has to be installed on the first hard drive.
[6] A raw device is a device that is accessed directly without using a filesystem

be booted from the first as well as from the second hard drive. The entries in the partition table that the standard DOS `MBR` uses to mark the active partition are ignored when using LILO in `MBR`.

An important difference to the standard DOS boot sequence is that you can select either of the installed systems at boot time when using LILO. After loading the `MBR` into memory, LILO is started and asks the user to select one of the installed operating systems (section 8.3.2).

8.2 Different boot concepts

The most simple boot concept affects only one machine with one operating system installed. A widely spread configuration is DOS or Windows 95 as the only system installed. The boot sequence for this case has already been outlined.

A similar concept can be used for Linux as well[7]; that is, if Linux is the only operating system being used. In this case one could theoretically skip the installation of LILO. The big disadvantage of doing this is that you can't pass additional parameters to the system kernel at boot time.

As soon as there is more than one operating system installed there are a couple of boot concepts.

8.2.1 Booting another OS from a floppy disk

The first OS can be booted from the hard disk; by use of boot-disks other operating system can be loaded alternatively

- *Requirements:*The floppy drive is bootable
- *Advantage:* You can skip the potentially tricky boot-loader installation
- *Disadvantage:* You have to make *sure* that you are not running out of working boot-disks.

8.2.2 Boot chaining of additional systems

The same OS is always booted and others can optionally be started from within the first OS.

- *Requirements:*Adequate programs for chain booting of operating systems are available.
- An *example* is the loading of Linux from DOS via **loadlin** or starting a NetWare server from DOS with **server.exe**.

8.2.3 Installing a bootmanager

Theoretically, this allows you to use an arbitrary number of operating systems on a single machine. The choice of systems is done at boot time. Changing operating systems requires a reboot.

- *Requirements:*The chosen boot manager works smoothly with all installed operating systems.

[7] You would have to write the Linux kernel directly onto a "raw" partition and launch this from MBR; this is rather uncommon, however

- An *example:* for (at least under certain circumstances) an cooperating bootmanager are OS/2[8] or the DOS boot loader **boot.sys**.

The following chapter describes installation and configuration of a boot manager, using the Linux boot manager LILOas an example. A complete description of LILO's abilities can be found in [Almesberger, 1994][9]

8.3 LILO custom fit

8.3.1 What is LILO?

LILO is a versatile boot manager; it can load and launch the following commands at system startup:

- Boot sectors of partitions (starting an operating system from this partition) — same as with other bootmanagers—,
- Linux kernel (starting Linux); this can not be done by most boot managers.

Furthermore it offers the important feature of passing a command line to the kernel. For security reasons this can be protected totally or partially by a password.

8.3.2 How to boot with LILO?

When LILO is launched it displays the text `LILO` and a greeting message (the latter you have entered during installation). Thereafter a command prompt appears

`boot:`

Here you select your operating system by entering its name, which is directly started up afterwards. The name of the operating systems also has been set by you during installation. You can now pass a parameter line to the Linux kernel. You can get a list of all names (of operating systems) by pressing ⟼ .

8.3.3 The components of LILO

The whole LILO–machinery includes the following components[10]:

- The beginning — the "first step" — of the LILO code in a boot sector (to be described later) which activates the system start.
- the LILOcode (its "heart"). Localised in `/boot/boot.b`.

---

[8] more in section 8.6.3

[9] this file can be printed by entering

```
root@earth:/usr/doc/packages/lilo # touch user.tex
root@earth:/usr/doc/packages/lilo # make all
root@earth:/usr/doc/packages/lilo # lpr user.dvi
```

[10] Besides: the boot sectors installed by LILO contains a typical virus boot sequence. Therefore it is not astonishing that DOS-virus scanners claim to have found a **AIRCOP** boot-sector-virus in files as `/boot/any_b.b` or `/boot/any_d.b`

- a *map* file where LILO enters the location of Linux kernels and other data during its installation. Normally: `/boot/map`
- The different Linux kernel and boot sectors that LILOshould offer.

> Any write access (even through file movements) on any of these files turns the map file corrupt thus requiring you to *reinstall* LILO (section 8.4.1). This mainly applies if you change to another kernel.

8.3.4 Where LILO can be installed

This is referring to the mentioned above "first step" of LILO. Before going into detail we would like to show the general restrictions:

> *Any* part of LILO has to be on the *first 1024 cylinders* on hard disks!

This is because only these are cylinders are available at BIOS system start. With older BIOSes and IDE disks one can be restricted to either of the first two hard disks (`/dev/hda` and `/dev/hdb`).

This should be considered *before* first time installation (section S1) – it is too late afterwards and causes you lots of additional work. More detailed information please find in section 8.7.3.

For the "first steps" there are three possibilities:

- On a *disk*

  This is the most secure but also the slowest alternative for booting with LILO(see section 8.5.2). For those of you who do not want to change boot sectors please (now) choose this alternative.

- In *boot sector* of a primary Linux partition on the first hard disk. This leaves the MBR untouched. Before being bootable it has to be marked active with **fdisk**. If Linux is fully installed on logical drives or partitions on the second hard disk there is only the boot sector of the second drive left (if there is one). Linux **fdisk** can even activate such a partition.

  If you like to boot multiple systems from hard disk this is quite annoying; every time you want to boot you have to activate the corresponding boot sector *beforehand*. The next two variants are much more comfortable.

- In *Master Boot Record* This variant offers the highest flexibility. Moreover this is the only alternative if all of your Linux partitions reside on the second hard disk and there is no extended partition on the first disk. Every setting of the MBR should be edited with extreme care since errors may have severe consequences. Safety aspects are described in section 8.4

- If there is *another* bootmanager already installed and running ...

  and you want to continue using it further there are some variations depending on the flexibility and power of it. A common case: You have a primary Linux partition on the second hard disk where you start your Linux from. Your bootmanager is able to start this partition via a boot sector. Then you can activate your Linux partition by installing LILO into its boot sector and tell your boot manager that it is active.

> *Be careful* while trying to make a *logical* Linux partition bootable by installing LILO onto it: success *is not guaranteed!*, even if your other boot manager could launch logical partitions (mai 1997).

Well, you can try it; the best way is to use a tiny Linux installation for making sure. Probably you are lucky. Anyway, creating a primary and bootable Linux partition is recommended.

8.3.5 Configuring LILO

Being a rather flexible boot manager LILO offers many ways of adapting a configuration to one's needs. The most important options and meanings are described below if you want to go into more detail consider having a look at [Almesberger, 1994].

Configuration of LILO is done in `/etc/lilo.conf`. If you are installing LILO for the first time we recommend to use YaST to configure LILO.. An adaptation of `/etc/lilo.conf` can be done later on.

It is recommended to keep any existing old (working) configuration file and to keep it in a save place. Your settings only take place if you reinstall LILO after having changed `/etc/lilo.conf` (see section 8.4)!

Construction of `lilo.conf`

`/etc/lilo.conf` starts with a **`global section`** followed by one or more **`system sections`** for each operating system LILO should start. A new section is started by either of **`image=`** or **`other=`**.

The order of entries in `/etc/lilo.conf` only matters in as much as the first one in the list is booted by default if no user interaction is taken — this can be set to **`delay=`** and **`timeout=`**.

file contents 8.3.1 (page page 127) shows a sample configuration on a machine with DOS and Linux. There are two Linux kernels (an older and a newer one) on the momentary root partition (primary on second disk) as well as MS-DOS (or Windows 85) on `/dev/hda1`.

Anything between a '`#`' and the end of line is regarded as a comment. Spaces and comments are ignored by LILO and can be used to improve readability.

Now we go through the most important lines step by step, the other options will follow.

- **Global section** (Parameter part)
  - **`boot=`**<**`bootdevice`**> The device on whose first sector LILO should be installed.
    <bootdevice> may be: a floppy disk drive (`/dev/fd0`), a partition (e. g. `/dev/hdb3`), or a whole disk (e. g. `/dev/hda`); the latter means installing LILO in MBR. Default: if this option is missing LILO is installed on the actual root partition.
  - **`prompt`**
    Forces sending the LILO prompt. Default is: no prompt! (compare to **`delay`** further down). This is recommended if LILO should manage

```
# LILO Configuration file
# Start LILO global Section
boot=/dev/hda                # LILO Installation target
backup=/boot/MBR.hda.970428  # Backup file for the old  MBR
                             # Apr 28 1997
#compact        # faster, but won't work on all systems.
linear
message=/boot/greetings      #  LILO's Greeting
prompt
password = q99iwr4           # General LILO password
timeout=100     # wait at prompt for 10 s before default
                # is booted
vga = normal    # normal text mode (80x25 characters)
# End LILO global section

# Linux bootable partition config begins
image = /vmlinuz             #  Setting
    root = /dev/hdb3         #  Root partition for kernel
    read-only
    label = Linux
# Linux bootable partition config ends

# Second Linux bootable partition config
image = /vmlinuz.old
    root = /dev/hdb3
    read-only
    label = Linux.old
# 2nd Linux bootable partition config ends

# DOS bootable partition config begins
other = /dev/hda1
    label = DOS
    loader = /boot/chain.b
    table = /dev/hda
# DOS bootable partition config ends
```

File contents 8.3.1: Sample configuration in `/etc/lilo.conf` more than one system. Additional **`timeout`** should be set for guaranteeing an automatic reboot if nothing is entered at the prompt.

- **`timeout=`**<**`tenth-seconds`**>

  Sets a time-out for the prompt option, thus enabling an automatic reboot if no entry occurs in the given time. <tenth-seconds> is the remaining time in 0.1 s. Pressing Shift ⇑ starts the time-out over. Default: infinite, i. e.no automatic reboot!

- **Linux section**

  - **`image=`**<**`kernelimage`**>

    Here has to be the name of the kernel. With your new system this most probably is `/vmlinuz` or `/zImage`.

  - **`label=`**<**`name`**>

    This name has to be unique in `/etc/lilo.conf` otherwise you can freely choose a name for the system (e. g.**`Linux`**). Maximum length is 15 characters; you should consider using letters, numbers and underscore —no blanks—, special characters as German umlauts[11]. Default is: the filename of the kernel image (e. g.`vmlinuz`).

    By entering this name you select which system to boot at boot time at the LILO prompt. It is recommended if there are many systems installed to keep track of them in a special message file (**`message=`**).

  - **`root=`**<**`rootdevice`**>

    This is for giving the kernel the name of the root partition (e. g. `/dev/hda2`) of your Linux system. This is recommended for security reasons! If this option is omitted the kernel takes its own root partition[12].

- **Other systems**

  - **`other=`**<**`partition`**>

    `other` tells LILO to start partitions of other systems (e. g. `/dev/hda1`).

  - **`loader=`**<**`Boot loader`**>

    For loading a boot sector that belongs to another operating system, a certainLILO constructs an "Pseudo MBR" in its map file (at boot time LILO first starts this "Pseudo MBR" and this starts the foreign boot sector. This option sets the file where to get the code for the Pseudo MBR.

    Default: `/boot/chain.b` (generally this is ok)[13]. But even this has to be set if the start partition is neither on the first floppy nor on the first hard disk. In our given example this is not necessary and only set for illustrational purposes.

  - **`table=`**<**`ptabelle`**>

    <partition-table> has to set the source device for the partition table that should be written into Pseudo MBR (normally `/dev/hda` oder `/dev/sda`).

[11] The specific rules on which characters to use can be found at [Almesberger, 1994], 3.2.1.

[12] it can be seen with the command **rdev** <kernelimage>

[13] some mor loaders for special purposes are `any_b.b`, `any_d.b` and `os2_d.b`.

– **label=<name>**

  Name (free choice) for the system. Recommended. The default — the raw device name — is not that explanatory.

Optional options for configuration (selection)

The last sector covered the neccesary entries in `/etc/lilo.conf`. Some additional and useful options should be discussed here.

Those options that are marked as image options belong to the appropriate section of the operating system. The others are considered for the global section of `/etc/lilo.conf`.

- **backup=<Backup file>** (optional)

  The file where LILO backups the boot sector. The default is `/boot/boot.xxxx` (where `xxxx` is the internal device number of the installation partition)[14].

  We recommend not using a cryptic name, as in our example with device name and date.

> If the backup file exists LILO does *not* create a new one! Please make sure to use a name not in use.

- **compact**

  This option is recommended if you want to install LILO onto a disk. If enabled LILO tries to read more sectors at a time and thus might boot faster. This does not work on every machine. You should not set this as it is safer and after all, it is only one or two seconds.

- **linear**

  This option causes all references to sectors to be written as logical instead of physical addresses. This option might come in handy if LILO does not recognize the geometry of the hard disk correctly. Still it does not make the 1024 cylinders border obsolete. This is scarceley ever needed.

- **message=<message-file>** (optional)

  Points to a textfile that should be shown on screen at system boot up. It should not contain more than 24 lines and can present an overview of the LILO boot selection. Recommended!

> If this option is set the message file belongs to the LILO boot machinery and after every change of this file LILO has to be reinstalled (section 8.4)!

- **password=<password>** (optional)

  May be located either in a global or system specific section. Locks LILO services and booting with a password. If you take this seriously you should remove the password from `lilo.conf` after first use. As `'root'` you can set a new password for LILO any time you like (just reinstall it).

[14] To be found in kernel sources under `/usr/src/linux/init/main.c`, function `parse_root_dev()`

- **`read-only`**

  This option tells the kernel to initially mount the root partition read-only as commom to all Linux systems. If this is omitted the kernel uses its internal settings[15]

- **`delay=`<`tenth-seconds`>**

  If the prompt is *not* explicitly set the user can order a prompt by typing (Shift ⇑ , Strg , Alt). The **`delay=`** option sets to time to elapse before LILOboots the first system in its list. Default is: 0, i. e.no waiting.

  The **`delay`** option is *obsolete* if a prompt is ordered by **`rompt`** anyway.

- **`vga=`<`mode`>**

  Selects VGA mode at startup. Valid <mode>s are `normal` (80x25), `ext` (80x50) or `ask` (well, ask).

- **`append=`¨<`parameter`>¨**

  Image option for Linux kernel. Enables passing kernel parameters and passing hardware components at the LILO prompt. The kernel first gets the **`append`** line then the prompt; that's why prompt commands predominate at collisions; For example: **`append="mcd=0x300,10"`**
  ⟶ list of kernel parameters (Page chapter 7, page 101)

8.4 Installation and deinstallation of LILO

During a new installation YaST leads you through all the steps interactively. Additional action in general is not necessary. Well, here we assume that LILO should be integrated with certain special options.

> The installation of a boot manager is rather tricky! Please ensure *in advance* that you are *100%* able to boot your Linux or other systems respectively. You must have **fdisk** installed on this disk otherwise you might find yourself in the bad situation of not being able to access your hard disk at all!

8.4.1 Installation after changing the configuration

If any of the LILO components have changed or you have modified your configuration in `/etc/lilo.conf` you will have to reinstall LILO. This is easily done by launching the "*Map Installer*".

```
root@earth: # /sbin/lilo
```

What happens now is that LILO writes a backup of the target-boot-sector, writes its "first step" into it and creates a new map file (comp. section 8.3.3). LILO now announces each installed system — in our case (see screen output 8.4.1):

After correct installation the machine can be rebooted.

```
root@earth: # shutdown -r now
```

[15] This can be seen using the command `rdev-R` <kernelimage>. Installation and freshly compiled kernels have read-only set by default (check!); therefore you normally do not need this option.

```
Added Linux*
Added Linux.old
Added DOS
```

Screen output 8.4.1: Output after launching LILO

First the BIOS performs its system test and directly afterwards you will see LILO and its command prompt where you can enter parameters and select a bootimage (the recently installed configurations). ⟼ shows you a list of all installed systems.

8.4.2 Installation recompiling a kernel

If you want to include a freshly created kernel into your boot concept there is a comfortable alternative apart from the LILO installation:

Organization of commands needed for configuring and compiling a kernel is written to `/usr/src/linux/Makefile`. There is a **`target`** named **`zlilo`** in `Makefile` which automatically copies `/vmlinuz` to `/vmlinuz.old` and the new kernel to `/vmlinuz` and reinstalls LILO. This can be done by the command:

```
root@earth:/usr/src/linux # make zlilo
```

This, of course, only is useful if you have adjusted `/etc/lilo.conf` *in advance*. The new as well as the old kernel should now be listed. file contents 8.3.1 shows an example.

At the LILO prompt you can launch either of the two kernels. That improves safety because you can boot even if one of your kernels fails.

⟶ creating a new kernel (Page chapter 13, page 181)

8.4.3 Removing LILO

> Removing a boot manager is rather tricky! Please ensure *in advance* that you are *100%* able to boot your Linux or other systems respectively. You must have **fdisk** installed on this disk otherwise you might find yourself in the bad situation of not being able to access your hard disk at all!

Perhaps one day it will be necessary for you to deinstall LILO`:-(` . This is achieved by writing back the target boot sector where LILO has been installed. This is no problem under Linux *if* there is a valid backup (comp. section 8.3.5, Option **`backup`**).

> A boot sector backup is turned invalid if the partition has got a new filesystem (For DOS users: being formatted). The partition table of an MBR backup becomes invalid if the hard disk in question has been repartitioned in the meantime. Such “backups” are time-bombs; the best is to delete them as soon as possible! Unpacking old and invalid backups into system sectors is a rather straight way to data loss!

It is very simple to get back a DOS, Windows 95 or OS/2 MBR. Just enter the MS-DOS command (available since 5.0)

```
C:\> FDISK /MBR
```

or on OS/2

```
C:\> FDISK /NEWMBR
```

These commands only write the first 446 bytes (the boot code) into MBR and leave partitions untouched[16].

Otherwise make a backup of the LILO sector in question — the safer the better. Now you should check (at least twice `:-)`) whether your old backup file is the correct one and if it is exactly 512 bytes in size! At last you play it back but do not confuse `if=` and `of=`!

- If LILO resides in partition yyyy (e. g. hda1, hda2, . . .):
  ```
  root@earth: # dd if=/dev/yyyy of=New-File bs=512 count=1
  root@earth: # dd if=Backup-Datei of=/dev/yyyy
  ```
- If LILO resides in MBR of zzz (e. g. hda, sda):
  ```
  root@earth: # dd if=/dev/zzz of=New-File bs=512 count=1
  root@earth: # dd if=Backup-Datei of=/dev/zzz bs=446 count=1
  ```

The last command is cautious and does not write partition table. By the way, note how easy and fast a boot sector backup is done! It is recommended to do this frequently!

8.5 Creating a Linux-bootdisk

A Linux boot-disk consists (a little simplified) of one or more Linux kernels, possibly managed by LILO. It serves for starting up your system even if booting directly from hard disk is not possible (possible reasons: overwritten MBR, misconfigured bootmanager, errors while installing and many more...).

Such a boot-disk *only* loads the kernel; everything else (system programs, `init` start scripts) has to be provided by the system and has to work. The connection is established via the kernel. The kernel gets the root device of your machine.

Do not confuse this with S.u.S.E.-boot-disk for installation and emergencies (which you can create according to `disks` anytime). The latter load a fully independent Linux system supplying the ability of accessing the existing Linux partitions from "the outside" (e. g. mounting). This is necessary if you encounter errors on your hard disk (damaged filesystems, forgotten 'root' password etc.) and resembles booting DOS or OS/2 via boot-disk.

8.5.1 Boot-disk without LILO

If you are in the lucky situation that your kernel does *not* need any hardware parameters the easiest way of creating a boot-disk is to just write the actual kernel onto a raw disk and to adjust the root device (if not done before).

[16] provided that the MBR (section 8.1.1) has valid code! If not it is considered invalid and the partition table is moved to "null" . . .

```
root@earth: # dd if=Ihr_Kernel of=/dev/fd0 bs=18k
root@earth: # rdev /dev/fd0 Your_Root_Partition
root@earth: # rdev -R /dev/fd0 1
```

The last command makes sure that the kernel initially mounts root read-only (as it is expected; the startup scripts rely on it).

8.5.2 Boot-disk with LILO

A much more comfortable boot-disk with greeting, prompt, kernel parameters and other LILO goodies is created by transferring a complete start machinery onto the disk (comp. section 8.3.3). For this the disk needs a filesystem; Minix suits this best.

If you only want to install a kernel and you do not want the greeting YaST can perform this task for you (in 'Administrating the system').

> Your kernel has to follow the old Linux tradition and should be labeled `vmlinuz` otherwise YaST will not find it!

Otherwise you proceed as follows:

- Assign a minix filesystem on a new and empty disk —force an error check for safety reasons—; after this is done mount the disk to, e. g., `/mnt`:
  root@earth: `# /sbin/mkfs.minix -c /dev/fd0 1440`
  root@earth: `# /bin/mount /dev/fd0 /mnt`
- now copy your kernel files and the LILO file `/boot/boot.b` to `/mnt` (i. e.onto the disk).
- optional: create a message file `/mnt/message`
- Create an own `lilo.conf` in `/mnt`. Of course you have to adapt this to your needs (give the correct name of the kernel etc.):
- install LILO with *this* `lilo.conf`:

  root@earth: `# /sbin/lilo -C /mnt/lilo.conf`
- Unmount floppy; you are done!

  root@earth: `# /bin/umount /mnt`
- Do not forget to check your boot-disk at the next system start as to whether it works or not `:-)`

8.6 Sample configurations

If your Linux is all alone on your machine there is no immediate action to be taken since everything needed has already be done by YaST.

8.6.1 DOS/Windows 95 and Linux

Required: There is at least a primary partition for each DOS/Windows 95 and Linux which is below the 1024 cylinders limit (section 8.3.4).

For this case we already know a configuration (file contents 8.3.1) — only the settings for **`root=`**, **`image=`** and **`other=`** have to be adapted. LILO is installed

```
# LILO Configuration file bootdisk
# Start LILO global Section
boot=/dev/fd0                # Installation: Floppy
install=/mnt/boot.b          # Of course LILO and
map=/mnt/map                 # map file onto floppy!
message=/mnt/message         # optional
prompt
timeout=100     # Wait at prompt: 10 s
vga = normal    #
# End LILO global section
#
# Linux bootable partition config begins
image = /mnt/First_Kernel      #   default
    root = /dev/Your_Root_Device # Here your root partition!
    label = linux
# Linux bootable partition config ends
#
# System sections for more kernel here:
```

File contents 8.5.1: `lilo.conf` for a bootdisk

in MBR. Of course you should omit the `Linux.old` if you do not have an old kernel installed.

Save your `/etc/lilo.conf` and a Linux boot-disk. Especially Windows 95 feels inclined[17] to eliminate "foreign" MBRs. If you are able to boot Linux using your boot-disk this problem is solved rather quickly.

```
root@earth: # /sbin/lilo
```

8.6.2 Windows NT and Linux on one hard disk

1. For booting the *NT bootmanager* is used. This can start images or the boot sector itself. Execution of the following steps prepares everything for a friendly coexistence of Linux and Windows NT:
   - Installation of NT
   - Partition the NT disks (using FAT so that Linux can write on it).
   - Install Linux "as usual" (in our example the root partition is on `/dev/sda3`). Mount DOS data disk (e. g. on `/dos`).
   - Install LILO but install it in Linux' root partition (`/dev/sda3`), **not** in MBR (`/dev/sda`)! You still have the possibility of configuring a selection of Linux kernel for LILO. Example-`lilo.conf`: see file contents 8.6.1.
   - Copy the LILO boot sector where NT can find it; e. g.
     `dd if=/dev/sda3 bs=512 count=1 of=/dosen/bootsek.lin`
   - Boot NT. Copy from data disk to main directory of NT's system drive (if it is not already there).

[17] A complete list is not known …

```
# LILO Configuration file
# Start LILO global Section
boot=/dev/sda3                # Target of installation
backup=/boot/boot.sda3.970428 # Backup previous boot sector;
                              # 28. Apr 1997
prompt
timeout=100     # Wait at prompt: 10 s
vga = normal    # force sane state
# End LILO global section
# Linux bootable partition config begins
image = /vmlinuz             #   default image to boot
    root = /dev/sda?         # Here the root partition!
    label = Linux
# Linux bootable partition config ends
```

File contents 8.6.1: `lilo.conf` for making a Linux root partition startable

- In `boot.ini` (setting attributes) supplement at the end:
  `c:\bootsek.lin="Linux"`
- After next booting (if everything went right) there should be an entry in NT's bootmanager!

2. Other possibility: install LILO in MBR and claim that it's DOS to Windows NT (as in our previous example).

> NT 3.5* does not recognize Linux' partition types 82 and 83! Please make sure that no NT programm tries to "repair" your partition table. Data loss would be the result! Always have valid backups of the LILOboot sector at hand.

8.6.3 OS/2 and Linux

1. Possibility: The *OS/2 bootmanager* is used for booting. It can launch unlimited primary and logical partitions provided they are below the 1024 cylinders limit. The user is responsible for the partitions This bootmanager is configured by OS/2 **fdisk**.

   Preparation on Linux side: is just making a partition startable (usually this is the root partition) with LILO. You can use the same `lilo.conf` as in the Windows NT example, but ther is one thing that you have to consider *in advance* ...

   Preparation on OS/2 side: OS/2 not only uses the conventional and understandable entries for existing partitions (in MBRs on hard disks) but uses "waste" space in this sectors for additional infos[18]. If these are inconsistent, OS/2's **fdisk** considers this partitions to be faulty and refuses bootmanager services. The **fdisk**s of other systems of course do not know how to handle these additional partitions ... Conflicts are rather natural.

[18] A new Support DataBase article is about to be written; keyword "OS/2"

Therefore: Before installing Linux load OS/2 (the installation system is sufficient) and create the Linux partitions with OS/2's **fdisk**, at least the logical drives. This initially creates additional OS/2 partitions which can be in the way. *Solution:* Immediately after creating the partitions load the Linux installation system (or the rescue disk from S.u.S.E. LinuxCD) and change partition types to 83 (Linux native). Now these partitions will be ignored by OS/2.

2. Possibility: Install LILO as a main bootmanager on a primary partition on the first disk[19]. This special case is considered in our next example where DOS is involved additionally.

8.6.4 DOS, OS/2 and Linux

1. If you have used the *OS/2 bootmanager* for DOS and OS/2 and want to continue using it you simply add Linux to its start menu as described in the previous example.
2. Possibility: *LILO* is installed as the main bootmanager on a primary partition of the first hard disk.

   The following, purposely complicated example for `lilo.conf` (file contents 8.6.2, page page 137) assumes that DOS' and Linux' startpartition are primary and on the first hard disk whereas OS/2 (logical drive) resides on the second hard disk — all of them below 1024 cylinders. OS/2 is on the second disk. That's why a special loader (`/boot/os2_b.b` instead of`/boot/chain.b`) is used. The MBR code might be either from DOS or OS/2 (of no consequence). The LILO startpartition (`/dev/sda4`) has to be marked active with any **fdisk**.

8.7 LILO problems

8.7.1 Some guidelines

Some simple guidelines at the beginning to avoid most LILO problems in advance (this is from the LILO documentation [Almesberger, 1994]):

- *Do not panic!* If anything does not work try to find the error and/or the cause first; check the diagnosis and then commence with fixing the problem.
- Always have an up to date and tested *boot-disk* at hand.
- S.u.S.E. Linux 5.0 contains a full Linux system on its boot-disk and installation CD (For rescue system see section 11.4) for enabling you to reach all your Linux partitions. Included are tools for repairing (almost) any problems that can occur.
- Read this documentation. Especially if the system does not do what you want it to do.
- Check `/etc/lilo.conf` *before* installing the map installer (`/sbin/lilo`).
- Be attentive if using big, or more than one disks. Be aware of the 1024-cylinders limit!

[19] Not that good an idea to install in MBR as repartitioning with another **fdisk** could erase the MBR and thus remove LILO.

```
# LILO Configuration file
# Start LILO global Section
boot   = /dev/sda4        # LILO in Linux root partition
backup = /boot/boot.sda4.970428
message = /boot/message  #  Greeting message
prompt
delay = 100
vga   = normal
#
# Linux bootable partition config begins
image = /vmlinuz
    label = linux
    root  = /dev/sda4
# Linux bootable partition config ends
#
# OS/2 bootable partition config begins
other = /dev/sdb5
    table = /dev/sdb
    label = os2
    loader = /boot/os2_b.b
# OS/2 bootable partition config ends
#
# DOS bootable partition config begins
other = /dev/sda1
    table = /dev/sda
    label = dos
# DOS bootable partition config ends
```

File contents 8.6.2: LILOwith DOS, OS/2 and Linux on two hard disks

8.7.2 Diagnosis of errors: LILO start-messages

This mainly is section 5.2.1 from [Almesberger, 1994].

If LILO is launched it displays the word `'LILO'`. Every letter signifies a certain state. If LILO cannot be launched the output letters give a hint on where the error occured.

none No part of LILO has been loaded. Either LILO is not installed at all or the partition with LILO's boot sector has not been started.

'L' ***error*** ... The "first step" has been loaded and started but the second step (`/boot/boot.b`) could not be loaded. This normally points to a physical error of the boot device or a faulty disk-geometry.

'LI' Second step has been invoked but could not be started. This can be due to a faulty disk-geometry or to moving `/boot/boot.b` without reinstalling LILO.

'LIL' Second step of LILO has been started but could not allocate needed data (pointer etc.) from its map file. This typically is due to a physical error of the boot device or a faulty disk-geometry.

'LIL?' The second step has been loaded at a wrong memory address. This can be due to a subtile error in disk-geometry or to moving `/boot/boot.b` without reinstalling LILO.

'LIL-' Data in the map file are invalid. This can be due to a faulty disk-geometry or to moving `/boot/boot.b` without reinstalling LILO.

'LILO' Every part of LILO has been loaded successfully.

The most common causes for *geometry errors* are not physical defects or invalid partition tables but errors in LILO installation.

- Disregard of the 1024 cylinders limit (see next section);
- Unsuccessfull attempt at starting LILO from a logical partition.

8.7.3 The 1024-cylinder limit

As emphasized a couple of times before (section 8.3.4) the whole LILO machinery (i. e.any data that is needed for starting) has to be accessible by BIOS calls. The sections that can be used (we call theses *allowed sections*) have already been highlighted.

Which possibilities (besides the restrictions) are still left? Quite a couple! Keep in mind that *only* the start-up machinery is concerned. There is now law that says it has to be on the Linux root-partition; it even is possible (but quite dangerous too) to put the start-machinery onto partitions of other operating systems as long as Linux has read- and write access.

> Never install the LILO boot sector on an unknown partition because you will severly damage their filesystem!

- The best way, of course, is to install a primary partition (within the allowed section) and to install all LILO files (including the LILO boot sector) onto this partition. This mostly will be the Linux root partition.

  You can also add it to `/boot` with YaST. The only condition is, that there has to be enough space for

  – `boot.b`, `map`, `message`,
  – the Linux kernes LILO should boot.

  Some mega-bytes will be sufficient. It does not matter where you put the rest of your partitions. There are no more restrictions; as soon as the kernel runs you have unrestricted access to all installed disks.

But what to do if there is no space for such a partition? If you neither want to repartition your disk, nor to upgrade to SCSI, nor want to purchase a new BIOS version, there still are two (makeshift) possibilities:

- Use a boot-disk instead of LILO on the hard disk, or if you are additionally running MS-DOS, you can use **loadlin** as well.
- Install LILO start-machinery onto a none Linux partition which is in the allowed section and where Linux has write access (e. g.a FAT/VFAT drive). Of course we cannot put the LILO boot sector there! So there are only two places where to put it. Either the start of an extended partition

on the first disk — as long as it is beneath the 1024 cylinder limit — or the MBR.

Consider that the partition in question is mounted to `mnt`. LILO should be installed in MBR (`/dev/hda`) and should additionally boot DOS from `/dev/hda1`. You should proceed as follows:

- Create a new directory (e. g.`/mnt/LINUX`) and copy the LILO files mentioned above thereto: `boot.b`, `map`, `message`, as well as the Chain-Loader of other operating systems (normally `chain.b`) and of course the Linux kernel LILO should boot.
- Create a `/mnt/LINUX/lilo.conf` where all paths point to `/mnt/LINUX`.

```
# LILO Configuration file
# Start LILO global Section
boot=/dev/hda              # Installation target
backup=/mnt/LINUX/hda.xxxx  # backup of old MBR
install=/mnt/LINUX/boot.b  # Of course LILO and
map=/mnt/LINUX/map         # map file are in /mnt/LINUX!
message=/mnt/LINUX/message # optional
prompt
timeout=100     # Wait at prompt: 10 s
vga = normal    #
# End LILO global section
#
# Linux bootable partition config begins
image = /mnt/LINUX/Erster_Kernel   #   default
    root = /dev/Ihr_Root_Device    # Root partition!
    label = linux
# Linux bootable partition config ends
#
# System section for other kernels:
#
# Ende Linux
# DOS bootable partition config begins
other = /dev/hda1      # MSDOS system drive
    label = dos
    loader = /mnt/LINUX/chain.b
    table = /dev/hda
# DOS bootable partition config ends
```

File contents 8.7.1: `lilo.conf` for other partitions

- Install LILO with *this* `lilo.conf`:

  `root@earth: #` `/sbin/lilo -C /mnt/LINUX/lilo.conf`

  After that LILO should run. Boot MS-DOS and protect the LILO files as well as possible against write access (Just to remind you: any write access disables LILO!). Anyway you should assign to all files under `X:\LINUX` (wher the '`X`' is the DOS drive mounted to `/mnt`) the DOS attributes *system* and *hide*.

As conclusion we want to point out to two HOWTOs in `/usr/doc/howto/mini/`, these are `LILO.gz` and `Large-Disk.gz`.

8.7.4 LILO and DOS-diskmanager (OnTrack-Manager, EZ-Drive)

There are particular details on this topic in the above mentioned mini-HOWTO `/usr/doc/howto/mini/Large-Disk.gz`.

General issues on the DOS diskmanagers

A DOS diskmanager is most often installed in the MBR of IDE drives. Since the BIOS and DOS can only handle 1024 cylinders the manager replaces the BIOS functions for accessing the disk with its own. It recalculates the existing geometry with a recalculation routine to a pseudo one below 1024 cylinders. The problem is that this *pseudo* geometry is entered into the partition table.

(This is no problem to SCSI or EIDE disks (connected to an EIDE controller) since this recalculation (if needed) is done by the hardware. Linux can handle this without any problems.)

Well, Linux is not compatible to diskmanagers since it does not access the disk with BIOS calls! There will be severe problems as Linux tries to read these "psudo" geometries and takes them seriously!

From kernel 1.3.29 onwards Linux can detect these OnTrack managers and EZ drives and support them. But this is only a makeshift!

The best solution however, is to disable the manager and let DOS only see the first 504 MB. Then Linux can be installed (except LILO) on the back part of the disk.

Furthermore you can think about installing an EIDE controller since then the manager is no longer needed.

OnTrack-Manager: Deinstallation

The following hint has kindly been given to us by **Matthias Wimpf**.

This description relies on version 7.04 of the OnTrack-manager. For other products this procedure can differ.

1. IMPORTANT!
   Do make a backup of the disk! All data will get lost while removing the diskmanager and have to be unpacked later on. For those of you who additionally install an EIDE controller this might not be relevant (see 3. and conclusion). Anyway, a backup never hurts.
2. Create a boot-disk using the OnTrack-Manager (`dm.exe`).
3. Start the OnTrack-DiskManager from floppy drive A:

   ```
   C:\> dm /m /y-
   ```

   The parameter '`/m`' causes the manager to start in manual mode. Now there are some more functions available. The parameter '`/y-`' causes not to read the Dynamic Drive Overlay. The '`-`' is important since otherwise the OnTrack-Manager will start anyway.

   - Choose Maintenance Menu

- Choose UnInstall Disk Manager
- Now a message appears telling us that all data will be lost (that's what we have the backup for).
  If an EIDE controller has been installed this message might not appear, so try (having a backup makes you feel safe!).
- Go back to the main menu; the manager has been removed

4. Entering the disk parameters into CMOS. If you do not have an EIDE controller or the BIOS only supports up to 1024 cylinders you can only use 504 MB for DOS however big your disk might be (this number varies depending on the calculation; the limit is 1024 cylinders). Now there is a way of letting DOS use the maximum of 504 MB and to assign the rest of the disk to Linux. Linux does not use BIOS functions for recognizing the disk's geometry and therefore knows the exact size.

   Configure the DOS partition: Generally one wants to use the maximum amount available for DOS (504 MB). So you enter the maximum amount of cylinders (1024) and all other parameters into the CMOS. If you do not know the other parameters (there should be a label on the hard disk) you can use the Ontrack-Manager to find out. Just enter 305 cylinders, 4 heads and 17 sectors/track. Then start the DiskManager as described above and choose the item CMOS setup. Then you should select your disk and Automatic CMOS Selection. If the values in CMOS are not optimal you will get a message with the correct values. These values have to be added to CMOS by hand. Now you can format and partition the disk with the DiskManager. This can be done by DOS' fdisk as well but the DiskManager provides higher flexibility and is faster. Anyway, do not forget to pass the `'/y-'` as the OnTrack-Manager will be reinstalled!
5. Install Linux.
   Now Linux can be installed as usual with **setup** and YaST. Linux' **fdisk** automatically recognizes the geometry and you can assign partitions as usual.

Conclusion

All of the steps mentioned above have been performed using a Maxtor 7850AV (850 MB) and everything ran without problems. The most important thing is not to forget the backup and a boot-disk with the OnTrack-Manager. Still, time needed for a backup and a (DOS) restriction to 504 MB is immense. On the other hand you get rid of the OnTrack-Manager, which accelerates booting and disk accesses enormously. Generally buying an EIDE controller or BIOS is the better solution. Perhaps not even a backup of the disk is needed, as removal of the DiskManager might be possible without loss of data. Anyway I have not tried this yet.

8.7.5 Special boot-problems with kernel from 2.0 onwards

Problems

Booting with LILO (from either floppy or hard disk) problems might occur after having loaded quite an amount of the kernel (e. g. S.u.S.E. installation kernel). `:-(`

You can select a kernel at the LILO prompt and this kernel is loaded (some dots are output onto screen) but starting the kernel fails. Before reaching `"uncompressing Linux"` the system crashes with different behaviours.

Possible error messages:

- System reboots
- Sytem just hangs
- `"crc-error"`
- `"no free space"`
- `"Error 0x00"`
- `"Error 0x01"`
- `"incomplete literal tree"`

Thereafter access on the floppy is attempted but the system hangs.

Cause

The cause lies in an combination of big kernels, LILO and faulty hardware. Roughly this concerns 1% of all machines.

We assume that this is due to a faulty BIOS that has problems with the fast memory access.

This problem does *not* occur, if:

- the machine is booted via **loadlin**
- the kernel has been (e. g.) copied to a floppy
  ```
  dd if=/vmlinuz of=/dev/fd0
  ```
  and booted from it
- a smaller kernel that has been created with
  ```
  make zImage
  ```
  (e. g.an older 1.2.13 kernel) and is booted via LILO.

The following BIOS settings did not cause any problems either:

- Disable Internal Cache
- **`DRAM Precharge Wait State`** value: 1 *and*
- **`DRAM Wait Burst Timing`** value: `0x3333`

Solution

First of all a you should be able to install a system. So, if you can neither boot via **loadlin** nor via **setup** you should use an old 1.2.13 boot-disk for installation. If you do not have such a disk at hand, you should change the BIOS settings accordingly.

After a successfull installation there is the question of how you want to boot in future. First you should use the same media as during installation. **loadlin** from DOS should not cause any problems. With a boot-disk you should enter the following parameters:

```
load_ramdisk=0 root=/dev/???
```

where `???` is your root partition (e. g. `/dev/hda1`). Thereafter you should build your own kernel since this can be booted by LILO.

Chapter 9

Printers

While installing Linux with YaST you probably have already done some printer configuration. This chapter now covers the essentials that stand behind the processes. It does not show each aspect but enough to understand the principle that acts behind the printer system.

9.1 Overview: Interfaces, Queues (Spooling)

9.1.1 The parallel ports

Connecting a printer to a Linux system is normally done via a parallel port. These ports are accessible via device files (as usual under Unix). Here are their properties:

File	major	minor	DOS name	IRQ	Port
`/dev/lp1`	6	1	`LPT1`	(7)	0x378
`/dev/lp2`	6	2	`LPT2`	(5)	0x278
`/dev/lp0`	6	0	`LPT3`	(5)-	0x3bc

As a standard, those parallel ports are set to polling mode (the CPU can regulary has to ask them, if there is a data transfer in progress). The systemadministrator '`root`' can

Of course you can only use those devices that really are connected to a physically existing device. At system start up the driver `lp.c` announces which ports have been found. You can have a look at these messages in `/var/log/boot.msg` or by entering **`dmesg`**.

> You can not use a kernel with PLIP (IP over parallel line) and the `/dev/lp?` devices simultaneously. The same applies to other devices connected to the parallel port such as CD-ROMS or ZIP drives.

The PLIP and the lp driver would "fight" for the port (a fix for this problem is currently under construction).

These ports use polling mode by default (CPU has to interrogate regularly if there is a data transfer). The system administrator ('`root`') can switch this to interrupt mode with **tunelp** and perform additional tasks for increasing performance. Usually IRQ 7 is used for `lp1` and IRQ 5 for `lp2`.

For testing the interface you can send the file directly to the printer using:

```
root@earth: # cat text-file >/dev/lp1
```

(only being `root`). For every day's work this is more than useless since there can be many print jobs at once and the printer would collapse. Therefore print jobs are performed via a *queue*.

9.1.2 Spooling mode, Printer queues

Printers are accessed via a *spooling mechanism* as usual under multitasking systems as Linux; i. e. printer jobs are temporarily saved in a file and are processed one after another by a job control program called a daemon. This ensures that many users can send printer jobs *simultaneously* to the printer without risking a conflict. After having sent a printer job the user can commence as usual with his work.

A printer queue consists of:

- an entry in `/etc/printcap`; here it is defined;
- a directory normally under `/var/spool`; here there is a data file and a control file for each printer job.

Each queue is processed by exactly one printer. There can be many printing queues for one printer.

With S.u.S.E. Linux there comes the BSD-spooling-system from University of California at Berkeley which is included in package `lprold`. Traditionally it is designed for use *over a network* and assumes that TCP/IP has been configured and is up and running. Whether queues are set local or remote makes little difference to the configuration.

Processing a print job takes place in three steps:

- The data file is copied to the queues directory and is completed together with a freshly created control file.
- This data file now has to pass a *filter program* which converts it to the printer specific format (e. g.PostScript). This conversion can take up several individual steps.
- The printer-specific file now is queued and printed in order.

9.2 Printer queues: running and Configuring

Let's have a look at the stations a print job has to pass before being printed. And let's also have a look at the programs that take care of the job.

Many of the details we have to skip can be found in the individual manpages of **lpr(1)**, **lpd(8)** and **printcap(5)**.

9.2.1 lpr: Please line up!

lpr is your "interface" to the printer queues on your machine. Generally a job is started using:

```
newbie@earth: > lpr [-P queue] text-file
```

If you omit the option **`-P`** ("Printer", for the queues name) the default is taken from **$PRINTER**. If this is not set either the traditional name **`lp`** is used. This also applies to **`lpq`**, **`lprm`** and **`lpc`** (see below).

lpr checks whether this queue exists by reading `/etc/printcap`. If so, it creates a control file (`cf` file) and a copy of the data file in the spool directory.

If it is a huge job it can be practical if you just link the file instead of copying it. This can be done by **lpr**-Option **`-s`**. Of course you should not write the file until the job has been processed.

lpr allows to set a certain filter (printer driver) by hand (this is hardly ever needed). More can be found in **lpr** and in section 9.3.1

9.2.2 Additional control tools for the user: Overview

Any of these commands (same as with **`lpr`**) accept the **`-P`** option for passing the queue. All details that have to be skipped in this chapter are explained in the corresponding man pages.

- **lpq** shows your own jobs in the queue: For example:
  `newbie@earth: >` `lpq -P queue`

```
queue is ready and printing
Rank   Owner      Job  Files        Total Size
active newbie    676  Hello.txt    259420 bytes
1st    newbie    677  letter.dvi     11578 bytes
2nd    newbie    683  picture.gif      37464 bytes
```

- **lprm** removes own jobs from a queue. For example:
  `newbie@earth: >` `lprm -P queue 676`

```
dfA676Aa05005 dequeued
cfA676Aa05005 dequeued
```

  If no job is given, the current active job is removed if it is your own job.
- **`lpc`** control commands for queues.

9.2.3 lpd: the print manager in background

lpd is launched by `/sbin/init.d/lpd` and runs in the background if the corresponding option **`START_LPD=yes`** has been set in `/etc/rc.config`.

lpd checks `/etc/printcap` to see which printer queues are defined. Its job is to organize the execution of spooled jobs:

- it manages local printer queues: it sends every job's data file through the appropriate filter—which is assigned by the entry in `/etc/printcap` and by an explicit setting in the control file—and to the printer's interface afterwards;
- it takes care of the order of the jobs;
- it checks the queues and printers state (`status`) in the local spool directories;

- it sends jobs to an **lpd** on a remote machine;
- it accepts requests from remote machines for local queues or refuses them if authorization fails.

Authorization: Only requests from hosts who are listed in `/etc/hosts.lpd` are executed (it is also possible to enter the name in `/etc/hosts.equiv` but this has *wide ranged* consequences as far as access and security are concerned and therefore should be avoided). Additional definitions of the queue can restrict access to certain users or users with an account on the local machine.

9.2.4 Filter: the workaholics

The job of the filter programs is to transfer the jobs into a printer specific format. They have to do almost everything. They resemble more or less the *printer drivers* on other systems such as Windows or OS/2. But unlike “drivers” they are also used to count jobs (amount, used up resources ...). See section 9.3.

9.2.5 /etc/printcap: Configuration of the queue

In `/etc/printcap` each queue is defined by an entry of its own (line feed ends the entry). The entry starts with one or more names for the queue (separator |) followed by a list of specifications[1] as `shorthand=`<value> (separator :). “Empty” lines or those starting with an `'#'` are regarded as comments.

There are a couple of example entries (commented) in `/etc/printcap`. A simple entry might look like this:

```
ascii|deskjet:lp=/dev/lp1:sd=/var/spool/ascii:sh:mx#10240
```

File contents 9.2.1: `/etc/printcap`: simple local queue

This queue can be accessed either by `ascii` or by `deskjet`. Its spool directory is `/var/spool/ascii`, its printer `/dev/lp1`. It does not show a header page at the beginning (sh = suppress header) and accepts jobs up to a size of 10240 bytes. Here now is an example for a queue on a remote machine:

```
lp1|HP-4P:\
        :rm=Galois.suse.de:rp=HP:\
        :sd=/var/spool/lpd/lp1:\
        :mx#0:sh
```

File contents 9.2.2: `/etc/printcap`: simple remote queue

(The backslashes `'\'` hide the linefeed — as with shell commands — this is now regarded as only one line.)

[1] all shorthands, their meaning and default settings are described in `printcaps` man page.

The name of the queue is `lp1` and `HP-4P`. Instead of the device (**`lp=`**) it is referred to the remote host `Galois.suse.de` and its queue `HP`.

Setting a filter is not necessary since the queue on `Galois.suse.de` is responsible for the job; so only the spool directory `/var/spool/lpd/lp1` and the size (**`mx#0`** means: none) is defined.

If you install **apsfilter** with YaST it enters three new queues in `/etc/printcap`. Further information under section 9.3.2.

9.3 Printer filter. The apsfilter

9.3.1 What is a filter and how does it work?

As mentioned above the job of a filter is to convert the data file of a job into a suitable format.

Only the papersize, login name, host name (local), name of the file for the count is passed to **lpd**. Being a Unix filter it gets the data file via standardin and sends the printable output through standardout.

Obviously the filter has to know the format of the file (ASCII, DVI, PostScript ...) for converting it. There are two ways of letting it know:

- The filter is "intelligent" so it checks the file and recognizes the format. It is put together of a "pre-filter" for checking the format and many other programs who actually do the conversion.

  This, very roughly, is how **apsfilter** works which is installed by default with your S.u.S.E. Linux.
- Different filters for different file formats. These filters are assigned to a certain queue by entering them in `/etc/printcap`. The user has to give the proper selection if calling **lpr**. There are eight options:

`printcap` Entry	if=	cf=	df=	gf=	nf=	rf=	tf=	vf=
lpr Option		-c	-d	-g	-n	-f	-t	-v

  Traditionally these options[2] are assigned to special file formats (only setting of **lpr**s option is obligatory not the tradition[3]. **`if=`** refers to the standard filter whereas **`df=`** refers to DVIfiles (that come from TEX and LATEX). You can set an own DVI filter for the HP Laserjet 4 by entering:

```
df="/usr/lib/teTeX/bin/i386-linux/dvilj4 -e- -"
```

File contents 9.3.1: DVI filter entry in `/etc/printcap`

9.3.2 apsfilter

A rather comfortable filter, **apsfilter** is included in the package `aps`. **apsfilter** exploits the following circumstances:

[2] see man page of **lpr**

[3] Thus a so called "null-filter entry" `cf=/bin/cat` is allowed and is used with a `lpr -c`

- The standard format for printable files under Unix is PostScript.
- There are many tools that turn other formats, such as tex and pictures into PostScript files (e. g., **dvips** for DVI files and **a2ps** for ASCII files).
- Then there is **Ghostscript** which is capable of transferring PostScript files to many non-PostScript formats for the printer (so to speak a collection of printer drivers).

apsfilter now concludes almost any of these tools including the tools of recognizing a certain format (**file**) and decompression tools to one. It sends the result of the format recognition through the appropriate tools, and, at last, to the printer.

In `/var/lib/apsfilter`[4], the central shell script the following file formats and compression types are available:

- ASCII, DVI, PS, Data (PCL, ...), GIF, TIFF, PBM, Sun Raster, X11 Bitmap
- Compression types: compress, gzip, freeze

apsfilter can either be configured by YaST or with its own menu driven configuration tool **SETUP**.

Configuration using YaST

This should suit your needs in the vast majority of cases. This configuration is done in YaST `'Administration of the system'`
`'Integrate hardware into system'`
`'Configure printers'`
and already has been described in section 4.7.1, page 83. YaST creates three queues in `/etc/printcap`. file contents 9.3.2, page 149 shows this using a Canon BubbleJet 800 as an example.

The three entries are constructed the same, that's why we shortened them. Spool directory (**`sd=`**), log file (**`sd=`**), count file (**`af=`**) and standard filter (**`if=`**) are assigned. To be precise all filters are symbolic links to
/var/lib/apsfilter/bin/apsfilter.

These queues are designed for use as:

- *lp* standard queue: for any file format
- *ascii* for printing files as ascii text, even if the spooling systems supposes another format[5]. Output is done in a format defined in **apsfilter**.
- *raw* for printing files that are already in a printable format; here no conversion is made.

Configuration by hand

A detailed configuration of **apsfilter** can be done by **apsfilter**'s SETUP script, which we will cover now in greater detail.

- Install the package `aps` with YaST.

[4] This used to be `/usr/lib/apsfilter`
[5] e. g. if there are German umlauts

```
# LABEL apsfilter
# apsfilter setup Fri May 16 12:17:54 MEST 1997
#
# APS_BASEDIR:/var/lib/apsfilter
#
lp|lp2|bjc800-a4-auto-mono|bjc800 auto mono:\
        :lp=/dev/lp1:\
        :sd=/var/spool/bjc800-a4-auto-mono:\
        :lf=/var/spool/bjc800-a4-auto-mono/log:\
        :af=/var/spool/bjc800-a4-auto-mono/acct:\
        :if=/var/lib/apsfilter/filter/aps-bjc800-a4-auto-mono:\
        :mx#0:\
        :sh:
#
ascii|lp1|bjc800-a4-ascii-mono|bjc800 ascii mono:\
#   [ ... abbreviated ... ]
#
raw|lp3|bjc800-a4-raw|bjc800 auto raw:\
#    [ ... abbreviated ... ]
#
```

File contents 9.3.2: **apsfilter** Queues in `/etc/printcap`

- Change to `/var/lib/apsfilter` (used to be `/usr/lib/apsfilter`).
- Launch the configuration script by **SETUP**.

```
root@earth: > cd /var/lib/apsfilter
root@earth:/var/lib/apsfilter > ./SETUP
```

The greeting message now prints out some information, telling you that it:

- can enter entries for parallel, serial and remote printers into `/etc/printcap`,
- will create spool directories under `/var/spool/lpd`,
- filter scripts will be linked to `/var/lib/apsfilter/bin`,
- a global configuration file `/etc/apsfilterrc` and printer specific files `/etc/apsfilterrc.<gsmode>` are created.

Now enter ↵ to proceed.

For adding a new printer you should choose '`ENTRY Add/Overwrite/Delete an apsfilter entry`'. Now you should go through the menu items in order.

- '`DEVICE Which printer interface`' lets you choose the interface for the printer and is added in the appropriate submenu. Normally the parallel port is `/dev/lp1` (`/dev/lp2` only applies if there are two parallel ports; `/dev/lp0` defines the parallel port on a **Hercules-graphics card.** Compare section 9.1.1). While configuring a serial port you should make sure not to use the mouses device; furthermore you have to enter the baud rate.
- '`PRINTER Which printer driver`' lets you select either of a couple of printers: PostScript, DeskJets, others and remote. See section 9.5.

Now we want to refer you to the abundant configuration file `/etc/apsfilterrc` which also contains some special settings as control sequences for printers, individual DVI filters, Ghostscript resolutions and many more.

9.4 Some words on Ghostscript

If you don't own a PostScript printer, **Ghostscript** is the most popular choice for a filter. Ghostscript accepts PostScript files and contains many printer drivers for conversion. Ghostscript (**gs(1)**) is a rather extensive tool with a lot of command line options. If you invoke Ghostscript directly it presents you its own dialog `GS` after having done the command line that can be left by entering `quit`. Unfortunately there is not enough space for covering all the gory details[6].

Anyway, we want to point out the very helpful help command

```
newbie@earth: > gs -h | less
```

which lists all possible options as well as — important! — version number and the *current list of supported devices*. This list (by time of going to press of this book) is to be found under section 9.5.

If you encounter any difficulties it might be helpful to invoke Ghostscript with a printable PostScript (.ps) file and to send the constructed printer specific data directly to the device. A summary of suitable PostScript files can (e. g.) be found under `/usr/share/ghostscript/<version>/examples` or
`/var/lib/apsfilter/test`.

The command for invoking Ghostscript e. g. for the included printer driver necp6 using a 360 × 360 resolution, and sending it to the printer might look like:

```
newbie@earth: > gs -q -dNOPAUSE -dSAFER -sDEVICE=necp6 \
       -r360x360 -sOutputFile=testfile.lpr testfile.ps
GS>  quit
newbie@earth: > su
root@earth: # cat testfile.lpr >/dev/lp1
```

Hint: in `/usr/share/ghostscript/<version>/doc` helpful documentation can be found (e. g.in `devices.txt`).

9.5 List of supported printers

Besides PostScript and HP-Deskjet printers lots of other printers are supported by Ghostscript (version 4.03):

Apple

Table 9.1: to be continued...

[6] A first overview can be found in Ghostscript's man page. Fortunately **apsfilter** releases you from the tedious construction of command line options.

appledmp	Apple Dot Matrix Printer (even Imagewriter)
iwhi	Apple Imagewriter, high resolution
iwlo	Apple Imagewriter, low resolution
iwlq	Apple Imagewriter, 320x216 dpi
Canon	
bj10e	Canon BubbleJet 10e
bj200	Canon BubbleJet 200
bjc600	Canon BubbleJet 600c, 4000c (Colour)
bjc800	Canon BubbleJet 800c (Colour)
lbp8	Canon LBP-8II
lips3	Canon LIPS III
DEC	
declj250	DEC LJ 250
la50	DEC LA50
la70	DEC LA70
la75	DEC LA75
la75plus	DEC LA75 Plus
lj250	DEC LJ250
ln03	DEC LN03
Epson	
ap3250	Epson AP3250
eps9high	Epson FX-80-compatible, 240 dpi
eps9mid	Epson FX-80-compatible, 120 dpi
epson	Epson FX-80-compatible 9 or 24 dot-matrix printer
epsonc	Epson LQ-2550, Fujitsu 1200/2400/3400, Colour prints
epscp2	Epson ESC/P2
st800	Epson Stylus 800
stcolor	Epson Stylus Color
Hewlett-Packard	
cdeskjet	HP DeskJet 500C, black
cdj500	HP DeskJet 500C, 540C
cdj550	HP DeskJet 550C, 560C
cdjcolor	HP DeskJet 500C, colour
cdjmono	HP DeskJet 500C, black
deskjet	HP DeskJet, HP DeskJet Plus

Table 9.1: to be continued...

djet500	HP DeskJet 500
djet500c	HP DeskJet 500c
dnj650c	HP DesignJet 650C
laserjet	HP LaserJet
ljet2p	HP LaserJet IIp
ljet3	HP LaserJet III
ljet3d	HP LaserJet IIID
ljet4	HP LaserJet IV
ljetplus	HP LaserJet Plus
lj4dith	HP LaserJet IV, ditherd
lp2563	HP 2563B LinePrinter
cp50	HP PaintJet 300XL
paintjet	HP PaintJet Colour printer
pj	HP PaintJet XL, Alternative
pjetxl	HP PaintJet 300XL
pjxl	HP PaintJet 300XL
pjxl300	HP PaintJet 300XL, HP DeskJet 1200C
IBM	
ibmpro	IBM Proprinter, 9 dot-matrix printer
jetp3852	IBM-Jetprinter 3852
OKI	
oki182	OKI MicroLine 182
okiibm	OKI MicroLine, IBM -compatible
Tektronix	
t4693d2	Tektronix 4693d, colour
t4693d4	Tektronix 4693d, colour
t4693d8	Tektronix 4693d, colour
tek4696	Tektronix 4695/4696
Andere	
imagen	Imagen ImPress
m8510	C.Itoh M8510
necp6	NEC P6, P6+, P60 bei 360 dpi
nwp533	Sony Microsystems NWP533 Laser printer (only Sony)
oce9050	OCE 9050
r4081	Ricoh 4081 Laser printer
sj48	StarJet 48

Table 9.1: to be continued...

sparc	SPARCprinter
xes	Xerox XES (2700, 3700, 4045)

Table 9.1: Supported printers (Ghostscript v. 4.03)

Of course any HP4 compatible PostScript printers are supported too.

During installation of **apsfilter** or during configuration using YaST the line given in the left column has to be set as printer device (e. g.djet500).

If the printer is not supported this does not necessarily mean that the **apsfilter** can not be used at all. In this case try a similar printer. Quite often this driver can be used.

A quite common error is setting the wrong resolution. If this is the case the correct value (the DPI rate the printer accepts) can be set in **$GS_RESOL** in `/etc/apsfilterrc` (e. g.**`GS_RESOL=360x360`**. Some entries (commented) do already exist).

9.6 Printer checklist: apsfilter

- Has package `aps`, series been installed for **apsfilter**?
- Has **apsfilter** been configured by YaST or by `/var/lib/apsfilter/SETUP`
- Does the kernel contain TCP/IP support? (check with **`dmesg`**)
- Have package `nkita` and package `nkitb` been installed?
- If the printer does not support the default setting of 300×300 dpi: has the correct resolution been set according to your printers manual by YaST or been entered in `/etc/apsfilterrc`
- What does **`lpc -status`** say?
- Try **`lpc up all`**
- At least the **`raw`** queue should work on any printer and enabling you to print ASCII files.
- Did you install the correct port (comp. section 9.1.1)?
- The kernel should *not* contain a PLIP driver (**`dmesg`**).

Chapter 10

Linux-machines and hardware

10.1 Preliminary notes

Meanwhile it is possible to integrate most PC components more or less easy into a Linux system. How to handle this in detail und which software is available can be found in this chapter[1]. Solutions for problems with "printings" ref. chapter 9. Some hints concerning PCMCIA-Hardware are to be found in chapter 6.

10.2 ISA "Plug and Play" hardware

10.2.1 General

In the meantime so called "PnP" (Plug and Play) are available (some people refer to them as "Plug and Pray"). Anyway. These adapters have been designed fora automatic recognition. The user should not bother with configuration. This might apply in the "Windows world". It used to be quite a problem for Linux to access these cards unless you were able to disable PnP (which adapter does have jumpers?)

10.2.2 Configuration

There are two tools for configure these adapters:

- **pnpdump**
- and **isapnp**

The first one is for creating the configuration file `/etc/isapnp.conf`. The latter reads this configuration file and sets up the PnP device.

10.3 Changeable media

10.3.1 General

Lots of "changeable media" may be used on Linux. These in particular are: disk drives, ZIP drives, JAZ drives, SyQuest drives, MO (magneto optical) drives – everything is possible.

[1] At least this is a try `;-)`

10.3.2 Disk drives

For accessing disks you need to enable "floppy" support in the kernel (see section 13.4.5, page 187).

You may access MS-DOS disks via the **mtools**. Usually one assigns a `minix` on disks. These may be mounted as usual.

10.3.3 LS 120 drives

LS 120 are ATAPI devices and are supported by the IDE driver . These devices may be integrated just as (E)IDE hard drives.

10.3.4 ZIP drives

There are different ZIP drives for parallel ports, ATAP as well as SCSI.

If you plan to use the parallel port versions you will need SCSI hard disk support (see section 13.4.7, page 192) as well as the `ppa` driver. The `ppa` may be found under the "low-level" SCSI drivers.

For avoiding conflicts with the printer you should activate the `ppa` as a module. Thus it may be loaded on demand using **`modprobe`** (see section 13.2, page 181). Now you may access ZIP disks as well as SCSI hard drives.

For the ATAPI version there applies section 10.3.3, page 156. For SCSI versions section 10.3.5, page 156.

10.3.5 SCSI changeable media drives

Magneto optical drives (MO), SyQuest drives, ZIP and JAZ drives attached to a SCSI interface are considered as normal SCSI devices.

10.4 Scanner

10.4.1 General

Meanwhile both software and drivers are available for addressing scanners under Linux. For being able to use this software, the scanner should be connected to the SCSI port and the proprietary delivered controllers should be avoided.

> The software **SANE - Scanner Access Now Easy** is still only available in a beta phase; for that reason system crashes and more can not be excluded.

From S.u.S.E. Linux5.2 on you will be able to install a scanner with YaST. Be aware that the scanner uses the generic SCSI devices. That is it directly accesses the scanner on a low level format. So, if you are not familiar with these devices make sure you read the instructions and READMEs *before* you change the settings!

10.4.2 Hardware: Flatbedscanner, Cameras, automatic document feed ...

For using your scanner under Linux, a matching "Scanner Device File" needs to installed. Under Linux, scanners are addressed as "Generic SCSI Devices" using the device drivers `/dev/sg0` to `/dev/sg6`; under S.u.S.E. Linux however only the generic SCSI drivers `0-3` come as a standard. In case the scanner should be linked to the generic device `/dev/sg4` or higher, you have create your own device. Please refer manpage of **sane-scsi** (**man sane-scsi**) and manpage of **mknod** (**man mknod**).

Set your SCSI-Scanner to the end of the SCSI-chain. In that case it is easy to determine the device - just count all the SCSI devices you have connected to the chain.

Example: You have connected an SCSI-harddrive, a CD-ROM-drive and an SCSI-scanner to the SCSI-bus. This makes up three SCSI-device. The conclusion is, since the scanner is linked up at the end of the chain, that the right device driver is `/dev/sg2` (Attention: the count starts at `0`), the third generic SCSI-device.

Now place a link from that device to `/dev/scanner`:

```
root@earth: # ln -s /dev/sg2 /dev/scanner
```

Afterwards you have to place the properties for the generic SCSI-Device. SANE also expects write properties:

```
root@earth: # chmod -m 777 /dev/sg2
```

Another option is to create a group `'scanner'` (e. g. mit YaST) and to locate only those users to the group which are allowed to use the scanner. Of course you have to execute:

```
root@earth: # chown root.scanner /dev/sg2
```

at the end.

In any case you should be able to access the scanner right now.

10.4.3 Software

Install the package `sane`[2]. After the installation process, please make sure to study the `README`-files located in `/usr/doc/packages/sane`! You then get an executable program **xscanimage**. When you start this program, it should be able to recognize your scanner if it belongs to the supported ones.

The graphics package "the GIMP" (package `gra`, series `gimp`) is designed that way – and realized with the same library – that SANE is able to recognize it as "plug-in". For managing this,you have to create the following link *after* the installation of GIMP (the path `\tld/.gimp` must exist already; it is automatically created at the first start of **gimp**):

```
newbie@earth: > cd ~/.gimp/plug-ins
newbie@earth:~/.gimp/plug-ins > ln -s /usr/bin/xscanimage .
```

Now the SANE-Tool is located automatically and is displayed in the menu `'Xtns'` of "the GIMP". If you now scan an image, it is loaded directly into GIMP and you can access and work with it.

[2] see. `ftp://ftp.mostang.com/pub/sane/`.

10.4.4 Problems

In case it is not possible to locate the scanner, please check if it was located by the SCSI-controller. This can be done e. g. via:

```
root@earth: # cat /proc/scsi/scsi
```

and then you should see line similar to:

```
Attached devices:
Host: scsi0 Channel: 00 Id: 00 Lun: 00
  Vendor: SEAGATE  Model: ST32550N       Rev: 0016
  Type:   Direct-Access                  ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 01 Lun: 00
  Vendor: PIONEER  Model: CD-ROM DR-U10X Rev: 1.07
  Type:   CD-ROM                         ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 02 Lun: 00
  Vendor: QUANTUM  Model: FIREBALL_TM3200S Rev: 300N
  Type:   Direct-Access                  ANSI SCSI revision: 02
Host: scsi0 Channel: 00 Id: 06 Lun: 00
  Vendor: SCANNER  Model:                Rev: 2.00
  Type:   Scanner                        ANSI SCSI revision: 01 CCS
```

If you don't get a prompt at this point, the SCSI-controller has not been located. In this case, please read in the manual section 13.4.7 on how to integrate your controller.

Should besides the existence of a scanner **xscanimage** not able to recognize your scanner, the reason for the problem can be that

- the scanner is not getting support yet or
- that you have to transfer certain parameters to xscanimage.

Both cases are handled in the `README` files that came with the package. Generally one can say that most HP-scanners meanwhile are supported; Mustek-scanners however not in all cases. Some of them perform quite well, others not at all. Of the newer Umax-scanners the majority is supported. We're currently working on the support for older Umax-scanners.

Chapter 11

Special features of S.u.S.E. Linux

11.1 Keyboard layout

For unifying keyboard mapping of certain programs the following files had to be adapted:
`/usr/lib/X11/Xmodmap`
`/etc/inputrc`
`/etc/skel/.exrc`
`/etc/skel/.less`
`/etc/skel/.lesskey`
`/etc/csh.cshrc`
`/etc/termcap`
`/usr/lib/terminfo/x/xterm`
`/usr/lib/X11/app-defaults/XTerm`
`/usr/share/emacs/19.34/site-lisp/term/*.el`
`/usr/lib/joerc`
These changes only apply to applications that make use of **terminfo** entries or change their configuration files (**vi**, **less**, etc.). Every "none" S.u.S.E. application should be adapted.

11.2 Differences in the structure of the file system

S.u.S.E. Linux tries to be File System Standard conform (FSSTD), therefore paths of some files had to be changed:

- Lock files are to be found under `/var/lock`.
- Log files (boot messages, warnings, output of **pppd** and many more) have been moved to `/var/spool/uucp/Log` as well as to `/var/spool/uucp/Stats`.
- The UUCP log files are know located under `/var/spool/uucp/Log` as well as under `/var/spool/uucp/Stats`.
- The directory `/usr/data` has been replaced by `/usr/share`; all pictures and sounds have been shifted there.
- **cron** tables are now located under `/var/cron/tabs` instead of `/var/lib/cron`. Since the introduction of S.u.S.E.Linux 5.0 , the file **/etc/crontab** has been configured as a systemwide library; additionally you must insert after the time setting, under which user the specific order should be handled.

11.3 linuxrc

linuxrc is started during the boot-up of the kernel, before the “real” booting commences. [1].

This allows you to boot a small modularized kernel and to load the required drivers as modules. It is (at the moment) still possible, to include all drivers which the kernel supports und which are needed for the installation (including PCMCIA) on one disk.

linuxrc is your assistant for loading all relevant hardware drivers. You can even use **linuxrc** as a boot-disk for an already installed system (e. g.as a rescue-disk). You can even start a totally independent RAM-disk based rescue system (e. g., if something severe happened to your hard disk or you have simply forgotten your root password. More in section 11.4.

11.3.1 Main menu

After you have selected language, screen and keyboard you find yourself in **linuxrc**’s main menu.

Figure 11.1: Main menu of **linuxrc**

You start the installation with `‘Installation / start-up system’`. Whether you can go there directly mainly depends on your hardware:

If all components that are needed for installation have already been recognized by the kernel you do not need to load additional drivers. This mainly applies to those that only have (E)IDE adapters (and of course only (E)IDE hard disks and CD-ROM).

If there is a SCSI adapter installed which is necessary for installation[2] you have to load the corresponding SCSI module. The same applies if you want

[1] Of course the kernel has to be configured (section 13.4.5)
[2] An adapter with only a scanner connected to it is not necessary at boot time

to install via an existing network. Here the suitable module has to be loaded first.

Furthermore there are a lot of older CD-ROMs that are driven by proprietary controllers and which therefore need their own kernel module. If PCMCIA devices are connected to a laptop you need modules as well.

11.3.2 System information

If you are not sure about your hardware the boot messages might help you,

You can check some system information under `'System information'`. Here you can check the used interrupts, I/O ports, used main memory and recognized PCI devices as detected by Linux.

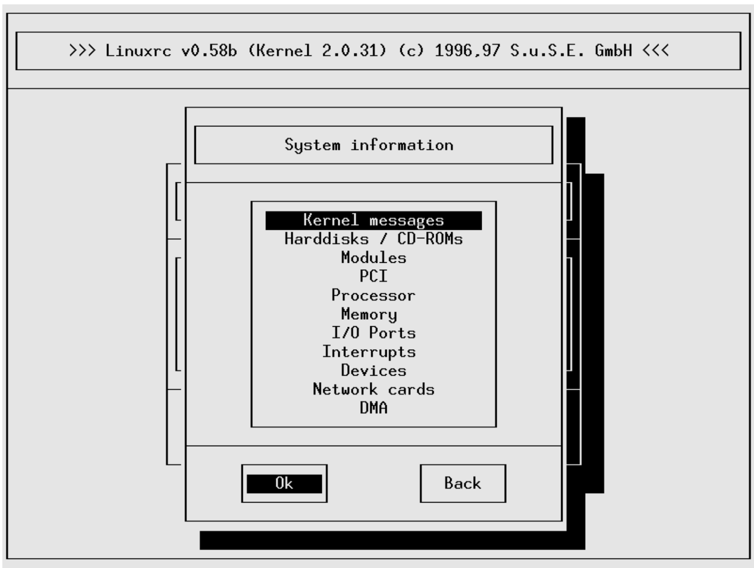


Figure 11.2: System information

The next lines show how a hard disk and a CD-ROM connected to an (E)IDE controller announce their start. In this case you do not need to load additional modules:

```
hda: ST32140A, 2015MB w/128kB Cache, LBA, CHS=1023/64/63
hdb: CD-ROM CDR-S1G, ATAPI CDROM drive
Partition check:
 hda: hda1 hda2 hda3 < hda5 >
```

If you booted a kernel that already has a SCSI driver integrated you of course do not need this SCSI driver as a module either. Quite typical announcements when loading SCSI adapters and connected devices might resemble:

```
scsi : 1 host.
Started kswapd v 1.4.2.2
scsi0 : target 0 accepting period 100ns offset 8 10.00MHz FAST SCSI-II
```

```
scsi0 : setting target 0 to period 100ns offset 8 10.00MHz FAST SCSI-II
  Vendor: QUANTUM   Model: VP32210          Rev: 81H8
  Type:   Direct-Access                     ANSI SCSI revision: 02
Detected scsi disk sda at scsi0, channel 0, id 0, lun 0
scsi0 : target 2 accepting period 236ns offset 8 4.23MHz synchronous SCSI
scsi0 : setting target 2 to period 248ns offset 8 4.03MHz synchronous SCSI
  Vendor: TOSHIBA   Model: CD-ROM XM-3401TA  Rev: 0283
  Type:   CD-ROM                            ANSI SCSI revision: 02
scsi : detected 1 SCSI disk total.
SCSI device sda: hdwr sector= 512 bytes. Sectors= 4308352 [2103 MB] [2.1 GB]
Partition check:
 sda: sda1 sda2 sda3 sda4 < sda5 sda6 sda7 sda8 >
```

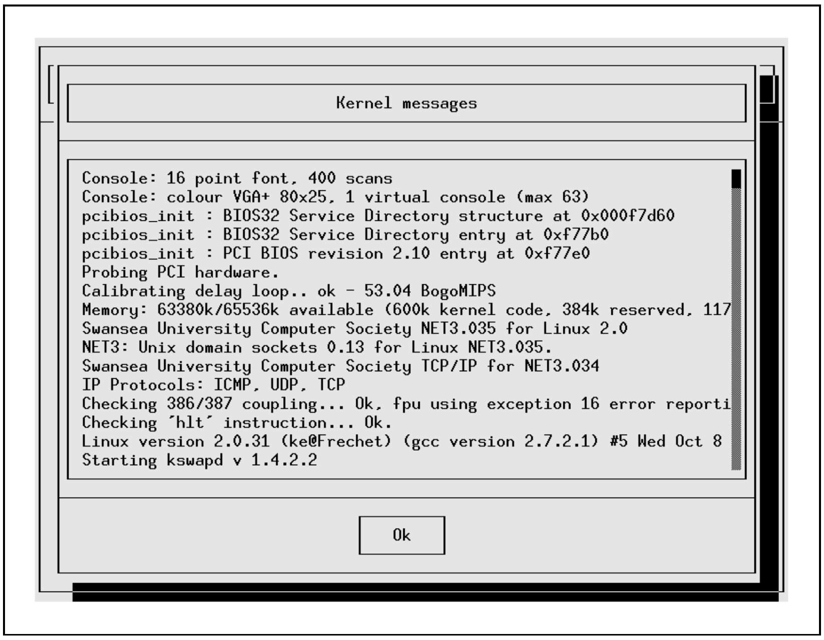


Figure 11.3: Kernel messages

11.3.3 Loading of modules

You select which kind of modules you need. If booted via disk the corresponding data has to be read by **linuxrc** and displayed in a list.

If you have booted from CD or from DOS (via **loadlin**) these modules are already set in **linuxrc**. This saves the tedious loading but needs additional memory. If your machine is supplied with less than 8 MB of RAM you have to boot from disk.

linuxrc offers you a list of available drivers. On the left there is the name of the module and on the right you can see a short message telling you what it can be used for.

For some components there are various drivers at your disposition (or newer alpha drivers). These are offered as well.

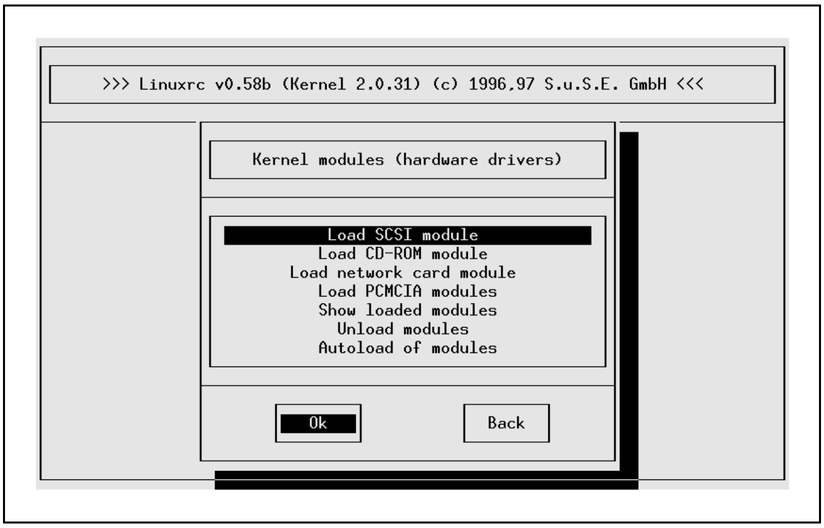


Figure 11.4: Load modules

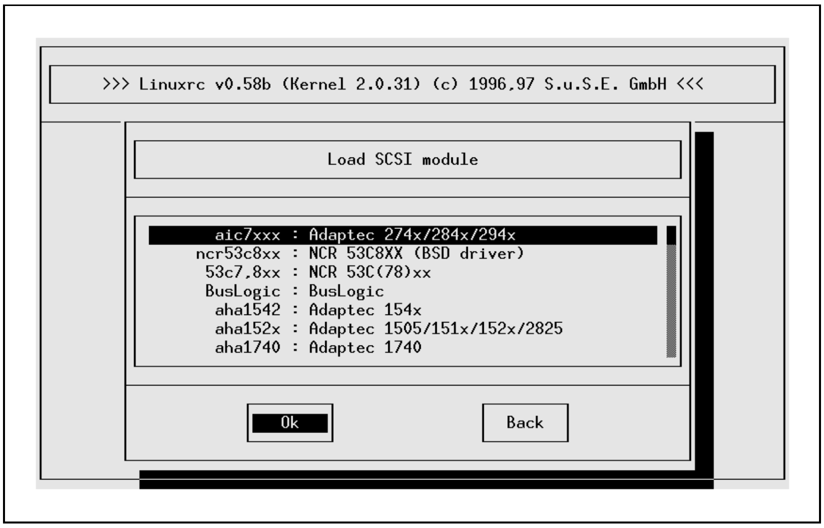


Figure 11.5: Selection of SCSI-drivers

11.3.4 Passing parameters

If you have found a suitable driver, move to it with the cursor and press ↵. Now there is a mask where you can add additional parameters for this module. More on module-parameters can be found in section 7.3.3, page 111.

We would just like to point out that on the contrary to the LILO prompt parameters for the same module have to be separated by blanks.

In most cases it is not necessary to specify the hardware in detail; most drivers find their components automatically. Most networking cards and proprietary CD-ROM drives however need parameters. In case of doubt just try ↵.

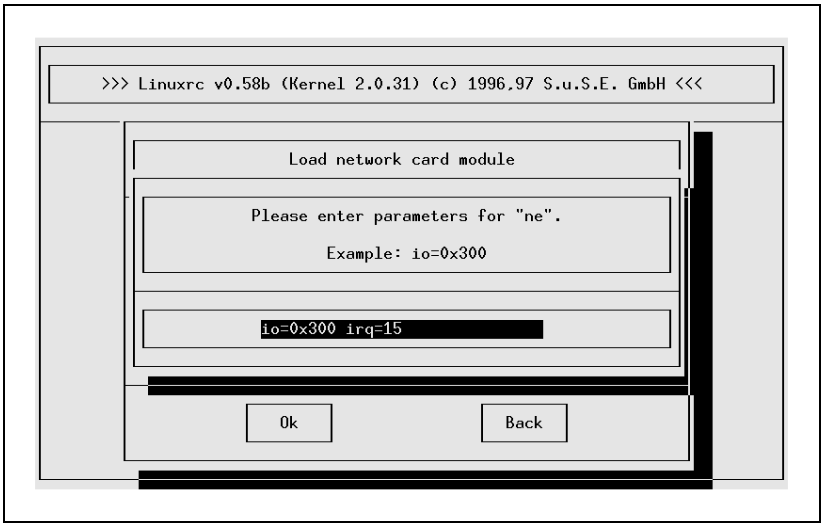


Figure 11.6: Entering parameters for loading a module

Recognizing and initializing of certain hardware can take quite some time. Switching to console 4 (Alt + F4) lets you watch the kernel messages while loading. SCSI drivers need quite some time as they have to wait for each device to load.

If loading succeeded the messages are displayed by **linuxrc** just for make sure everything ran smoothly. Otherwise (if it fails) the messages might give you a hint why it failed.

11.3.5 System / Start installation

If you achieved hardware support via modules you can switch to `‘System / Start installation’`.

There are many items to choose from for installation and/or rescue system:

11.4 The S.u.S.E. rescue system

11.4.1 Overview

Since version 4.2, S.u.S.E. Linux contains a completely independent rescue system which enables you to access your system (“from outside”) in a case of emergency. This system consists out of a special selection of system tools. They should be sufficient to solve most hardware and configuration errors and problems.

The rescue system consists of a boot-disk (the same as for the new-installation) and an installation CD. Since this boot-disk can be created at any time by means of the CD (the boot-disk can be found in `disks`) this serves as a rather secure and easy to handle rescue system. If you copy this file using the command

```
root@earth: # /sbin/badblocks -v /dev/fd0 1440
root@earth: # dd if=/cdrom/disks/rescue of=/dev/fd0 bs=18k
```

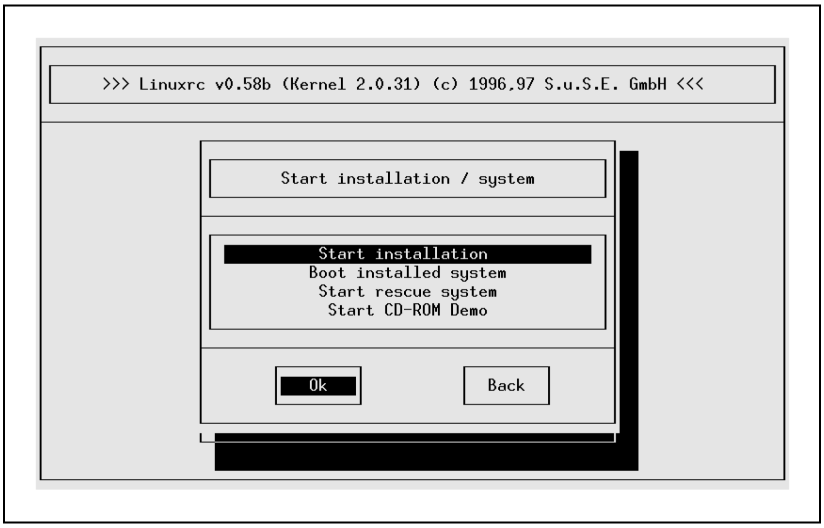


Figure 11.7: Target of **linuxrc**

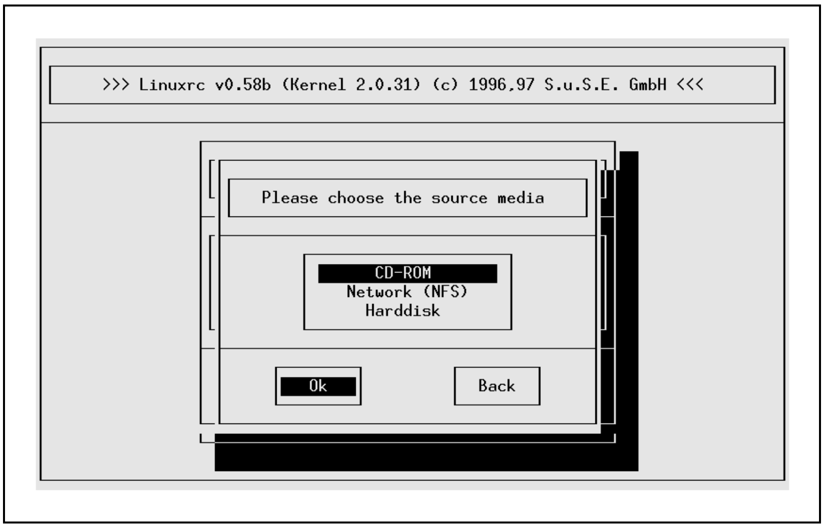


Figure 11.8: Selecting source media in **linuxrc**

or using the DOS command (provided `Q` is your CD-ROM under DOS)

```
Q:\dosutils> rawrite.exe
```

onto a 2nd error-free disk ("rescue-disk") you also can launch the rescue system from the boot-disk and this rescue disk.

Please be aware that you cannot mount the rescue disk because it does not contain a file-system but a compressed image (the uncompressed image would take up to 3.5 MB which would not fit on a disk). If you want to have a look at it you have to decompress it in advance and then mount it (the name has to be '`root`'). Provided that your Linux kernel supports the *loop-device* you enter:

```
root@earth: # /bin/cp cdrom/disks/rescue /root/rescue.gz
root@earth: # /bin/gunzip /root/rescue.gz
root@earth: # /bin/mount -t minix -o loop /root/rescue /mnt
```

Now you can have a look at it under `/mnt`.

> Always have *some* boot-disks at hand! Creating such a disk does not take long; well, anyway, it takes less time than searching for a disk in case of emergency (in such a situation you can be sure that your CD-ROM does not work either (Murphy's law...))!

11.4.2 Launching the rescue system

The rescue system, like as the normal installation, is launched using the S.u.S.E. boot-disk. Step by step:

- *Requirements:* The floppy drive is bootable (if not so, please run CMOS setup to modify the settings).
- Launch system with the S.u.S.E. boot-disk.
- Enter language, keyboard etc. until you get to the main menu.
- Now select `'Installation/Start system'`.
- Insert the CD or the disk containing the compressed image of the rescue system.
- In `'Installation/Start system'` you should choose either, depending on the source media: `'Load rescue system from CD'`, or. `'Load rescue system from disk'`.

The rescue system is now being decompressed and loaded into a RAM-disk as new root-file system, mounted and started. Now it is ready for use.

11.4.3 Working with the rescue system

The rescue system provides three virtual consoles on keys Alt + F1 to Alt + F3 . Here `'root'` may log in without a password. Alt + F4 brings you to the system console where you can view the kernel and syslog messages.

A shell and lots of other useful utilities (net-tools) can be found under `/bin`. In `sbin` you cand find **efsck** which is very useful for checking and repairing file systems.

In `sbin` there are the important binaries for system maintenance, such as **fdisk**, **mkfs**, **mkswap**, **mount**, **mount**, **init**, **shutdown**, as well as **ifconfig**, **route**, **netstat** for maintaining the network.

As editor, **vi** is located under `/usr/bin`; also additional tools like (**grep**,**find**,**less**, etc.) and, most important of all, **telnet** are avaiable.

Example: Accessing your normal system

For mounting your Linux system, you should use mountpoint `/mnt`. Of course you can also use or generate another directory.

$PATH already contains the standard directories and subdirectories as `/mnt/bin` and `/mnt/usr/bin`. Consider the following example: Provided your

```
/dev/sdb5      swap      swap      defaults   0   0
/dev/sdb3      /         ext2      defaults   1   1
/dev/sdb6      /usr      ext2      defaults   1   2
```

File contents 11.4.1: Example: `/etc/fstab`

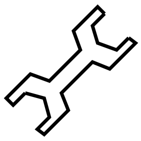
system is structured (according to `/etc/fstab`) as given in file contents 11.4.1, page 167:

Then you may mount it step by step to `/mnt`; just execute the following commands in order:

```
root@earth:/ # mount /dev/sdb3 /mnt
root@earth:/ # mount /dev/sdb6 /mnt/usr
```

Now you can access your whole system and (e. g.) correct mistakes in configuration files (as `/etc/fstab`, `/etc/passwd`, `/etc/inittab`). Of course those files now are located under `/mnt/etc` instead of `/etc`!

> *Every* experienced Linux user does a hard copy of `/etc/fstab` and of the output of `fdisk-l` to his "files". Even completely corrupt partitions can be reassigned if the *exact* geometry and the parameters are known!



Example: Repairing file-systems

Damaged file-systems are tricky tasks for the rescue system. This might happen after a failed shutdown (current down) or a system crash. Generally, file-systems can not be repaired on a running system. If you encounter really severe problems you might not even be able to mount your root-file-system and the system start ends up in a `"kernel panic"`. Here the only chance is to repair the system from the "outside" using a rescue system.

The S.u.S.E. Linux rescue system contains the utilities **e2fsck** and, for the diagnosis, **dumpe2fs**. These should cover most problem. In an emergency there normally are no man pages available, that is why we included them in this manual under appendix E, page 405.

Example:

If mounting a file-system fails due to an *invalid* superblock, then **e2fsck** would most probably fail too. If this is the case your superblock might be corrupted too. There are copies of the superblock located every 8192 blocks (8193, 16385 . . .) If your superblock got corrupted you might want to try one of the copies instead. This is achieved by entering the command:

```
root@earth: # e2fsck -f -b 8193 /dev/damaged_partition
```

The **-f** option forces the file-system check and overrides **e2fsck**'s error so that — since the superblock is intact — everything is fine.

11.5 init

The init scripts `/etc/rc.d`, which so far have regulated the start up of the system, have been substituted through a system which allows an orderly administration of the different Runlevels similary to **System V**. Since the filesystem-standard does not allow exectutable files under `/etc`, the **init**-scipts have been located under `/sbin/init.d`.

11.6 Changes made to software packages

11.6.1 package `curses`

On our S.u.S.E. Linux CD we newly included the package package `ncurses`. The corresponding libraries are named `libncurses.so.<xx>`. This means that some Makefiles have to be adapted concerning the linker. Packages of your own should be linked with the command **`-lncurses`** instead of **`-lcurses`**. For those of you who want to do that anyway try

```
-I/usr/include/termcap -I/usr/include/curses
-L/usr/lib/termcap -L/usr/lib/curses
```

11.6.2 manpages

Some man pages (e. g. **tar**) are not maintained anymore. They have been replaced by info files. **Info** (**`info`**) is **GNU**'s hypertext system. Typing **`man info`** gives you first help. **`info`** can be launched via **`emacs -f info`** or standalone: **`info`**. The most comfortable way is **`xinfo`**.

11.6.3 sendmail

Since **GNU dbm** has been replaced by **BSD db** the commands for installing and creating mail tables have slightly changed. Instead of entering

```
makemap dbm mailertable
```

you have to type

```
makemap hash mailertable
```

11.7 Misc

The tasks of `/etc/securetty` have been transferred to `/etc/login.defs`.

Chapter 12

The S.u.S.E. boot concept

Booting and initialisation of a Unix system challenges even an experienced system administrator. This chapter gives you an short overview of the S.u.S.E. Linux' boot concept.

This concept is much more complex but also more flexible than other Linux' concepts. It is based on a modern boot concept of a **System V** workstation as described in [Frisch, 1993].

The simple words `"Uncompressing Linux..."` show that the kernel, takes over control over all your hardware. It checks and sets your console[1] for reading BIOS settings and to initialize basic hardware interfaces. Next your drivers "probe" existing hardware for initializing it accordingly. After partition check and mounting of root-filesystem [2] the kernel starts **/sbin/init** which starts the main system with all its programs and configurations. The kernel will maintain the whole system, including CPU time of programs and hardware access.

12.1 The init program

The program **/sbin/init** is responsible for a correct initialization for all system processes; thus being the father of all processes in the whole system. **Init** takes up a special position: **Init** is directly started by the kernel and is resistant against signal 9 which normally enables you to kill processes. All further programs are either started directly by **init** or from one of its "child" processes.

Init is centrally configured via `/etc/inittab` file; here the so-called "run-levels" are defined (more about run-levels in the next chapter); here is set what should happen in the several levels. Depending on entries in `etc/inittab` several scripts are started by **init** which, for clarity, all reside in the same directory `/sbin/init.d`.

The whole starting up of the system (of course the shutting down as well) is therefore maintained by **init**; from this point of view the kernel can be considered as a "background process" whose task it is to maintain all other processes and to adjust CPU time and hardware access according to requests of other programs.

[1] precisely BIOS registers of graphic cards and output format.
[2] assigning root partition to "`/`"

12.2 Runlevels

Under Linux there are so-called *run levels* which define how the system is going to be started up. After booting the system starts as defined in `/etc/inittab` in line **`initdefault`**. An alternative is assigning a special runlevel at boot time (e. g.at the LILO-prompt): the kernel passes parameters which it doesn't need directly to **init**. For changing run-levels while the system is up and running you can just enter **init** with the appropriate number. e. g.

```
root@earth:/ > init S
```

brings you to *single user mode* which is for maintenance and administration of your system. After having finished work in *S* mode the system administrator can change run-level to 2 again by typing

```
root@earth:/ > init 2
```

Now all essential programs are started and users can log in and work with the system. The table below gives you an overview about available run levels [3]:

Runlevel	Meaning
0	Halt
S	Single User Mode
1	Multi User without network
2	Multi User with network (Standard)
3	Multi User with network and xdm
4	Unused
5	Unused
6	Reboot

Table 12.1: List of Linux' valid runlevels

Thus leading to the fact that you can halt the system using

```
root@earth:/ > init 0
```

or restart it with

```
root@earth:/ > init 6
```

respectively.

If you have already installed and configured the X-Window System properly (Section section 14.1) and want the users to log in via a graphical user interface you can easily change runlevel to `3`. You should give it a try first by typing

```
root@earth:/ > init 3
```

to see whether the system works as you expected.

[3] Runlevel 1 should not be used on a system whose `/usr`-partition is mounted via NFS!

> With a damaged `/etc/inittab` you can end up in a system which can not be brought up properly! Therefore be extremely careful while editing `/etc/inittab`!

12.3 Changing of runlevels

Generally there are a couple of things that happen if you change runlevels. First, so-called *stop-scripts* of the current runlevel are launched (closing down some programs which are essential for the current runlevel); thereafter *start scripts* of the new runlevel are started. Here, in most cases, some programs will be started.

For more clarity we will show you a change from runlevel 2 to 3:

- The administrator (`'root'`)[4] tells **init** to change runlevels:
  `root@earth:/ >` `init 3`
- **init** now consults its configuration file (`/etc/inittab`) and realizes that it should start `/sbin/init.d/rc` with the new runlevel as parameter.
- now **rc** calls all the stopscripts of the current runlevel; but only for those where there is no start script in the selected new runlevel. In our example these are all scripts which reside in `/sbin/init.d/rc2.d` (old runlevel was 2) and which start with a `'K'` [5]. The number following `'K'` guarantees a certain order to start as there are some dependencies which have to be taken into consideration.
- The last thing to start are the start scripts of the new runlevel; those are (in our example) under `/sbin/init.d/rc3.d` and begin with an `'S'`. The same procedure regarding the order in which they are started is applied here.

If you want to change into the same runlevel which you are already in, **init** only checks `/etc/inittab` for changes and starts the appropriate steps (e. g.starting a getty on another interface).

12.4 Init scripts

Scripts under `/sbin/init.d` are divided into two sections:

- Scripts which are executed directly by **init**: this only applies while booting as well while shutting down the system immediately (power failure or pressing Ctrl Alt Del by the user).
- Scripts which are started indirectly by **init**: This happens while changing the runlevel; here generally **/sbin/init.d/rc** is executed which guarantees the correct order of the relevant scripts.

All scripts are located under `/sbin/init.d`. Scripts for changing the runlevel are also to be found there but are called via symbolic link from one of the subdirectories `/sbin/init.d/rc0.d` to `/sbin/init.d/rc6.d`. This

[4] of course changing of runlevels can only be done by `'root'`

[5] names of stop scripts always start with an `'K'` whereas start scripts always start with an `'S'`

is just for clarity reasons and avoids duplicate scripts (e. g.if they are used in several runlevels). Since every script can be executed as start as well as stop-script those scripts have to "understand" parameters "start" and "stop".

Example An example: While leaving runlevel 2 `/sbin/init.d/rc2.d/K40network` is executed among others; this results in `/sbin/init.d/network` being executed from `/sbin/init.d/rc` with the "stop" parameter. When entering runlevel 3 the same script is started but with the "start" parameter instead of "stop".

Links in those runlevel-specific subdirectories simply serve for allocation purposes of the scripts to a certain runlevel.

Here we give an overview of all scripts being used with a short description:

Boot and shutdown Here we give you a short introduction on the boot and stop scripts that are launched first (or last respectively) as well as an explanation of the maintaining script.

- *boot*
  Is executed while starting the system directly by **init**. It is independent from the chosen runlevel and is only executed once. Here filesystems are checked, the kernel daemon is launched some unnecessary files under `/var/lock` are deleted and the network is being configured for loopback device (if it is selected in `/etc/rc.config`). Furthermore the system time is setup and Plug and Play hardware is initialized by the `isapnp` tools .

  If an error occurs while automatically checking and repairing the filesystem the system administrator can intervene after having entered the root password.

  Directory `/sbin/init.d/boot.d` is assigned to this script; all scripts in this directory are executed while bringing up the system. This is the right place for your own personal extensions which should be executed only once.

  Last to be executed is the script **boot.local**
- *boot.local*
  Here you can enter additional commands to be executed at boot time before changing into a runlevel; it can be compared to `AUTOEXEC.BAT` on DOS systems.
- *boot.setup*
  General settings that have to be performed while changing from *single user mode* to another runlevel.
  Here keyboard maps are loaded and the kerneldaemon is started which takes care of the automatic loading of modules.
- *halt*
  This script is only executed while changing into runlevel 0 or 6. Here it is executed either as *halt* or as *reboot*. Whether the system shuts down or reboots depends on how **halt** is called.
- **rc**
  The script of overriding importance which is executed at each change of

runlevels. It calls the appropriate stop scripts of the current runlevel and the start scripts of the new selected runlevel.

Creating scripts

You may add necessary scripts to this skeleton very easily. A template my be found under `/sbin/init.d/skeleton`. For enabling a script via `/etc/rc.config` it is recommended to create a <START_> variable in this file. Additional parameters should only be added if really needed. (see the `/sbin/init.d/gpm` script for reference).

Now you need to create the links from the correspondent `rc` to your script. This is for making sure that it is launched when changing into this given runlevel (see above section 12.3, page 171 for script names etc.). manpage of **init.d** (**man 7 init.d**) gives you all the needed technical background.

12.5 /etc/rc.config and /sbin/SuSEconfig

Almost every configuration of S.u.S.E. Linuxcan be done via a central configuration file called `/etc/rc.config`. Here a couple of environment variables are set which are (among others) checked by the init scripts; each of the scripts under `/sbin/init.d` executes `/etc/rc.config` as first steps for reading the values of those variables.

Moreover very many configuration files can be generated with `/etc/rc.config`; this is the task for **/sbin/SuSEconfig**. If you change the network configuration (e. g.) the file `/etc/resolv.conf` will be re-generated as it depends on the configuration you have made.

So, if you change `/etc/rc.config` manually you should invoke **/sbin/SuSEconfig** afterwards to make sure all changes are made at the correct places. If you change configuration with YaST you don't have to bother; YaST automatically executes **/sbin/SuSEconfig** and updates your configuration files.

This concept enables you to make basic changes with your configuration without having to reboot your system. Since some changes are rather complex some programs probably have to be restarted for the changes to take effect. This procedure has been explained in network configuration (see section 16.3) where those programs are forced to be restarted using the command:

```
root@earth:/ > /sbin/init.d/network stop
root@earth:/ > /sbin/init.d/network start
```

As you can see you can easily start init scripts by hand.

Generally we recommend the following steps for configuring your system:

- Bring the system into *single user mode*:
  root@earth:/ > `init S`
  As an alternative you can select runlevel 1; here you have the possibility of login in on several consoles:
  root@earth:/ > `init 1`
- Change configuration file `/etc/rc.config` as needed. This can be done by an editor of your choice or by YaST under 'Changing configuration file' (look at section 12.6).

- Execute **/sbin/SuSEconfig** to make the changes take effect. If you have changed **/etc/rc.config** via YaST this is done automatically.
- Bring your system back into the previous runlevel:
  `root@earth:/ > init 2`

This procedure mainly is relevant if you have changed system wide settings (e. g.network configuration); it is not necessary for "small" changes to go into *single user mode*; but it ensures you that all relevant programs are correctly restarted.

12.6 Options of configuration

Now all parameters of the system are described including their default settings. If you don't use YaST for changing `/etc/rc.config` make sure to set "empty" parameters as two quotation marks (e. g. `KEYTABLE=""`) and to surround parameters that contain a blank by quotation marks (parameters consisting of only one word do not have to be quoted). In our description each parameter is given a value for making their settings as clear as possible:

- `ENABLE_SUSECONFIG=yes`
  With this entry you can disable **SuSEconfig** completely. Please don't contact our support if you have trouble configuring your system after having disabled **SuSEconfig** ;-)
- `MAIL_REPORTS_TO=newbie`
  SuSEconfig can mail reports (created by YaST or included in packages) to you. Here you can set the address. If you don't want it, simply set it to `""`.
- `MOUSE=/dev/ttyS2`
  Interface where mouse is connected to. YaST (**SuSEconfig** respectively) creates a link from `/dev/mouse` to the given device.
- `MODEM=/dev/ttyS1`
  Interface where the modem is connected to. YaST (**SuSEconfig** respectively) creates a link from `/dev/modem` to the given device.
- `KEYTABLE=de-latin1-nodeadkeys`
  Defines keymaps.
- `KBD_NUMLOCK=no`
  NumLock on/off.
- `KBD_CAPSLOCK=no`
  CapsLock on/off.
- `KBD_RATE=30`
  Sets the automatic keyboard frequency. Possible values are from twice to 30 times per second. For this change to take effect keyboard delay has to be set as well.
- `KBD_DELAY=250`
  This is the delay whereafter automatic repetition of the pressed key commences. This value is in milliseconds but isn't very accurate. You have to assign `KBD_RATE` as well.

- **FONT=mr.fnt**
  This is the console font. Not all fonts support German umlauts. YaST provides a little window where you can test all fonts and choose the one you like best.
- **GMT=-u**
  If your hardware is set to GMT (*Greenwich Mean Time*) you should set this variable to `-u` [6]; otherwise you leave it "empty". This setting is relevant for automatic changing to summer or winter time respectively.
- **TIMEZONE=Europe/Berlin**
  Your timezone. Is important for automatic switching to summer or winter time respectively
- **START_LOOPBACK=yes**
  Sets if sort of an "mini" network is created by configuring *Loopback*-device. Since rather an amount of programs rely on this functionality it should be set. [7].
- **CHECK_ETC_HOSTS=yes**
  SuSEconfig can do some checks and modifications for `/etc/hosts`.
- **SETUPDUMMYDEV=yes**
  Setup dummy network device. This is useful for non permanent network connections (e. g. SLIP or PPP).
- **CREATE_HOSTCONF=yes**
  SuSEconfig can create and check `/etc/host.conf`.
- **CREATE_RESOLVCONF=yes**
  SuSEconfig can maintain `/etc/resolv.conf`. If set to `yes` *and* one of **SEARCHLIST** and **NAMESERVER** is empty, it is assumed, that no DNS is wanted and `/etc/resolv.conf` will be deleted. `no` simply lets `/etc/resolv.conf` untouched.
- **NETCONFIG=_0**
  Tells how many networking cards (or other net devices) are installed. This is an example for one networking card (they start with 0); for a system with two cards installed it should resemble **NETCONFIG="_0 _1"**. For a system without network it should not be set.
- **IPADDR_0=193.141.17.202**
  IP-address of the first networking card.
- **NETDEV_0=eth0**
  Name of the first network device (normally an ethernet card therefore `eth0`). Other possible settings are `str1` or `plip1`. If there is more than one card installed those additional cards are supplied with the variables NETDEV_1 to NETDEV_3.
- **IFCONFIG_0="193.141.17.205 broadcast 193.141.17.255 netmask 255.255.255.192"**
  Configuration command for the first networking device installed. Those settings can easily be assigned using YaST. If you have more than one card installed just enter the corresponding values in the appropriate variables.

[6] `-u` is an abbreviation for *universal time*.
[7] Of course your kernel must have be compiled with networking support!

- **NETWORK_0="-net 193.141.17.0"**
  Network address for your first card. For use with Point-to-Point connections (e. g. PLIP) YaST will create an entry which resembles the line `"-host 193.141.17.202"`. The address of the PPP partner is set.
- **GATEWAY_0=193.141.17.200**
  Gateway address of the first networking card. All IP packages which can't be delivered locally are sent to this address. This typically is the system which realizes internet connectivity.
- **CLOSE_CONNECTIONS=false**
  If `true` *and* if the "runlevel" is 0 or 6 (halt or reboot), **/sbin/init.d/route** sends **SIGTERM** to the processes.
- **FQHOSTNAME=earth.cosmos.com**
  Fully qualified hostname of your machine.
- **SEARCHLIST=cosmos.com**
  This entry is used for completing a not fully qualified hostname. If, e. g. you give in `venus` it is checked whether `venus.cosmos.com` is a valid address. This variable *has* to be set if you plan to use DNS! At least enter your domain name here. You can enter up to three entries which should be separated by blanks.
- **NAMESERVER=helios.cosmos.com**
  Address of name server which is to be interrogated if a hostname has to be transposed to an IP address, You can enter up to three nameservers which should be separated by blanks. If you plan to use a nameserver **SEARCHLIST** *has* to be set!
- **ORGANIZATION="S.u.S.E. GmbH, Fuerth, Germany"**
  This text appears in every newsposting you send.
- **NNTPSERVER=helios**
  Address of your news server; if you receive your news via UUCP and they are saved locally you should enter `localhost`.
- **IRCSERVER=helios**
  This is the place for your IRC server (*Internet Relay Chat*). Names of the servers should be separated by blanks.
- **START_INETD=yes**
  Controls whether the **inetd**-super daemon should be activated. This daemon reacts on calls from other hosts and starts (depending on the port) the appropriate service. You need it if you want to log in via **telnet** or **rlogin**.
- **SENDMAIL_TYPE=uucp-only**
  Sets how email is sent. The example is configured for a host which receives (and sends) email only via UUCP (section 17.4). Possible entries are: `uucp-only`, `smtp` and `no`. The latter is if you don't want to receive or send email at all.
- **SENDMAIL_SMARTHOST=uucp-dom:Galois**
  Entry for a host which transports all of his mail via UUCP. All mail is send to *smarthost*; the local system relies on the *smarthost* to deliver its mail.
- **SENDMAIL_LOCALHOST=localhost**
  Here you can enter a couple of hosts which are considered local hosts.

electronic mailto one of these hosts is delivered locally. You can enter up to three hosts here.

- `SENDMAIL_RELAY=helios`
  If this variable is set all incoming and outgoing mail is delivered via this host. No email is delivered locally. This might come in handy if certain hosts only occasionally run under Linux.
- `SMTP=no`
  Sets if a sendmail daemon should be activated. If you receive your email exclusive via UUCP you don't need it, provided you call **sendmail -q** after each polling; (**rmail** started by UUCP just puts mail into a queue but doesn't deliver it). If mail-spool-directories are mounted via NFS (e. g.a network) and the single host has got only outgoing mails this could be set to *no* as well; same applies with use of *relay hosts*.
- `FROM_HEADER=cosmos.com`
  This address appears as sender in your electronic mail. If this is not set the fully qualified hostname is used instead which is not desirable in a net using a central mail-server.
- `START_KERNELD=yes`
  This variable sets whether kerneldaemon should be started automatically at boot time. This daemon is responsible for automatic loading kernel modules on demand. A short description of the module concept and functions of **kerneld** are to be found in chapter section 13.2.
- `START_PORTMAP=no`
  Assigns whether to start portmapper or not. You need portmapper if you plan to use your host as NFS-server (see section 16.6); without this daemon **rpc.mountd** and **rpc.nfsd** can't run! It also is necessary for NIS (see section 16.6).
- `NFS_SERVER=no`
  If the host is going to be used as NFS-server this variable has to be set to `yes`. This initializes the start of **rpc.nfsd** and **rpc.mountd**. A forthgoing description of an NFSserver (e. g.definition of directories to be exported) is described in chapter section 16.6.
- `REEXPORT_NFS=no`
  `yes`, if imported NFS should be re-exported.
- `START_ROUTED=no`
  Whether to start the route daemon.
- `START_NAMED=no`
  Whether to start the name daemon.
- `CREATE_YP_CONF=yes`
  Sets up if **SuSEconfig** should create the necessary files for YP; this depends on the next two entries as well (look at section 16.5). Furthermore `/etc/passwd` and `/etc/group` will be adjusted as well.
- `YP_DOMAINNAME=cosmos.com`
  YP-domainname of hosts. For detailed information please refer to chapter section 16.5.
- `YP_SERVER=helios.cosmos.com`
  Name of the NIS server.

- **`START_GPM=yes`**
  Sets whether mouse console support is to be started; this enables you to exchange text between consoles using the mouse. **Gpm** can cause problems in connection with certain bus mice; if you encounter problems while starting X you should disable **gpm**. The other alternative is to start **xdm** since **gpm** is not started in runlevel 3.
- **`GPM_PARAM=" -t logi -m /dev/mouse"`**
  Initialization parameters for **gpm**; those are normally set via YaST.
- **`RUN_UPDATEDB=yes`**
  Decides whether database for **locate** should be updated once a night; **locate** serves for quick finding of files.
- **`CREATE_INFO_DIR=yes`**
  Decides whether to create `/usr/info/dir` which serves as a general index of all indexpages. This is useful after installing a package which contains infopages. Please keep in mind that **perl** needs to be installed for this to work.
- **`CHECK_PERMISSIONS=set`**
  ...
- **`PERMISSION_SECURITY="easy local"`**
  ...
- **`RPMDB_BACKUP_DIR=/var/adm/backup/rpmdb`**
  ...
- **`MAX_RPMDB_BACKUPS=5`** Maximum number of the backup files of the RPM database.
- **`DELETE_OLD_CORE=yes`**
  Corefiles are memory images of programs which have been killed due to a segmentation fault; those images are very useful for debugging. If this is enabled a regular search and deletion of old corefiles is launched.
- **`MAX_DAYS_FOR_CORE=7`**
  Assigns how old corefiles should be before they are deleted.
- **`MAX_DAYS_FOR_LOG_FILES=365`**
  If a log file (mainly under `/var/log`) reaches a distinct size it will be automatically compressed and archived; `'root'` is informed via electronic mail. This parameter sets how long those files are to be kept on disk before being deleted automatically. If you set this value to 0 no compressing and no archiving will be done; files will be written eternally and can reach a remarkable size! Compressed logfiles can be viewed anytime using **zless**.
- **`MAX_DAYS_IN_TMP=30`**
  /tmp-directories are checked daily whether they have been touched during the last days (this variable is set in days). If this is true they will be deleted automatically.
- **`TMP_DIRS_TO_CLEAR="/tmp /var/tmp"`**
  Enter all directories which are to be searched for old files (as given above) here.
- **`OWNER_TO_KEEP_IN_TMP="root bs"`**
  Files of system users given here should not be deleted even if they have not been touched within the given time.

- `START_LPD=yes`
  ...
- `START_NNTPD=yes`
  ...
- `START_ATD=yes`
  ...
- `START_HTTPD=yes`
  ...
- `START_SQUID=no`
  ...
- `START_MARSNWE=no`
  ...
- `SUSEWM_UPDATE`=yes
  ...
- `CRON=yes`
  Sets the start and stop of *cron daemon*. This daemon lets you start certain programs at a given time. This daemon is only started in runlevel 2 or 3 respectively! It is highly recommended to activate this daemon mainly if your computer runs all time.

> There are a lot of options which require you to regularly run certain commands and programs therefore crondaemon should be activated on every system

Chapter 13
The kernel

The kernel which is written to disk after installation (and which is found in a correctly installed system under `/`) is configured to support a wide range of hardware. Therefore this certainly is not the best choice since it's rather tedious to wait until all (unnecessary) drivers are installed. Moreover some kB are wasted since lots of unnecessary drivers are loaded.

This leads to the great advantage of building an kernel of your own. Furthermore creating an own kernel in some cases provides access to exotic hardware which is not supported by our standard-kernels (e. g.rather exotic bus-mice or sound devices). What is more, kernel configuring provides a view into the inside of the actual status quo of kernel development.

For compiling a kernel there already are a couple of `Makefiles` for the C compiler. These `Makefiles` almost automatically generate everything for you. The only thing you have to do by hand is to give in the list of your hardware.

13.1 Kernel sources

For building a kernel obviously the kernel sources and the C compiler have to be installed. Both of them are located in series D (series `D` (Development)). Generally it is highly recommended to install the C compiler since **C** language is inseparable from Unix operating systems.

Kernel sources are to be found under `/usr/src/linux`. If you plan to experiment with different kernel sources it is obvious that you can unpack them in different directories and access the actual sources via a symbolic link. That's what YaST does automatically.

For clarity: Kernel sources can be put and compiled (theoretically) in any directory. Since there is a lot of software that relies on the sources being under `/usr/src/linux` you should keep this directory providing an error free compilation of system programs.

13.2 Kernel modules

Many drivers and features do not have to be compiled directly into the kernel anymore but can be loaded at runtime via kernel modules. Which drivers are to be compiled into the kernel and which are loaded as runtime modules is defined in kernel configuration.

Kernel modules are located at /lib/modules/<version> where <version> is the actual kernel version.

13.2.1 Handling of modules

The following commands are at your disposition:

- **insmod insmod** loads the given module. The module is searched in a subdirectory of /lib/modules/<Version>.
- **rmmod** Unloads the corresponding module. This obviously is only possible if this module is not needed any more. It is not possible to unload **isofs** module while a CD is still mounted.
- **depmod** This command creates in /lib/modules/<version> the file modules.dep where dependencies of all of the modules are defined. This is necessary to ensure that all dependent modules are loaded along with the selected ones. If kernel daemon is set in /etc/rc.config this file is created each time the system is started.
- **modprobe**
  Loading or unloading of a given module while considering dependencies of this module. This command is extremely powerful and can be used for a lot of things (e. g. probing of all modules of a given type until one is successfully loaded). Contrary to **insmod modprobe** checks /etc/conf.modules and should be used for loading modules. For detailed information on this topic please refer to the corresponding manual page.
- **lsmod**
  Shows you which modules are currently loaded and by how many other modules they are being used. Modules started by kernel-daemon are marked by (autoclean) which shows you that those modules will be removed automatically if they reach their idle-time.

Loading of modules is further influenced by /etc/conf.modules. Here you can add parameters to those modules that directly access your hardware (e. g.CD-ROM drivers..). Those parameters given here are mainly identical to those given at the LILO-prompt (see chapter 7), but they differ in quite some cases. If loading of a module failed you can try to specify your hardware in this file.

13.2.2 The kernel daemon

The most elegant way to use modules is the kernel-daemon anyway. This process runs in the background and takes care of necessary modules to be loaded as soon as this is requested by the kernel. Moreover, unused modules are unloaded automatically after a certain idle time (default: 1 minute).

For using kernel-daemon you have to set the corresponding variable in /etc/rc.config (see section 12.6).

> Those drivers that are needed for accessing the root-filesystem have to be compiled directly into the kernel! So you should not load your SCSI-driver or your filesystem (normally: **ext2**) via modules!

13.3 Kernel configuration

Configuring your kernel can be done in three different ways:

1. On command line
2. In a menu in textmode
3. In a menu under the X Window System

Here we give a short overview on these three possibilities.

13.3.1 Configuring on the command line

For configuring your kernel just change to `/usr/src/linux` and enter:

```
root@earth:/usr/src/linux # make config
```

Now you are asked a couple of options which should be supported by your freshly to be created kernel. Here there are two or three possible answers. Either [y] or [n] or [y] [n] or [m]. 'm' means that this device is not compiled directly into the kernel but as a module instead. It is obvious that any driver that is needed to bring up the system properly should not be loaded as a module. If you press any other key you get a short help text on this subject.

13.3.2 Configuring in text mode

There is a much more comfortable way by typing

```
root@earth:/usr/src/linux # make menuconfig
```

Moreover if you see that you have mistyped something by accident you do not have to fiddle through all the questions.

13.3.3 Configuring under the X Window System

If you have installed the X-Window System (package `xf86`) Tcl/Tk (package `tcl` and package `tk`) you can use

```
root@earth:/usr/src/linux # make xconfig
```

as an alternative. You will be presented an GUI (Graphical User Interface) which makes kernel configuration very comfortable. You should have started the X-Window System as 'root' or you have to take additional steps into consideration (e. g.gathering the display from another user).

newest documentation

All options are roughly previewed in the following chapter. Please keep in mind that several options might differ from version to version. The brand-new documentation is always to be found under `/usr/src/linux/Documentation`.

13.4 Settings in kernel configuration

13.4.1 SMP

In general a kernel for a one-processor machine is built. Compiling an SMP version is not included in standard configuration, so, if you plan to compile an SMP version you have to edit the top-level Makefile by hand. Just remove the '#' from line 14 (**SMP** now has the value 1) and you are done.

13.4.2 Experimental drivers

☞ **Experimental drivers**

Config Question: `Prompt for development code/drivers`

Config Variable: `CONFIG_EXPERIMENTAL`

Allows you to choose from brand new drivers which are not fully tested yet. If you answer `'No'`, a couple of drivers will not be present.

13.4.3 Modul support

☞ **Support of loadable modules**

Config Question: `Enable loadable module support`

Config Variable: `CONFIG_MODULES`

Recommended

Enables use of loadable modules. Those are drivers that are not compiled directly into the kernel but are loaded via **`insmod`** at runtime. If you enable this option there will be an additional select option `'m'` (module); otherwise you will only be presented `'y'` and `'n'`. More information is located under `/usr/src/linux/Documentation/modules.txt`

☞ **Version information on all symbols**

Config Question: `Version information on all symbols`

Config Variable: `CONFIG_MODVERSIONS`

Your modules are more or less independent from the kernel version if you enable this feature (normally any modules have to be recompiled if you install a newer kernel). For novices it is not recommended to use this feature. Kernel and modules have to be compiled the same! If you install an **`MODVERSIONS`** kernel you have to install the modules exactly the same!

☞ **Kernel daemon support**

Config Question: `Kernel daemon support`

Config Variable: `CONFIG_KERNELD`

Recommended

Enables automatic loading of modules at runtime. It is recommended to only use this feature for drivers that are not urgently needed (e. g. additional filesystems or similar).

13.4.4 General configuration

☞ **Kernel math emulation**

Config Question: `Kernel math emulation`

Config Variable: `CONFIG_MATH_EMULATION`

Only set this option if you plan to install this kernels on machine without FPU (e. g. 386, 486SX).

> If you omit this option the kernel will not run on a machine without FPU! An emulation on a machine with a FPU does not hurt but uses up memory.

☞ **Networking support**

Config Question: `Networking support`

Config Variable: `CONFIG_NET`

This is obligatory if you plan to install a network (normal **`TCP/IP`** or **`loopback`**). Disabling the network support should only be done if there is little memory available. Drivers for certain networking devices are specified later on. **Recommended!**

☞ **Limit memory to low 16 MB**

Config Question: `Limit memory to low 16 MB`

Config Variable: `CONFIG_MAX_16M`

Activating this option makes only use of the first 16 MB of RAM. This might be necessary on machines that can not handle caching of memory beyond 16 MB. In most cases this 'loss' of memory leads to a higher performance than loss of cache.

☞ **PCI bios support**

Config Question: `PCI bios support`

Config Variable: `CONFIG_PCI`

Enables 32 bit calls on PCI boards. Some older boards have bugs in their BIOS and can cause trouble. Consider `PCI-HOWTO` for more help. There should not be any problem on newer machines.

☞ **PCI bridge optimisation (experimental)**

Config Question: `PCI bridge optimisation (experimental)`

Config Variable: `CONFIG_PCI_OPTIMIZE`

If you have a PCI motherboard enabling this option can enhance access to some devices.

☞ **System V Inter Process Communication**

Config Question: `System V IPC`

Config Variable: `CONFIG_SYSVIPC`

Inter Process Communication are services such as *messages*, *shared memory* and *semaphores* which are provided by the kernel and which are used by many programs (e. g.the DOS-emulator). **Recommended!**

☞ **Kernel support for a.out binaries**

Config Question: `Kernel support for a.out binaries`

Config Variable: `CONFIG_BINFMT_AOUT`

a.out is the old binary format under Linux. If you still have some of those old programs built with this binary format you have to activate this option. Since there still are (mainly commercial products) that use this format it is recommended to load it as a module.

☞ **Kernel support for ELF binaries**

Config Question: `Kernel support for ELF binaries`

Config Variable: `CONFIG_BINFMT_ELF`

Necessary! **ELF** is Linux' current binary format for libraries and programs. Since almost any software has been ported to ELF you have to activate this option!

> If you do not select this file or select it as a module you will not be able to boot your system since all relevant files use ELF!

☞ **Kernel support for JAVA binaries**

Config Question: `Kernel support for JAVA binaries`

Config Variable: `CONFIG_BINFMT_JAVA`

Enables direct starting of **JAVA** applets by the kernel itself. For benefit of this option the **Java Developers Kit** (JDK) has to be installed on your system. For gathering more information please refer to `/usr/src/linux/Documentation/java.txt`. Loaded as a module it will not hurt anyway.

☞ **Support of different binary formats**

Config Question: `Kernel support for MISC binaries`

Config Variable: `CONFIG_BINFMT_MISC`

Enables directly launching different binary formats by the kernel. It is mainly used for DOSEMU but may also replace the JAVA-loader. Please be aware that this is an experimental feature that is not included in the official kernel version. For using it you have to take the kernel extended by S.u.S.E.. A complete description of this feature can be found under `/usr/src/linux/Documentation/binfmt_misc.txt`.

☞ **Compile kernel as ELF**

Config Question: `Compile kernel as ELF`

Config Variable: `CONFIG_KERNEL_ELF`

Assigns the kernel to use the ELF format. You should answer this question in the affirmative.

☞ **Processor type**

Config Question: `Processor type`

Config Variable: `386, 486, Pentium, PPro`

Enables you to optimize your code according to the processor you use. If the kernel should run on different processors you have to optimize it for 386.

If you set your processor to `Pentium` really aggressive optimization is done (a real fast memcopy routine is used which really exploits the processor). There are hints that some processors can cause problems with these routines (**Cyrix**, **Intel** 100 MHz B-Step). If your system does not run reliably you should not optimize your kernel for Pentium for excluding this cause.

A 486 or Pentium kernel will *not* run on an 386!

13.4.5 Disks, (E)IDE and other block devices

☞ **Normal floppy disk support**

Config Question: `Normal floppy disk support`
Config Variable: `CONFIG_BLK_DEV_FD`

Necessary for using floppy drives under Linux. Disabling this option for security reasons might be useful.

☞ **Enhanced IDE/MFM/RLL support**

Config Question: `Enhanced IDE/MFM/RLL support`
Config Variable: `CONFIG_BLK_DEV_IDE`

Is needed if you have non-SCSI devices installed (harddisk, ATAPI CD-ROM etc.). If your machine does not contain an EIDE disk, or is driven via network, or without any hard disks you can switch off this option.

If you enable this option you may define configuration for these drivers later on.

Some drivers themselves require additional arguments for fully exploiting the drivers capabilities (section 13.4.5, page 188).

☞ **Use old disk-only driver for primary i/f**

Config Question: `Use old disk-only driver for primary i/f`
Config Variable: `CONFIG_BLK_DEV_HD`

Only IDE hard disk support without CD-ROM support. In this special case no enhanced IDE disks are supported. Normally this option is obsolete.

☞ **Include IDE/ATAPI CDROM support**

Config Question: `Include IDE/ATAPI CDROM support`
Config Variable: `CONFIG_BLK_DEV_IDECD`

Necessary for using CD-ROM devices connected to an EIDE controller (ATAPI CD-ROMs)

☞ **Support for ATAPI streamer**

Config Question: `Include IDE/ATAPI TAPE support`
Config Variable: `CONFIG_BLK_DEV_IDETAPE`

Support for ATAPI streamers.

☞ **Partial support for PCMCIA**

Config Question: `Support removeable IDE interfaces`
Config Variable: `CONFIG_BLK_DEV_IDE_PCMCIA`

Support for devices connected to a PCMCIA port. PCMCIA is not part of the official kernel but has to be compiled separately. Is is required to install the sources of the PCMCIA subsystem (package `pcmcia_s`). Please have a look at chapter 6.

☞ **Support for CMD640 EIDE chipsets**

Config Question: `CMD640 chipset bugfix/support`
Config Variable: `CONFIG_BLK_DEV_CMD640`

This chipset used on many EIDE controllers is buggy. If you call one your own you have to activate this option for an error free system.

☞ **Optimization for CMD640 EIDE Chipset**

Config Question: `CMD640 enhanced support`
Config Variable: `CONFIG_BLK_DEV_CMD640_ENHANCED`
Enables auto tuning of some PIO-modes of this special chipset.

☞ **Support for RZ1000 EIDE chipset**

Config Question: `RZ1000 chipset bugfix/support`
Config Variable: `CONFIG_BLK_DEV_RZ1000`

This chipset used on many EIDE controllers is buggy. If you call one your own you have to activate this option for an error free system.

☞ **Support for Mainboard chipset Intel 82371 PIIX**

Config Question: `Intel 82371 PIIX (Triton I/II) chipset`
Config Variable: `CONFIG_BLK_DEV_TRITON`

Enables DMA capabilities if you have one installed. This reduces CPU accesses on your system.

☞ **Other IDE chipset support**

Config Question: `Other IDE chipset support`
Config Variable: `CONFIG_IDE_CHIPSETS`

Support for lots of other chipsets can be set here. Some or them might not even let you handle your secondary controller without this option (most of them have a CD-ROM connected to this controller). Most of these drivers have to be activated by a command line parameter as well (section 7.3.2)!

☞ **ALI M14xx support**

Config Question: `ALI M14xx support`
Config Variable: `CONFIG_BLK_DEV_ALI14XX`
This driver is activated by the kernel parameter **`ide0=ali14xx`**. This enables support for the secondary EIDE controller as well as it increases I/O speed.

☞ **DTC-2278 support**

Config Question: `DTC-2278 support`
Config Variable: `CONFIG_BLK_DEV_DTC2278`
This driver is activated by the kernel parameter **`ide0=ht6560b`** This enables support for the secondary EIDE controller as well as it increases I/O speed. This driver is famous for creating timeouts and retries at an high I/O load. CD-ROMs and tapes are not supported yet.

- ☞ **QDI QD6580 support**

  Config Question: `QDI QD6580 support`
  Config Variable: `CONFIG_BLK_DEV_QD6580`
  This driver is activated by the kernel parameter **`ide0=qd6580`**. This enhances I/O speed.

- ☞ **UMC 8672 support**

  Config Question: `UMC 8672 support`
  Config Variable: `CONFIG_BLK_DEV_UMC8672`
  This driver is activated by the kernel parameter **`ide0=umc8672`**. This enhances I/O speed and supports the secondary EIDE controller.

☞ **Loopback device support**

Config Question: `Loopback device support`

Config Variable: `CONFIG_BLK_DEV_LOOP`

Do not confuse this wit networks loopback mode. Loopback mode allows you to treat a file the same way as a filesystem (e. g.you can mount it). Selfmade CD images or disk images can be tested that way. Moreover encryption is possible using this feature.

☞ **Multiple devices driver support**

Config Question: `Multiple devices driver support`

Config Variable: `CONFIG_BLK_DEV_MD`

Enables you to treat many physical devices as one logical device. This enables software linear and RAID-0. Only for experts.

- ☞ **Linear (append) mode**

  Config Question: `Linear (append) mode`
  Config Variable: `CONFIG_MD_LINEAR`
  Here partitions are bundled together logically by the multiple device driver thus leading to a large mass storage media.

- ☞ **Software Raid-0 (Stripping)**

  Config Question: `RAID-0 (stripping) mode`
  Config Variable: `CONFIG_MD_STRIPPED`
  This enables software RAID-0. Enhances performance if the bundled partitions reside on different disks.

☞ **Providing RAMDISK devices**

Config Question: `RAM disk support`

Config Variable: `CONFIG_BLK_DEV_RAM`

Allows use of a RAM-disk which can be accessed via `/dev/ram`. It is not necessary to assign the size in advance (e. g.at booting). Memory is not allocated and released dynamically. If you plan to use a RAM disk please have a look at `/usr/src/linux/Documentation/ramdisk.txt`.

☞ **Initial RAM disk (initrd) support**

Config Question: `Initial RAM disk (initrd) support`
Config Variable: `CONFIG_BLK_DEV_INITRD`
A special kind of RAM-disk which is dealt with by the loader. This feature is not necessary for every days work. It is used (e. g.) in our installation system for loading the root image from CD.

☞ **Support for old XT hard disks**

Config Question: `XT harddisk support`

Config Variable: `CONFIG_BLK_DEV_XD`

This driver only is needed if you plan to use an XT-controller (8 bit). Normally this is not the case.

13.4.6 Network options

☞ **Network firewalls**

Config Question: `Network firewalls`

Config Variable: `CONFIG_FIREWALL`

Enables firewalling in kernel for letting you run your Linux machine as a firewall. A *firewall* serves for protecting your machine against illegal attacks from the Internet or another network by supervising the network traffic. If you plan to use your machine as a firewall you should enable `CONFIG_IP_FORWARD` too.

☞ **Network aliasing**

Config Question: `Network aliasing`

Config Variable: `CONFIG_NET_ALIAS`

Supports multiple IP addresses on one physical network device. This normally is for services that behave differently on different addresses (e. g.the **Apache** server).

☞ **Support of TCP/IP protocol**

Config Question: `TCP/IP networking`

Config Variable: `CONFIG_INET`

Very much recommended!

TCP/IP are the most important network protocols in Unix/Internet. Since many services use these protocols you should enable this feature even if there is no networking device installed in your machine. At least a so called *loopback device* is provided which enables network access to your own machine. Without this option printing is impossible!

☞ **IP forwarding/gatewaying**

Config Question: `IP forwarding/gatewaying`

Config Variable: `CONFIG_IP_FORWARD`

This option is only needed if the machine should be installed as a router in a local network, or should be connected to the Internet or should serve as a SLIP server.

☞ **IP multicasting**

Config Question: `IP multicasting`

Config Variable: `CONFIG_IP_MULTICAST`

Enables simultaneously addressing multiple networked machines (Fan out). **gated** needs this to be activated.

☞ **IP accounting**

Config Question: `IP accounting`

Config Variable: `CONFIG_IP_ACCT`

Serves for logging of IP packages and is not needed normally. Some programs on the other hand (e. g.**xosview**) can not show network traffic without this option being set.

☞ **PC/TCP compatibility mode**

Config Question: `PC/TCP compatibility mode`

Config Variable: `CONFIG_INET_PCTCP`

If you encounter problems while addressing Linux via PC/TCP software you should answer `'y'`.

☞ **Reverse ARP**

Config Question: `Reverse ARP`

Config Variable: `CONFIG_INET_RARP`

Machines that must boot without network do know their ethernet hardware address at booting but not their IP address. For gathering their address they send a *Reverse Address Resolution Protocol* over the net. If your Linux machine should answer such requests you have to enable this option.

☞ **Disable Path MTU Discovery**

Config Question: `Disable Path MTU Discovery`

Config Variable: `CONFIG_NO_PATH_MTU_DISCOVERY`

Normally the MTU (maximum transfer unit) is assigned automatically. This starts with huge packages going down to smaller ones if necessary. If you encounter problems using this feature you should disable it.

☞ **Drop source routed frames**

Config Question: `Drop source routed frames`

Config Variable: `CONFIG_IP_NOSR`

Packages whose routing is already assigned by the sender should be ignored. For security reasons you should answer yes unless you are pretty sure of what you are doing.

☞ **Allow large windows**

Config Question: `Allow large windows`

Config Variable: `CONFIG_SKB_LARGE`

Enables a higher network performance by use of big sized buffers taken from RAM. If you have less than 16 MB installed you should omit this option.

☞ **The IPX protocol**

Config Question: `The IPX protocol`

Config Variable: `CONFIG_IPX`

Support of the IPX protocol of **Novell**. It enables you to access an **Novell Netware** server from either within **ncpfs** or from the DOS-emulator.

☞ **Appletalk DDP**

Config Question: `Appletalk DDP`

Config Variable: `CONFIG_ATALK`

Enables integration of a Linux machine into an **Appletalk** network.

☞ **Amateur Radio AX.25 Level 2**

Config Question: `Amateur Radio AX.25 Level 2`

Config Variable: `CONFIG_AX25`

Enables data transfer via CB. You will need additional software which can be found on our CD under `/unsorted/Amateur-Funk` (see also `/usr/src/linux/Documentation/networking/ax25.txt`).

☞ **Bridging**

Config Question: `Bridging`

Config Variable: `CONFIG_BRIDGE`

Enables you to use your Linux machine as an ethernet bridge. This means that the segments the machine is connected to seem like one big network to the other users. Bridging code is still under test and should only be used by experienced users.

☞ **Kernel/User network link driver**

Config Question: `Kernel/User network link driver`

Config Variable: `ONFIG_NETLINK`

This driver enables communication between the kernel and user processes in both directions.

13.4.7 SCSI

☞ **SCSI Support**

Config Question: `SCSI support`

Config Variable: `CONFIG_SCSI`

If you have SCSI devices installed activate this option.

☞ **Scsi disk support**

Config Question: `Scsi disk support`

Config Variable: `CONFIG_BLK_DEV_SD`

Activates the SCSI hard disk driver. Do not select this option as a module if you have to boot from a SCSI device! Please be aware that some SCSI controllers can cause problems when loaded as modules!

☞ **Support of SCSI tapes**

Config Question: `Scsi tape support`

Config Variable: `CONFIG_CHR_DEV_ST`

Activates SCSI streamer driver. The tape can be addressed via the device `/dev/st0`.

☞ **Scsi CDROM support**

Config Question: `Scsi CDROM support`

Config Variable: `CONFIG_BLK_DEV_SR`

Activates SCSI-CD-ROM driver.

☞ **Scsi generic support**

Config Question: `Scsi generic support`

Config Variable: `CONFIG_CHR_DEV_SG`

Enables sending generic SCSI commands to the appropriate devices. This enables you to use special SCSI devices as scanners, synthesizers or SCSI CD-ROMs, provided you have the proper software. A direct kernel support for such devices is not implemented yet.

☞ **Probe all LUNs on each SCSI device**

Config Question: `Probe all LUNs on each SCSI device`

Config Variable: `CONFIG_SCSI_MULTI_LUN`

Some SCSI devices support LUNs (logical unit number) like some CD juke-boxes . Activate this kernel support if you plan to use one of these.

☞ **Verbose SCSI error reporting**

Config Question: `Verbose SCSI error reporting`

Config Variable: `CONFIG_SCSI_CONSTANTS`

Extensive SCSI error reports instead of cryptical hex-codes need app. 12 KB of kernel size. Only necessary if you have problems and want to research.

In this section you can select drivers for different SCSI hostadapters.

13.4.8 Networking cards

☞ **Network device support?**

Config Question: `Network device support?`

Config Variable: `CONFIG_NETDEVICES`

If there is a networking card installed or you plan to use SLIP (Serial Line Internet Protocol) you should answer yes.

☞ **Dummy net driver support**

Config Question: `Dummy net driver support`

Config Variable: `CONFIG_DUMMY`

The dummy driver comes in handy for simulating network connections as well as for configuration and use of SLIP and PPP.

☞ **EQL (serial line load balancing)**

Config Question: `EQL (serial line load balancing)`

Config Variable: `CONFIG_EQUALIZER`

If you call two serial lines your own (e. g.two modems) and use SLIP or PPP you can double your speed by separating the stream on both lines simultaneously. Please also have a look at the README file in `/usr/src/linux/drivers/net/README.eql`.

☞ **FRAD (Frame Relay Access Device) support**

Config Question: `FRAD (Frame Relay Access Device) support`

Config Variable: `CONFIG_DLCI`

Frame Relay is a fast and cheap way for connecting to one's Internet provider or for maintaining a private wide area network. Via a physical connection multiple logical connections are transmitted. You need special hardware to run Frame Relay. It is recommended to look at `/usr/src/linux/Documentation/framerelay.txt`.

☞ **Max open DLCI**

Config Question: `Max open DLCI`

Config Variable: `CONFIG_DLCI_COUNT`

Maximum number of point-to-point connections the driver can maintain.

☞ **Max DLCI per device**

Config Question: `Max DLCI per device`

Config Variable: `CONFIG_DLCI_MAX`

Maximum number of point-to-point connections per device.

☞ **Sangoma S502A FRAD support**

Config Question: `Sangoma S502A FRAD support`

Config Variable: `CONFIG_SDLA`

Support for Sangoma S502A, S502E and S508 Frame Relay Access Devices.

☞ **PLIP (parallel port) support**

Config Question: `PLIP (parallel port) support`

Config Variable: `CONFIG_PLIP`

PLIP (Parallel Line Internet Protocol) enables use of the Internet protocol via the parallel interface. A special cable is needed for this. `/usr/src/linux/drivers/net/plip.c` describes the pins of this cable. Please keep in mind that you are not able to connect a printer if the kernel is compiled using this option! In this case you should not activate `CONFIG_PRINTER` (section 13.4.12, page 202).

☞ **PPP (point-to-point) support**

Config Question: `PPP (point-to-point) support`

Config Variable: `CONFIG_PPP`

PPP is a more modern protocol than SLIP but in fact contains the same features. Please have a look at `PPP-HOWTO` and the documentation under `/usr/src/linux/Documentation/networking/ppp.txt`. If you have set `CONFIG_MODVERSIONS` you cannot add the PPP device directly into the kernel but have to compile it as a module.

☞ **SLIP (serial line) support**

Config Question: `SLIP (serial line) support`

Config Variable: `CONFIG_SLIP`

SLIP (Serial Line Internet Protocol) enables the use of the internet protocol via a serial line (e. g. a modem). Also if you plan to use the machine as a SLIP server you need to activate this option.

☞ **CSLIP compressed headers**

Config Question: `CSLIP compressed headers`

Config Variable: `CONFIG_SLIP_COMPRESSED`

Use of compressed headers enhances SLIP's performance but is only possible if used and configured on both sides.

☞ **Keepalive and linefill**

Config Question: `Keepalive and linefill`

Config Variable: `CONFIG_SLIP_SMART`

Enhancements mainly for using SLIP on interrupted analog lines.

Now you can select drivers for different networking cards. The kernel even contains Ethernet pocket adapter support. Please consult `Ethernet-HOWTO`.

13.4.9 ISDN subsystem

☞ **ISDN support**

Config Question: `ISDN support`

Config Variable: `CONFIG_ISDN`

ISDN4linux (package `i4l`) has become integrated into the standard kernel in the meantime. We recommend to use the modularized version.

☞ **Support synchronous PPP**

Config Question: `Support synchronous PPP`

Config Variable: `CONFIG_ISDN_PPP`

Enables synchronous PPP via ISDN used e. g.by Cisco and Sun. A complete documentation can be found under
`/usr/src/linux/Documentation/isdn/README.syncppp`.

☞ **Support audio via ISDN**

Config Question: `Support audio via ISDN`

Config Variable: `CONFIG_ISDN_AUDIO`

Enables you to make telephone calls over the internet. The only low-level driver that supports this feature is Teles. Moreover this function is necessary if you want to use the answering machine **vbox**.

☞ **ICN 2B and 4B support**

Config Question: `ICN 2B and 4B support`

Config Variable: `CONFIG_ISDN_DRV_ICN`

Support for **ICN**'s active ISDN cards. Documentation concerning this topic may be found at `/usr/src/linux/Documentation/isdn/` in `README` and `README.icn`.

☞ **PCBIT-D support**

Config Question: `PCBIT-D support`

Config Variable: `CONFIG_ISDN_DRV_PCBIT`

Support of **Octal**'s ISDN cards. For using this card you need additional firmware which has to be loaded into your card. More information can be found under `/usr/src/linux/Documentation/isdn/` in `README` and `README.pcbit`.

☞ **(old)Teles/NICCY1016PC/Creatix support**

Config Question: `Teles/NICCY1016PC/Creatix support`

Config Variable: `CONFIG_ISDN_DRV_TELES`

Support of **Teles S0-16.0**, **Teles S0-16.3**, **Teles S0-8**, and compatible ISDN cards. Please look at `/usr/src/linux/Documentation/isdn/README`. You should use the newer **HiSax** anyway.

13.4.10 Proprietary CD-ROM drives

☞ **Aztech/Orchid/Okano/Wearnes CDROM**

Config Question: `Aztech/Orchid/Okano/Wearnes CDROM`

Config Variable: `CONFIG_AZTCD`

Drivers for these devices with an proprietary CD-ROM controller. AT-API drives are specified via `CONFIG_BLK_DEV_IDECD` (section 13.4.5, page 187).

☞ **Goldstar R420 CDROM support**

Config Question: `Goldstar R420 CDROM support`

Config Variable: `CONFIG_GSCD`

Driver for Goldstar CD-ROM drive R 420.

☞ **Matsushita/Panasonic CDROM driver support**

Config Question: `Matsushita/Panasonic CDROM driver support`

Config Variable: `CONFIG_SBPCD`

Driver for all CD-ROM drives (even **Creative Labs**, **Kotobuki**,...) that are connected to a SoundBlaster card. This driver can drive up to 4 devices. Please read `CDROM-HOWTO`.

☞ **MicroSolutions backpack CDROM support**

Config Question: `MicroSolutions backpack CDROM support`

Config Variable: `CONFIG_BPCD`

This is a beta driver for CD-ROMs connected to a parallel port. Since this driver is not implemented in the standard kernel this option only is present if you have installed an appropriately patched kernel. If you use this driver your parallel port cannot be used for anything other that is why it is recommended to load this device as a module.

☞ **Support of proprietary Mitusmi CD-ROMs**

Config Question: `Mitsumi (not IDE/ATAPI) CDROM driver`

Config Variable: `CONFIG_MCD`

Driver for proprietary **Mitsumi** drives with their own controller. This does not apply to ATAPI CD-ROMs e. g.an **FX-400**.

☞ **Mitsumi [XA/MultiSession] CDROM support**

Config Question: `Mitsumi [XA/MultiSession] CDROM support`

Config Variable: `CONFIG_MCDX`

Driver for proprietary **Mitsumi** drives with their own controller which additionally supports XA and multisession (Photo CDs).

☞ **Optics Storage DOLPHIN 8000AT CDROM**

Config Question: `Optics Storage DOLPHIN 8000AT CDROM`

Config Variable: `CONFIG_OPTCD`

Driver for the **Optics Storage** CD-ROM drive **DOLPHIN 8000AT**.

☞ **Philips/LMS CM206 CDROM support**

Config Question: `Philips/LMS CM206 CDROM support`

Config Variable: `CONFIG_CM206`

Driver for **Philips CM206**.

☞ **Sanyo CDR-H94A CDROM support**

Config Question: `Sanyo CDR-H94A CDROM support`

Config Variable: `CONFIG_SJCD`

Driver for **Sanyo CDR-H94A**.

☞ **Soft configurable cdrom interface card**

Config Question: `Soft configurable cdrom interface card`

Config Variable: `CONFIG_CDI_INIT`

Supports software configured drives which have to be initialised at boot time. At the moment there is support for **ISP16/MAD16/Mozart**.

☞ **Sony CDU31A/CDU33A CDROM driver support**

Config Question: `Sony CDU31A/CDU33A CDROM driver support`
Config Variable: `CONFIG_CDU31A`
Driver for Sony CDU31A or. CDU33A. Please be aware that the driver does no longer autoprobe. Now you have to give the necessary parameters at the boot prompt. Therefore you should add these values to your LILO configuration file in **append** line (chapter 7).

☞ **Sony CDU535 CDROM driver support**

Config Question: `Sony CDU535 CDROM driver support`
Config Variable: `CONFIG_CDU535`
Driver for Sony CDU535.

13.4.11 File systems

☞ **Quota support**

Config Question: `Quota support`
Config Variable: `CONFIG_QUOTA`
Enables you to assign disk space to the several users on your system (diskquotas). This only is useful for several users and only works with ext2 file system.

☞ **Mandatory lock support**

Config Question: `Mandatory lock support`
Config Variable: `CONFIG_LOCK_MANDATORY`
Mandatory locking is used by some System V database applications. For using this feature you will need newer NFS daemons as well as new SAMBA,new Nettalk, new Mars-nwe and more. At the moment none of them is available Therefore you should only activate this option if you really need it and if you are absolutely sure of what you are doing.

☞ **Standard (minix) fs support**

Config Question: `Standard (minix) fs support`
Config Variable: `CONFIG_MINIX_FS`
Minix was the first file system available for Linux. Due to its restrictions (maximum size of a file system is 64 MB, filenames only 14 characters long filenames) it is more or less only used for installation disks.

☞ **Extended fs support**

Config Question: `Extended fs support`
Config Variable: `CONFIG_EXT_FS`
Extended file system was the first Linux-specific file system which got rid of Minix' restrictions. It is not used any more and has only nostalgic value.

☞ **Second extended fs support**

Config Question: `Second extended fs support`
Config Variable: `CONFIG_EXT2_FS`
The Second Extended Filesystem in the meantime has become Linux' standard file system.

> This file system *has* to be compiled into the kernel since it is root's standard file system! Use as loadable module therefore is not possible!

☞ **xiafs filesystem support**

Config Question: `xiafs filesystem support`
Config Variable: `CONFIG_XIA_FS`
Xia file system was the "opponent" of the Second Extended File System in Linux' early days. Nowadays it is not important anymore.

☞ **DOS FAT fs support**

Config Question: `DOS FAT fs support`
Config Variable: `CONFIG_FAT_FS`
Basic FAT support for FAT file systems such as MS-DOS and VFAT.

☞ **MSDOS fs support**

Config Question: `MSDOS fs support`
Config Variable: `CONFIG_MSDOS_FS`
If you enable this option you will be able to access your MS–DOS partitions from within Linux. Many installation packages for Linux come with disks in DOS format. If you want to use them you need this option enabled.

☞ **VFAT (Windows-95) fs support**

Config Question: `VFAT (Windows-95) fs support`
Config Variable: `CONFIG_VFAT_FS`
Enables accessing Windows 95 partitions with long filenames. This file system can not be used as a root partition. Take the UMSDOS filesystem instead. You can mount a Windows 95 partition either as DOS partition or as VFAT partition; you will only get the long filenames with the latter.

☞ **umsdos: Unix like fs on top of FAT**

Config Question: `umsdos: Unix like fs on top of FAT`
Config Variable: `CONFIG_UMSDOS_FS`

Demo mode

UMSDOS is an add-on for the MS-DOS file system which enables you to create files on normal DOS partitions with all properties and permissions of Unix files. These information is kept in one special file which will be created in any directory of the used partition and which is kept up to date by the UMSDOS file system. Furthermore this file system enables you to install a Linux demo-mode on an existing DOS partition. Of course, accessing this file system is much slower than the Second Extended File System.

☞ **`/proc` filesystem support**

Config Question: `/proc filesystem support`

Config Variable: `CONFIG_PROC_FS`

Recommended

The process-file-system represents an easy to handle means for receiving process information and kernel addresses. This in fact is a virtual file system, thus it does not occupy any disk space. This file system is needed for gathering information on processes (e. g.**ps**, **free** and more) and is therefore really important and necessary.

☞ **NFS filesystem support**

Config Question: `NFS filesystem support`

Config Variable: `CONFIG_NFS_FS`

Enables transparent mounting of remote partitions via a network in one's own directory tree.

☞ **Root file system on NFS**

Config Question: `Root file system on NFS`

Config Variable: `CONFIG_ROOT_NFS`

Allows you to mount even the root directory via NFA. This is for installing diskless workstations.

☞ **SMB filesystem support**

Config Question: `SMB filesystem support`

Config Variable: `CONFIG_SMB_FS`

If you need to mount disks running under Windows 95 or NT you need this file system. If you want the Linux machine to serve as a SMB server for the appropriate clients this file system is not needed. More information concerning this topic can be found under `/usr/src/linux/Documentation/filesystems/smbfs.txt` and in chapter 18.

☞ **NCP filesystem support**

Config Question: `NCP filesystem support`

Config Variable: `CONFIG_NCP_FS`

Enables you to connect volumes of a Novell NetWare server in your directory tree, thus gaining full access on it. It is required that the kernel supports IPX (see section 13.4.6, page 192). Please have also a look at `/usr/src/linux/Documentation/filesystems/ncpfs.txt`.

☞ **ISO9660 cdrom filesystem support**

Config Question: `ISO9660 cdrom filesystem support`

Config Variable: `CONFIG_ISO9660_FS`

The file system used for CD-ROMs. If this is disabled no access onto CD-ROM drives is possible.

☞ **OS/2 HPFS filesystem (read only)**

Config Question: `OS/2 HPFS filesystem (read only)`

Config Variable: `CONFIG_HPFS_FS`

OS/2's own *High Performance File System*. Since there is no information on this file system and the Linux implementation mainly relies on guesses there is only read access to it.

☞ **System V/Coherent filesystem support**

Config Question: `System V/Coherent filesystem support`

Config Variable: `CONFIG_SYSV_FS`

SCO, **Coherent** and **Xenix** all are commercial Unix systems for PCs and use this file system. If you plan to access one of these disks or floppies you have to activate this option. If you only plan to access these file systems via ☞*NFS* of course you do not need it in your own kernel.

☞ **Amiga FFS filesystem support**

Config Question: `Amiga FFS filesystem support`

Config Variable: `CONFIG_AFFS_FS`

The Fast File System (FFS) is used by Amiga machines since version AmigaOS 1.3 (34.20). If you plan to access one of these partitions you have to enable this option. Unfortunately access of Amiga disks is not possible since their disk controllers are not compatible.

☞ **UFS filesystem support (read only)**

Config Question: `UFS filesystem support (read only)`

Config Variable: `CONFIG_UFS_FS`

BSD, some Unix derives (**SunOS**, **FreeBSD**, **NetBSD**, **NeXTstep**) use this file system. Activating this option enables read only access on these file systems.

13.4.12 Character devices

☞ **Standard/generic serial support**

Config Question: `Standard/generic serial support`

Config Variable: `CONFIG_SERIAL`

Support for serial ports. Is needed if you plan to install a mouse. **Recommended!**

☞ **Digiboard PC/Xx Support**

Config Question: `Digiboard PC/Xx Support`

Config Variable: `CONFIG_DIGI`

Drivers of Digiboard's multi-serial cards.

☞ **Cyclades async mux support**

Config Question: `Cyclades async mux support`

Config Variable: `CONFIG_CYCLADES`

Cyclades multi-serial drivers.

☞ **Stallion multiport serial support**

Config Question: `Stallion multiport serial support`
Config Variable: `CONFIG_STALDRV`
Stallion multi-serial driver.

☞ **Parallel printer support**

Config Question: `Parallel printer support`
Config Variable: `CONFIG_PRINTER`
Driver for connecting a printer to the parallel port. If you want to use PLIP (`CONFIG_PLIP`, section 13.4.8) you should not activate this option. If you want to benefit from both you must either patch the kernel or compile both drivers as modules (see section 13.2) and load them on demand. More information can be found under `Printing-HOWTO`.

☞ **Mouse Support (not serial mice)**

Config Question: `Mouse Support (not serial mice)`
Config Variable: `CONFIG_MOUSE`
If you do not use a serial mouse but a mouse with its own interface, or a PS/2 mouse you have to set this option. Selecting a busmouse is done like this. The interrupt used by the mouse (default is IRQ 5) is set in `/usr/src/linux/include/linux/busmouse.h`. Either you change this value or you parse the appropriate parameter at boot time.

☞ **ATIXL busmouse support**

Config Question: `ATIXL busmouse support`
Config Variable: `CONFIG_ATIXL_BUSMOUSE`
Driver for ATIXL busmice. (on ATI graphics cards)

☞ **Logitech busmouse support**

Config Question: `Logitech busmouse support`
Config Variable: `CONFIG_BUSMOUSE`
Driver for Logitech busmice.

☞ **Microsoft busmouse support**

Config Question: `Microsoft busmouse support`
Config Variable: `CONFIG_MS_BUSMOUSE`
Driver for Microsoft mice.

☞ **PS/2 mouse (auxiliary device) support**

Config Question: `PS/2 mouse (auxiliary device) support`
Config Variable: `CONFIG_PSMOUSE`
Driver for PS/2 mouse. In almost any notebook the integrated mouse is touchpad, trackpoint and PS/2 compatible.

☞ **Support for user misc device modules**

Config Question: `Support for user misc device modules`

Config Variable: `CONFIG_UMISC`

Support for some modules which are not part of the kernel and which should be loaded on demand (e. g.touch screens). If you intend to design such a driver or just want to test you should say yes, all others say no.

☞ **QIC-02 tape support**

Config Question: `QIC-02 tape support`

Config Variable: `CONFIG_QIC02_TAPE`

Driver for streamers which are neither connected to a SCSI nor to an IDE bus (floppy tape).

☞ **Ftape (QIC-80/Travan) support**

Config Question: `Ftape (QIC-80/Travan) support`

Config Variable: `CONFIG_FTAPE`

Driver floppy tape streamer.

☞ **number of ftape buffers**

Config Question: `number of ftape buffers`

Config Variable: `NR_FTAPE_BUFFERS [3])`

Amount of buffers for the FTape driver.

☞ **Advanced Power Management BIOS support**

Config Question: `Advanced Power Management BIOS support`

Config Variable: `CONFIG_APM`

Enables power management with machines that fulfil the standard. This mainly is of interest for notebook owners. This feature does not work with **TI 4000M TravelMate** and **ACER 486/DX4-75** since they do not follow the standard.

☞ **Watchdog Timer Support**

Config Question: `Watchdog Timer Support`

Config Variable: `CONFIG_WATCHDOG`

If you enable this you release a "watchdog". If a certain file is not written for more than one minute (`/dev/watchdog`) the machine is booted automatically. This might come in handy for network servers that want to keep the time-down as short as possible.

☞ **Enhanced Real Time Clock Support**

Config Question: `Enhanced Real Time Clock Support`

Config Variable: `CONFIG_RTC`

Allows access onto the real time hardware clock. Further information can be found under `/usr/src/linux/Documentation/rtc.txt`.

☞ **Sound card support**

Config Question: `Sound card support`

Config Variable: `CONFIG_SOUND`

Driver for different sound cards. They are configured in a separate step. The Linux kernel in the meantime supports almost any well-known sound card. More information is to be found under `drivers/sound`.

☞ **Kernel profiling support**

Config Question: `Kernel profiling support`

Config Variable: `CONFIG_PROFILE`

This is for kernel hackers who want to know what the kernel does and how long it takes.

13.5 Compiling the kernel

After having adopted the kernel to your needs you start compilation by entering:

```
root@earth:/usr/src/linux # make dep
root@earth:/usr/src/linux # make clean
root@earth:/usr/src/linux # make zImage
```

These three commands can be entered in one line as well. The are started one after the other. This might be useful if you want to (e. g.) compile a kernel over night. Just enter:

```
root@earth:/usr/src/linux # make dep clean zImage
```

Dependent on your systems power it now takes app. 4 minutes (fast **PentiumPro**[1] up to several hours (386 with 8 MB) until you get your own kernel. While compiling you can of course examine the system on one of the other consoles. After a successful compilation you will find the kernel under `/usr/src/linux/arch/i386/boot`. The kernel image (the file containing the kernel) is known as `zImage`. If this file does not exist most definitely an error occurred. This might got lost in all the output. Verify it by entering:

```
root@earth:/usr/src/linux # make zImage
```

And watch out for error messages. But do not panic: Errors are very rare! If you have configured some parts as modules you have to start compiling these modules as well. This is done by entering

```
root@earth:/usr/src/linux # make modules
```

If the modules have been compiled successfully you can install them into the appropriate directories (`/lib/modules/`<`Version`>) by entering

```
root@earth:/usr/src/linux # make modules_install
```

13.6 Install kernel

After having compiled a kernel you have to make sure that it is used from now on. If you use LILOyou have to reinstall it. Normally you just copy the new kernel to `/vmlinuz` and invoke LILO

```
root@earth:/ # lilo
```

[1] A very popular test for hard and software s to compile the kernel with the `make -j`. You will need a quite an amount of RAM (more than 100 MB). A compiler is launched for each source file

For reassuring that nothing unexpected happens it is recommended to keep the old kernel just in case of an emergency.

Enter an additional label **/vmlinuz.old** into /etc/lilo.conf as boot image and rename the old kernel to /vmlinuz.old thus making sure that you can boot the old one if the new one happens to fail. This is thoroughly described in chapter 8. If you have adopted /etc/lilo.conf to your needs you can enter

```
root@earth:/usr/src/linux # make zlilo
```

for installing LILO after having compiled the kernel automatically.

If you boot Linux via DOS using linux.bat (loadlin) you have to copy the kernel to /dosc/loadlin/zimage[2] for it to become active at next booting.

Furthermore there is the following to keep in mind: The file /System.map contains kernel symbols which are needed by the modules for launching kernel functions correctly. This file depends on the current kernel. Therefore you should copy /usr/src/linux/System.map[3] to the root directory (/). If you create your kernels using **make zlilo** this task is automatically done for you.

If you get an error message like "System.map does not match actual kernel" then probably System.map has not been copied.

13.7 Creating a boot disk

If you like to create a boot disk with the new kernel you can use the following command:

```
root@earth:/usr/src/linux # make zdisk
```

13.8 Cleaning your disk after compilation

You can delete the object files created during compilation if you do not have enough space on your disk by

```
root@earth: # cd /usr/src/linux
root@earth:/usr/src/linux # make clean
```

If there is plenty of space and you plan to compile a kernel more often you might skip the last step. A new compilation will be faster as only those parts of the system are re-compiled that have changed.

[2] or to the directory where you have installed loadlin respectively
[3] this file is created every time you create a new kernel

Chapter **14**

The X Window System

The **X Window System** is a de facto standard GUI for Unix. **X11** was developed as an enterprise of **DEC** (Digital Equipment Corporation) and the project Athena at **MIT** (Massachusetts Institute of Technology). The first release of **X11R1** was released in september 1987. Since release 6 the **X Consortium, Inc.** is responsible for the development of the **X Window System**.

Up to now the **X Window System** supports a wide variety of machines. **X11** is a network oriented product. Applications running on one machine `earth` can display their results on another machine provided the machines are connected via a network. This might be a local LAN or another machine reachable via Internet.

XFree86™ is a freely available implementation of X servers for PC systems. It was developed by a handful of ambitious programmers who founded the XFree86 team in 1992. This team lead to the foundation of the **The XFree86 Project** in 1994 whose aim it is to continue research and development on **X11** and to provide it to public. We would like to thank the XFree86 team for their help and for the permission to include beta servers on our CDs,[1] because without these realisation of this CD would have been a lot more work (if it would have been possible at all).

The current release, XFree86™ 3.3.2, is a realisation of an X11R6.3 system for PC based Unix systems.

The next chapters cover the configuration of your X server. **XF86Setup** is presented which enables you to do this rather easily. For really exploiting your hardware (graphic adaptor, monitor) there exists an option for optimizing the configuration. Detailed information on configuring the X Window System can be found under `/usr/doc/packages/xf86`.

> Please configure your X Window System with extreme care! Never start the X Window System until the configuration has finished. A faulty configured system can lead to irreparable damages to your hardware (this specially applies for fixed frequencies monitors). The authors of this book and S.u.S.E. cannot be held responsible for damages. This text has been written with extreme care, but this does not guarantee that all methods presented in this book are correct and cannot damage your hardware.

[1] Parts of this documentation are taken from chapter *XFree86konfigurieren* from [Hetze et al., 1995] which was kindly given to us by **Dirk Hohndel**

Besides **xf86config** S.u.S.E. Linux contains another one: **XF86Setup**. On the contrary to xf86config **XF86Setup** directly works together with your server and is mouse driven.

Install the package `xfsetup` with YaST (and all dependencies) as well as the appropriate X server.

14.1 Configuration using XF86Setup

XF86Setup serves for a menu driven easy configuration of the XFree86™ system. Here, on the contrary to **xf86config**, configuration is almost done in graphical mode. Nevertheless **XF86Setup** does not try to replace **xf86config** (see page 210). There are known situations where configuring using **XF86Setup** fails (e. g.if you have a graphics card that is not 100% VGA compatible, or you use a fixed-frequency monitor). In these cases you should switch back to **xf86config**.

For configuration the following data should be held at hand:

- Type of mouse and port where the mouse is connected to.
- Vendor's name of your graphics device, better, the technical specifications.
- Your monitor's data (mainly horizontal and vertical frequencies, etc.)

If these data are known, or at hand you can start configuring. This, of course, can only be done by '`root`'.

The configuration is launched using the command:

```
root@earth:/root # /usr/X11R6/bin/XF86Setup
```

First the program checks whether all required files and packages are installed. If there already exists an `/etc/XF86Config` on your machine you are asked if you would like to use it as default.

Then you should press Enter for entering graphical mode. After having pressed Enter it can take up to two minutes (depending on the load of your machine) until the graphic screen appears and you can see the welcome message of **XF86Setup**.

A short help screen is presented which tells you how to proceed. Now just press Enter ; a help screen appears that tells you the keyboard settings. This page can be viewed anytime by pressing ? or Alt + H .

14.1.1 Mouse

After pressing Enter once again you are on the mouse configuration page. Please do not move the mouse right now but use p for selecting the appropriate protocol instead.

For most modern serial mice **`Microsoft`** or **`LogiTech`** is the right choice. Most bus mice are **`PS/2`** compatible. After having selected a protocol you should press Enter for selecting the interface. First serial interface is **`/dev/ttyS0`**, the second is **`/dev/ttyS1`**. Bus mice are connected to **`/dev/psaux`** (normally). By means of the cursor keys you can scroll the list of interfaces to find the correct one. If you have already entered a port for your mouse at installation you should just enter **/dev/mouse**.

After having set these items, press a (for apply) for writing the settings. Now you should try to move your mouse. The cursor should follow the mouse and a mouse-click should cause the pictogram on the right to blink accordingly. If this does not work you should select another protocol (and eventually another interface) by pressing p and n .

So, if the middle mouse button does not behave as it should you could try to activate **ChordMiddle** (either click it or select it by c). Quite similar you can select e (or click on **Emulate3Buttons**) for emulating the third button by pressing button one and two simultaneously.

14.1.2 Keyboard

If the configuration of you mouse is done you can go to keyboard configuration either by clicking on **Keyboard** or by pressing Alt + k .

14.1.3 Graphics device

Now on it goes with specification of your graphics device. Either you click on **Card** or you press Alt + c .

Now there is a long list of supported graphics cards. You can scroll through this list with the scrollbars on the right side. If you find your card in the list just click on it. Most probably this is all you have to do. You should not blindly accept a card with a similar name since even on cards with similar names there can be slight variations in clock-chips and ramdac[2]!

If your card is not on this list configuration becomes slightly more complicated. Anyway you should select **Detailed Setup** and decide yourself which server and which options to choose.

First you should select the X-server you are going to use.

Here **Mono** stands for the simplest server of the XFree86server family. which should run on any VGA card. On the other hand it does not exploit the card.

VGA16 is a similar generic server which should also work on most graphics cards. This server at least offers 16 colours.

SVGA-server is the server of your choice for most SVGA cards that do not have a special server.

The other servers are only interesting for a small group of cards. **8514** for older 8514 compatible cards, **AGX** for XGA cards and compatible, **I128** for cards based upon Number Nine's I128 chip, **Mach8**, **Mach32** und **Mach64** for the corresponding ATI cards. **P9000** is designed for Weitek P9000 chipset only, they can not be used with P9100 cards. **S3** suits every S3 cards except ViRGE and ViRGE/VX. **S3V** server supports these cards. Finally the **W32** server only supports Tseng ET4000/W32 and ET6000. These cards are supported by **SVGA** as well. Which one to choose mainly depends on your card. Some W32 cards only run with the **W32** server, others only run with **SVGA**.

[2] **R**andom **A**ccess **M**emory **D**igital-to-**A**nalogue **C**onverter

And only the latter supports higher colour depths with ET6000 cards. W32 cards are only supported (by both servers) with 256 colours.

All other settings should only be done by experienced users and only if the default settings fail. In this case you are obliged to find out which chipset and which ramdac is installed on your card (if this is not part of the graphic chipset). Studying the online documentation and the man pages is inevitable.

14.1.4 Monitors

Next you should select your monitor. Click on **`Monitor`** or you press Alt + n .

Now you can enter your monitor's settings. There are many default configurations listed. If none of them suits you you have to enter the correct values in the above entries.

Now configuration is almost done. The options that are sorted under **`Other`** usually do not have to be changed. Now click on **`Done`** or press Alt + d and confirm by pressing **`Okay`**.

XF86Setup now tries to launch the selected server. Normally this should not fail. If you receive an error message or if the system crashes you should restart configuration and try to solve the problems by means of the online documentation as well as the man pages.

If the server started as expected you will be presented a screen where you can finally write your configuration. Normally there is no reason why you should change the path `/etc/XF86Config`, so just click on **`Save the configuration and exit`** and confirm by pressing **`Okay`**.

Finally you are asked whether you want to create a link **X** to the selected server. Please answer yes.

The X Window System now can be launched by any user using the **startx** command[3] All error messages that occur can be found in `~/.X.err`

14.2 Configuration using xf86config

In most cases **XF86Setup** is superior to **xf86config** as a simple configuration tool. There are nevertheless some rare cases where **XF86Setup** fails, in this case just use **xf86config** which works in almost any cases.

Please have the following information at hand.

- Mouse type, port where the mouse is connected to and baud rate (the latter normally is optional).
- Specification of the graphics card
- Monitor data (Frequencies etc.)

If these settings are known, or you have your manuals at hand you can start configuring. This can only be done by `'root'`!

The configuration is launched by:

`root@earth:/root #` `/usr/X11R6/bin/xf86config`

[3] A pre-configured X-Window GUI (using the fvwm windowmanager) is provided along with the `'sample user'`; it is recommended to start **startx** as the `'sample user'` instead of `'root'`

14.2.1 Mouse

After the welcome screen you are asked your mouse type. There are the following selections:

```
1. Microsoft compatible (2-button protocol)
2. Mouse Systems (3-button protocol)
3. Bus Mouse
4. PS/2 Mouse
5. Logitech Mouse (serial, old type, Logitech protocol)
6. Logitech MouseMan (Microsoft compatible)
7. MM Series
8. MM HitTablet
```

Screen output 14.2.1: Selection of mouse for X

While assigning mice you should consider that many of the new Logitech mice are Microsoft compatible or use the MouseMan protocol. The selection **`Bus Mouse`** refers to any bus mice, even Logitech!

The suitable type is selected by giving its number. There might be a request whether “ChorMiddle” should be activated. This is necessary for some Logitech mice or trackballs to activate the second mouse button (the middle one).

```
Please answer the following question with either 'y' or 'n'.

Do you want to enable ChordMiddle?
```

If you use a two-button mouse you can emulate the third button by answering `'y'` to the next question.

```
Please answer the following question with either 'y' or 'n'.

Do you want to enable Emulate3Buttons?
```

The middle button is emulated by simultaneously pressing the two mouse buttons.

Next you have to specify the mouses interface:

```
Now give the full device name that the mouse is connected to, for example

/dev/tty00. Just pressing enter will use the default, /dev/mouse.

Mouse device:
```

If you already entered a port for your mouse at system installation you should enter it (**`/dev/mouse`**)

14.2.2 Keyboard

Now you are asked whether to assign **`Meta`** (ESC) to the left Alt-key and to assign **`ModeShift`** (AltGr) for the right.

```
Please answer the following question with either 'y' or 'n'.

Do you want to enable these bindings for the Alt keys?
```

You should answer `'y'` for being able to access all keys via AltGr and the left Alt can serve as Meta-key[4].

[4] e. g.in Emacs

14.2.3 Monitor

Next you have to specify your monitor. You should be extreme careful with vertical and horizontal frequencies! These can be found in your monitors handbook.

> Setting frequencies wrongly can lead to irreparable damages to your monitor! The X-Window system only addresses video modes which drive the monitor in the given frequency range. Entering frequencies which the monitor was not designed for can severely damage it!

Some monitors are listed under `/usr/X11R6/lib/X11/doc/Monitors`[5].

For entering horizontal frequency the following selection is presented:

```
hsync in kHz; monitor type with characteristic modes
 1 31.5;               Standard VGA, 640x480 @ 60 Hz
 2 31.5 - 35.1;        Super VGA, 800x600 @ 56 Hz
 3 31.5, 35.5;         8514 Compatible, 1024x768 @ 87 Hz interl.
                       (no 800x600)
 4 31.5, 35.15, 35.5;  Super VGA, 1024x768 @ 87 Hz il.,
                       800x600 @ 56 Hz
 5 31.5 - 37.9;        Extended Super VGA, 800x600 @ 60 Hz,
                       640x480 @ 72 Hz
 6 31.5 - 48.5;        Non-Interlaced SVGA, 1024x768 @ 60 Hz,
                       800x600 @ 72 Hz
 7 31.5 - 57.0;        High Frequency SVGA, 1024x768 @ 70 Hz
 8 31.5 - 64.3;        Monitor that can do 1280x1024 @ 60 Hz
 9 31.5 - 79.0;        Monitor that can do 1280x1024 @ 74 Hz
10 Enter your own horizontal sync range
Enter your choice (1-10):
```

Screen output 14.2.2: Entry for the monitor's horizontal frequency

Only if the settings of your monitor are unknown you should take one of the predefined modes. Selection `'10'` lets you enter the correct frequencies.

```
 1 50-70
 2 50-90
 3 50-100
 4 40-150
 5 Enter your own vertical sync range

 Enter your choice (1-5):
```

Screen output 14.2.3: vertical frequencies in detail

Again, using the known values should be preferred to one of the items `'1'` to `'4'`

Next you have to enter a name for the monitor:

[5] Of course these values are not in our liability!

```
Enter an identifier for your monitor definition:
```

the vendor

```
Enter the vendor name of your monitor:
```

and the model

```
Enter the model name of your monitor:
```

Here a name can be given or you select the predefined values by pressing Enter . Setting up your monitor is now done.

14.2.4 Graphics cards / X-Server

Now setup commences with specifying your graphics card:

```
Do you want to look at the card database?
```

If you enter `'y'` a selection of predefined cards is presented.

Here you can select your card by pressing the corresponding number. Do not trust this list blindly since there can be differences in clock-chip and ramdac[6] settings!

That is why later on there is a menu item to select a ramdac and a clock-chip although you have entered it already. Anyway, the predefined settings for this card are set as defaults.

The card definitions contain information on clock-chips, ramdac and the X-Server to be used. Furthermore some valuable information concerning the card is written to the device section in `XF86Config`.

If your card is not listed, do not panic. In this case switch back to the menu by selecting `'q'`. Please do only select one of the defined cards if it matches your card exactly! Selecting a card with a similar name is not recommended. Similar names do not necessarily refer to similar hardware ...

Further information on how to configure your card are described in chapter section 14.3.

After having specified your card, the X-Server is next.

1 A server for monochrome (b/w) monitors. This should run on any VGA compatible graphics card and at least offer a resolution of 640x480.

2 16 colours server XF86_VGA16. Should run with any VGA compatible card.

3 SVGA server XF86_SVGA. This server supports a wide variety of SVGA cards. Graphic acceleration is used with some Cirrus or WD cards. The 16, or 32 bit colour mode can be activated with some Cirrus cards.

4 Server for accelerated cards. Here there are some server at your disposition (see below).

5 This item only exists if you have entered a card definition in the previous selection. Here the server is selected (default) that suits the selected card.

If a server has been selected you are asked whether to create a symbolic link to `/usr/X11R6/bin/X`; if this is answered with `'y'` you are asked whether to put it in `/var/X11R6/bin/X`

[6] **R**andom **A**ccess **M**emory **D**igital-to-**A**nalogue **C**onverter

```
1 The XF86_Mono server. This a monochrome server that should work on
  any VGA-compatible card, in 640x480 (more on some SVGA chipsets).
2 The XF86_VGA16 server. This is a 16-color VGA server that should
  work on any VGA-compatible card.
3 The XF86_SVGA server. This is a 256 color SVGA server that supports
  a number of SVGA chipsets. It is accelerated on some Cirrus and WD
  chipsets; it supports 16/32-bit color on certain Cirrus
  configurations.
4 The accelerated servers. These include XF86_S3, XF86_Mach32,
  XF86_Mach8, XF86_8514, XF86_P9000, XF86_AGX, XF86_W32 and
  XF86_Mach64.

These four server types correspond to the four different "Screen"
sections in XF86Config (vga2, vga16, svga, accel).

5 Choose the server from the card definition, XF86_S3.

Which one of these four screen types do you intend to run
by default (1-4)?
```

Screen output 14.2.4: Selecting a X-Server

```
Do you want to set it in /var/X11R6/bin?
```

Please reply in the affirmative, since it might be not possible to write to `/usr`[7]

Afterwards (if you have pressed `'4'` in the previous selection) there is a menu with all available accelerated X-servers.

```
Select an accel server:

1 XF86_S3
2 XF86_Mach32
3 XF86_Mach8
4 XF86_8514
5 XF86_P9000
6 XF86_AGX
7 XF86_W32
8 XF86_MACH64

Which accel server:
```

Screen output 14.2.5: Accelerated X-server

These servers support the corresponding cards. Assigning this link requires a suitable X-server to be be installed, i. e.the correct X-server has been selected.

After selecting the X-server the graphics card has to be adapted. First you have to enter the memory installed:

Afterwards the name, vendor name and type of your card is requested. If a graphic card has been selected, pressing Enter will do.

[7] e. g.at CD installation

```
How much video memory do you have on your video card:

1 256K
2 512K
3 1024K
4 2048K
5 4096K
6 Other

Enter your choice:
```

Screen output 14.2.6: Selecting memory

```
Enter an identifier for your video card definition:

Enter the vendor name of your video card:

Enter the model (board) name of your video card:
```

If an accelerated X-server has been selected you have to enter the ramdac settings. These are only relevant for S3 and AGX servers.

```
 1  AT&T 20C490 (S3 server)            att20c490
 2  AT&T 20C498/21C498/22C498 (S3)     att20c498
 3  AT&T 20C505 (S3)                   att20c505
 4  BrookTree BT481 (AGX)              bt481
 5  BrookTree BT482 (AGX)              bt482
 6  BrookTree BT485/9485 (S3)          bt485
 7  Sierra SC15025 (S3, AGX)           sc15025
 8  S3 GenDAC (86C708) (autodetected)  s3gendac
 9  S3 SDAC (86C716) (autodetected)    sdac
10  STG-1700 (S3)                      stg1700
11  TI 3020 (S3)                       ti3020
12  TI 3025 (S3)                       ti3025
```

Screen output 14.2.7: Setting a RAMDAC

In most cases a simple Enter will do. If your graphics card has been selected that supports a certain ramdac this should be chosen.

After having answered this question you can enter a clock-chip for accelerated cards (if there is one). Giving a clock-chips avoids clock lines as the clocks needed can be programmed.

If a card without clock-chip is selected, a simple Enter will do (thus not selecting a clock-chip). If a card has been selected the clock-chip is set as default (if there is any).

If no clock-chip has been set **xf86config** suggests running **X -probeonly** for determining the supported clock timings. These are automatically written in `XF86Config` in a separate *Clocks* line.

Here we have to put straight why the automatically defined settings can be *really dangerous*: If the card has a programmable clock-chip the X-server can not distinguish between the different clocks and only recognizes clocks

```
 1  AT&T 20C490 (S3 server)            att20c490
 2  AT&T 20C498/21C498/22C498 (S3)     att20c498
 3  AT&T 20C505 (S3)                   att20c505
 4  BrookTree BT481 (AGX)              bt481
 5  BrookTree BT482 (AGX)              bt482
 6  BrookTree BT485/9485 (S3)          bt485
 7  Sierra SC15025 (S3, AGX)           sc15025
 8  S3 GenDAC (86C708) (autodetected)  s3gendac
 9  S3 SDAC (86C716) (autodetected)    s3_sdac
10  STG-1700 (S3)                      stg1700
11  TI 3020 (S3)                       ti3020
12  TI 3025 (S3)                       ti3025
13  Normal DAC                         normal
```

Screen output 14.2.8: Setting of clock-chip

0, 1, and sometimes 2. All other values are more or less random numbers (normally clocks 0, 1, and 2 repeat and are replace by zeros.)

Well, any clock (despite 0 and 1) depends on the programming of the clock-chip. Thus clock 2 can have a different setting at probing as at runtime (the value entered in `XF86Config`). Then all timings are wrong and the monitor can be damaged severely!

A good hint on a programmable clock-chip (and the problems that follow) are many zeros or repeating timing values. Never ever write such values to `XF86Config`!

For determining the clock-chips commence as follows:

- The best way is to enter an existing (*programmable*) clock-chip (if there is one). Now it is programmed accordingly; your `XF86Config` will not contain clock lines. You can compare chips on the card with the chips offered in the menu. Most modern S3 cards do have a programmable clock-chip.
- If you *do not have a programmable* clock-chip you should launch **X -probeonly** and compare these values with those of the manual. If these values correspond (± 2) you should enter them in `XF86Config`. If there are no hints in the manual you can determine the values by running **X -probeonly** (best on an unloaded machine). Check if the values are correct since clock values can not be determined on every card (many zeros or repeating values are a sign for not valid settings). Enter these valid values into `XF86Config`. Do not omit values, do not try to rearrange them nor change them in any way. The values have to be entered in the exact order.

  If the P9000 server is used order is irrelevant; just enter the modes for the desired clock in the *Clocks-line*.
- In general: If there is a programmable clock-chip there should be *no* clocks-line in `XF86Config` (exception P9000).
  For cards without a programmable clock-chip there should be a *Clocks-line* in `XF86Config`. This avoids the tedious (and sometimes even dangerous) testing at each startup. Furthermore for cards with unreadable values there are no invalid values and there is no risk for your monitor.

So, if (after having read the previous section) you want to let clocks be recognized automatically, just enter `'y'` to the following question:

```
Do you want me to run 'X -probeonly' now?
```

Now the screen turns black and then a list of determined clocks is presented; or a message appears that no clocks could be found. If you have selected a clock-chip this question does not appear since clocks are programmed automatically. In this case this section is skipped.

> If the previous question has been answered with `'y'` and the screen remains black for more than 30 seconds you should cancel testing immediately by Ctrl + Alt + ←, or Ctrl + c respectively! If this does not work switch off monitor and machine so that the hardware is not damaged!

14.2.5 Saving your configuration

So far for installation. Now the configuration file has to be written. It is recommended to do so in `XF86Config` in directory `/etc`. This is to ensure that even in a networking environment each machine has its own configuration file even if they share the `/usr` filesystem.

xf86config first suggests writing `XF86Config` to the current directory. This should be answered in the negative:

```
Do you want it written to the current directory as 'XF86Config'?
```

Then you are asked where to save your configuration:

```
Please give a path+filename to write to:
```

Here you have to enter: "`/etc/XF86Config`"!

Now **xf86config** is done and configuration of your X-Window system is over.

14.3 Optimizing the X Window System

xf86config creates `/etc/XF86Config`. This is the primary configuration file for the **X Window System**. Here you can find all settings concerning your graphics card, mouse and monitor.

`XF86Config` is divided into several sections (**`Sections`**) which serve for configuring of a certain topic. A section always resembles:

```
Section <name of section>
  entry 1
  entry 2
  entry n
EndSection
```

There exist the following types of sections:

`Files`	This section describes all paths used and the RGB colour table.
`ServerFlags`	Here general switches are set.
`Keyboard`	Servers for describing and setting up your keyboard. In our case this can only be `Device`"Standard"
`Pointer`	Assigns all necessary references to the appropriate pointer. In most cases this will be a mouse. Further possibilities are LightPens or GraphicsBoards. Important settings are the `Protocol` and `Device`.

`Monitor`	Describes the monitor in use. Elements of this sections are a name that is referred to by defining `Screen`, as well as bandwidth (`bandwidth`) and sync frequencies (`HorizSync` and `VertRefresh`). Settings are given in MHz, kHz and Hz. Normally the server refuses any modeline that does not correspond with the specification of the monitor. This is to prevent that too high frequencies are sent to the monitor by accident.
`Device`	This section defines a certain graphics card. Its referenced by its name.
`Screen`	This section now puts together a `Driver` (e. g.`vga2`) a `monitor` and a `Device` thus forming all necessary settings for XFree86. Subsection `Display` lets you assign a size to the virtual screen (`Virtual` of `ViewPort` using `Modes`).

We now take a closer look at `Monitor` `Device` and `Screen`. Information on the other sections can be found under [The XFree86™-Team, 1996].

There can be many `Monitor` sections in `XF86Config`. Even multiple `Screen` sections are possible; which one is started depends on the launched server.

14.3.1 Screen section

First we have a closer look at the screen section. This, as mentioned above combines a monitor and a device section and determines which resolution using which colour depth is used.

A screen section can, e. g., resemble the following diagram:

`Driver` determines which X-server it applies to. Servers listed on page 214 are accessed via the keywords .

`Accel`	For special accelerated servers
`Mono`	Not VGA 1 and 4 bit server
`SVGA`	Super VGA server
`VGA2`	1 Bit (monochrome-)VGA server
`VGA16`	4 Bit VGA server

Table 14.1: Keywords for X server in `/etc/XF86Config`

```
Section "Screen"
    Driver      "accel"
    Device      "Miro Crystal 40SV"
    Monitor     "EIZO T563-T"
    DefaultColorDepth  16
    Subsection  "Display"
        Depth       8
        Modes       "1024x768" "800x600" "640x480"
        ViewPort    0 0
        Virtual     1024 768
    EndSubsection
    Subsection "Display"
        Depth       16
        Modes       "1280x960" "1152x864" "1024x768" "800x600"
        ViewPort    0 0
        Virtual     1280 960
    EndSubsection
    Subsection "Display"
        Depth       32
        Modes       "1024x768" "800x600" "640x480"
        ViewPort    0 0
        Virtual     1024 768
    EndSubsection
EndSection
```

File contents 14.3.1: Screen section of `/etc/XF86Config`

There can be a screen section each in `XF86Config` which will be used if the corresponding server is started.

The next two lines, **`Device`** and **`Monitor`**, set the graphics card and the monitor which belong to this definition. These are just points to the Device- and Monitor sections with the corresponding names. This section is discussed further on more thoroughly.

Using **`ColourDepth`** one can set the colour mode to start if it is started without explicitly setting the mode.

Now there is a **`Display`** subsection for each colour depth. **`Depth`** assigns the colour depth of this subsection. Possible values for **`Depth`** are: 8, 15, 16 24 and 32. Not every X-Server supports all these modes. For most systems 24 and 32 are equal, some others take 24 for packed-pixel 24bpp mode whereas others choose 32 for padded-pixel mode.

After the colour depth a list of resolutions is set (`Modes`). This list is checked by the server coming from the left to the right. For each resolution a suitable `Modeline` is searched for which has to correspond with one of the given clock rates or a clock rate to program the card.

The first one that is found is the so-called **`Default mode`**. Using Strg + Alt + grey + one can switch to the right, using Strg + Alt + grey - to the left. Thus enabling you to vary the resolution at runtime.

The last two lines of this subsection refer to the size and anchor of the virtual screen . Size depends on the amount of memory installed on your card and the desired colour depth, not on the maximum resolution of the monitor. If the card has 1 MB video RAM installed the virtual screen can be up to 1024x1024 by 8 bit colour depth. Especially for accelerated cards it is not recommended to use up all your memory for the virtual screen since this memory on the card is used for several font and graphic caches.

Size of the virtual screen is assigned by **`Virtual`**.

`Viewport` assigns a *viewport*. This is the point where the upper left corner of the physical screen is reflected in the virtual screen. Assigning `0 0` means that the upper left corners overlap at startup time. The visible section is moved across the virtual screen by moving the mouse to the edge. Thus the viewport is only important at startup time and even then only if size of the virtual screen differs from the size of the real screen.

14.3.2 Device section

A device section describes a certain graphics card. There can be as many device entries in `XF86Config` as you like, as long as the names of them differ.

We do not go into greater details of the device section. Instead we recommend to use the extensive documentation (`/usr/X11/lib/X11/doc` and the manual pages [The XFree86™-Team, 1996]).

This chapter mainly shows where to get predefined device sections and sets which cards are supported by XFree86.

In `/usr/X11R6/lib/X11/doc/Devices` there is a collection of device sections. If you used a card mentioned in this file you should set the appropriate section in `/etc/XF86Config` and adopt the screen section by entering the specific device.

If the card is not listed this does not necessarily mean that this card is not supported by XFree86! This mainly means that no adequate device section has been sent to the XFree86team. A list of supported cards can be found under `/usr/X11/lib/X11/doc/README`, in `/usr/X11/lib/X11/doc/AccelCards` there is a list of accelerated cards.

14.3.3 Monitor section

Monitor sections describe a monitor analogous to the device sections. Again, there can be as many **`Monitor`** sections as desired in `XF86Config`. Screen section sets which monitor section is relevant.

For monitor definition the same applies: it should only be set by experienced users. A vital part of the monitor section are the so called modelines which set horizontal and vertical timings for the appropriate resolution.

> Without a profound knowledge of the monitors and graphics cards functions nothing should be changed in the modelines since this can lead to severe damage to your monitor!

For those of you who might want to develop their own monitor descriptions the documentation in `/usr/X11/lib/X11/doc` might come in handy. It is recommended to have a look at [Fang et al., 1993] where function, hardware and the creation of modelines are explained in great detail.

Predefined monitor sections that are taken over in `XF86Config` are to be found in `/usr/X11/lib/X11/Monitors`. A monitor not listed there should be driven with VESA standard timings such as set by **xf86config**. It is really important, that horizontal and vertical sync frequencies are set correctly!

If you have a tested configuration, which was not mentioned here, we S.u.S.E. or the XFree86 team will include it in the list.

Chapter **15**

The windowmanager – Window to your machine

Once the configuration of the X-Server is realized, it is evident that the typical user wants to implement a color full desktop with windows, menus and loads of other stuff that an adequate desktop needs.

This chapter deals with window managers[1]. Here the following topics are available:

- The window manager and its tasks.
- **fvwm2** – *the* window manager.
- **fvwm2** – the K Desktop Environment as an alternative.
- **susewm** – a very elegant way to your own configuration file.
- Real life: Adapting and configuring your personal desktop.

Even if you want to go ahead right now, you still need some theory to begin with, so hold on!

15.1 Some theory

15.1.1 General

On contrary to the monolithic graphic user desktops used e. g. in Windows or OS/2, the various functional layers used by Unix or Linux are separated into independent sections. This at first glance makes the system more comples. On the other hand it gains flexibility and the system is able to handle more complex tasks.

The first layer is the operating system, which handles the "trivial" tasks like the memory management for example.

On the next layer is located the ☞*X Server* (X Window System), which corresponds to the "graphics device driver" used in other systems. The X Server serves provides a transparency layer below the GUI itself. Thus you are able to use these services via a whole network (including the Internet).

In a nutshell: The "only" things the X-server can not are:

- correspond with the graphics card,

[1] *Window Manager* is abbreviated as *WM*

- draw dots, lines, rectangles and text, and
- distribute services over the net or on the local host.

Most users run a local X Server anyway but even they may exploit the services it provides. You may launch applications on a server, e. g.in the office, and redirect the output to your local screen. Another example. You don't need to sit in the same root with your noisy, roaring server , just connect a small and smart workstation via Ethernet and you can work on the server remotely.

Now, for being able to display all these graphics stuff such as rectangles and things known as "windows" the services of a window manager are essential.

Windows are rather important for clarity; windows let you start applications in windows next to each other. Menus make the use of your system easy at your fingertips.

The windowmanager is an additional layer between the X-server, your application programs and the user[2].

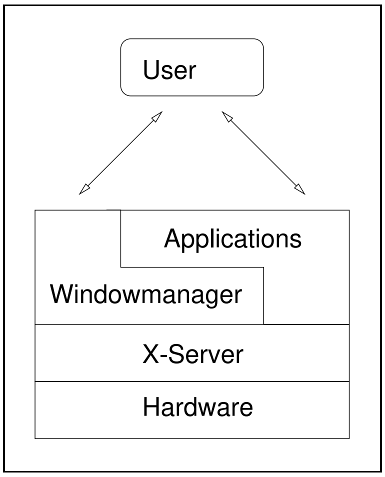


Figure 15.1: Layers of GUIs under Linux

Under Linux there is a wide variety of window managers, e. g.:

- **fvwm**, or **fvwm2**, respectively (*the* window manager)
- **fvwm95** (Win 95 clone)
- **bowman** (look & feel of **NeXTSTEP**)
- **ctwm**
- **mwm** (**Motif** window manager)
- **olvwm** (**OpenLook** Virtual window manager)
- **CDE** – Comon Desktop Environment (commercial)
- **Kwm** – window manager of the K Desktop Environments (KDE)

Besides there are many other windowmanagers available. There are among others: **Wm2**, **Mlvwm**, **Qvwm**, **Enlightenment**, **9wm**, **Twm**, **Icewm**, **scwm**, ...

[2] Of course X application developers may access the server directly; just for clarity

Which window manager you use mainly depends on your wishes and the supplied functions. There are remarkable differences in memory exhaust between the window managers. More over the setup and the flexibility towards new features and updates play an important part. In figure 15.2, figure 15.3 and figure 15.4 you see three examples of window decoration used by Fvwm, Fvwm95 and AfterStep.

Figure 15.2: Fvwm2 window decoration

Figure 15.3: Fvwm95 window decoration

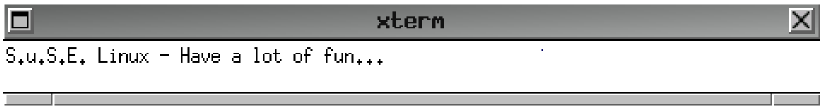


Figure 15.4: AfterStep window decoration

Of course you may test and install the various window managers. If you have decided on one window manager you can adapt it and your personal desktop to your needs.

Most of the WM mentioned locate their configuration files and related data in subdirectories in path `/usr/X11R6/lib/X11`. Please feel free to rummage around!

> A hint: Most of the information in this chapter refers to **fvwm2**, or **KDE** respectively so if you are unsure, we recommend using either of them!

15.1.2 What does a windowmanager manage?

Here we give you a short (and incomplete) list what properties of your desktop you can change via the use of a windowmanager:

- Appearance of your windows
  - Width and height, colours, 3D effects of the window-frame
  - Window-controls, e. g. for moving, enlarging , title, fonts (window decoration), headlines and fonts
- Overlapping of windows
  - Raising of windows (e. g. AutoRaise).

  - Pinning of windows
- Focusing of windows by:
  - Clicking
  - “Touching” with the mouse pointer
- Popup menus
  - Look&feel of menus (colours, fonts)
  - Behaviour of menus and submenus
- Screen background
- Virtual desktop (multiple desktops/screens)
- Icon management
- Linking sounds to actions on your desktop

Unfortunately only a few windomanagers provide a means for configuring themselves. There is no configuration tool as you might expect. Some of them do have a GUI for setup. These are: AfterStep, Olvwm, CDE und KDE.

Most WM read one or more configuration files at startup. The behaviour of the WMs may be set in these files. The syntax itself is more or less mnemonic. It’s a pity that all of the WMs use a different format for saving their configuration files. You are obliged to read the appropriate manpages.

At last you will find “your” WM. You will pay attention to it and configure its configuration files to suit all your needs. The rest of the time should be “work” not only configuration..;)

15.1.3 Starting different Windowmanagers

For starting a WM, S.u.S.E. Linux provides different possibilities depending on how you start your X Window System[3].

Starting via the susewm-Menu

susewm provides a list of menus that may be launched directly from the task bar.

You may switch to and fro between the WMs. Normally the windows on your screen will remain untouched (well the decoration changes). This feature is not available along with ctwm, mwm, kwm and CDE (the developers just forgot to implement it).

The variable $WINDOWMANAGER

On the long run, it is of course annoying if one wants to use for example Fvwm 95 and has to start the pre-set Fvwm2 all the time just for changing over to Fvwm 95 via the menu.

For that reason the environment variable **$WINDOWMANAGER** was introduced for starting a specific WM right at the beginning. The value of the variable **$WINDOWMANAGER** is used for starting a certain WM.

[3] either via the XDM or a text console

Starting with `startx`

First we want to provide you with a way on how to start the X Window System right out of the text console (via **`startx`**).

You can do it for example easily via the following command:

```
newbie@earth: > startx fvwm95
```

for starting Fvwm95 directly. This works for most of the WMs included in S.u.S.E. Linux. This command may be extended by the colour depth of you want to start the rather colour intensive **AfterStep** (**`afterstep`**):

```
newbie@earth: > startx afterstep -- -bpp 16
```

starts the X Window System in 16 bit color mode (65536 colors) using the AfterStep WM. In these examples we always use the name of the WM binary.

If you don't want to enter the name of an WM all the time at the beginning, or if you rather want to use another WM than Fvwm2 as standard, you can insert resp. change the following line in the file `~/.bashrc` in path **$HOME**:

```
export WINDOWMANAGER=fvwm95
```

Here you need to set the WM's binary. You might need to give in the complete pathname especially if `/usr/X11R6/bin` is not included in **$PATH**.

You may as well insert this entry in the systemwide `/etc/profile` for setting it systemwide as default. As any user might overwrite this setting in his/her personal `~/.bashrc` file we at S.u.S.E. did not set it (e.g. with YaST).

Start via `xdm`

Setting an environment variable `~/.bashrc` is the best way to use your favourite WM if you plan to work with the XDM.

> If you start your X Window System via KDM (KDE GUI) you may not set the environment variable **$WINDOWMANAGER** but select your WM from the KDM pulldown menu. See section 15.4.

15.2 The fvwm2 window manager

15.2.1 General

Now we want to deal with **Fvwm2** (**`fvwm2`**), a windowmanager that soon became *the* windowmanager under Linux, or XFree86™ rrespectively.

Fvwm2[4] is the successor of the old fvwm1. It now needs much more memory than the older version but provides lots of new functionalities and configuration possibilities.

Besides those usual functions for managing windows (sic!) and look & feel of the buttons (desktop) it now provides background menus and modular applications which can be loaded at runtime. Those features provide interesting functions as, e. g., a button-bar.

[4] In S.u.S.E. Linux **fvwm2** is packed under package `fvwm`, series `xwm`; the previous version **fvwm** in package package `fvwm1`, series `xwm`

More information about functions, starting and configuration of **fvwm2** and its modules is reached via the corresponding manual pages (manpage of **`fvwm2`** (**`man fvwm2`**), manpage of **`FvwmAudio`** (**`man FvwmAudio`**), manpage of **`FvwmButtons`** (**`man FvwmButtons`**), etc.) or under `/usr/doc/packages/fvwm` which automatically is installed when installing the fvwm package. In any case of doubt you should consider looking at those documents first.

> As the original packages of both Fvwm and Fvwm2 use the same place for their manual pages (manpages for **fvwm2** can be viewed as usual), there is a special action to be taken to view them. You can read the manpages of Fvwm2 as usual with the read command. The manpages of Fvwm (version 1) and Fvwm95 have been located in different paths under S.u.S.E. Linux.

For reading an Fvwm (version 1) manpage (e. g.) you should proceed as given:

```
fvwmman FvwmButtons
```

for reading a Fvwm95 manpage (e. g.) you should enter:

```
fvwm95man FvwmButtons
```

Then you will get access to the manpage to **FvwmButtons** of the corresponding package. You also can view the manpages of the different WM right out of the menu. You can find this "special" manpages (if available) in the menu `'Windowmanager'`, submenu `'Manpages'`.

If you don't want to start from scratch you may generate a "frame" with **susewm** and tailor it to your needs afterwards. **susewm** is described in section 15.5.

Configuration of fvwm2

The Fvwm2 can be configured via two files:

- A system wide configuration file which should be available at any time: This is `.fvwm2rc` and lies under `/usr/X11R6/lib/X11/fvwm2`.
- A user specific file which is not really necessary (but recommended). This file is called `~/.fvwm2rc` and lies in the home directory of the user.

The package **susewm** (described later on) ensures the existence of a system wide configuration file for **fvwm2**.

fvwm2 reads these configuration files at startup time[5] First **fvwm2** tries to read the user's configuration file; if this doesn't exist it reads the system wide file.

All **fvwm2** modules read this file too, but they only use the statements corresponding to themselves.

It is recommended for every user to create his/her own configuration file which he/she can change and adapt to his/her personal needs.

[5] In general the windowmanager is either started from `/usr/X11R6/lib/X11/xinit/xinitrc` or from the user's private configuration file `~/.xinitrc`. The environment variable **$WINDOWMANAGER** should be set to the full pathname of the corresponding window manager.

> After changing configuration files the WM has to be restarted for the changes to take effect.

For restart there is a menu entry (supplied with pre-installed S.u.S.E. Linux configuration) in `'Working menu'` called `'windowmanager'`. You could as well restart the X-server. It is also possible to restart the WM from within an xterm or a text console. Just enter:

```
newbie@earth: > killall -10 fvwm2
```

Interactive changing of configurations in **fvwm2** is only available via module **`FvwmConfig`**, this tool however only offers very limited capabilities concerning the configuration since those modifications can not be stored. Besides this one there exist two more modules – FvmSave and FvwmSaveDesk – which in fact can store the actual state of the WM. However, their files can not be read automatically at the next start of the WM. If you're interested, please read the Manpages of the respective module.

15.3 Fvwm2 settings

15.3.1 General

Now we shall deal with your personal Fvwm2 configuration file. Of course you should have created one in advance. The howto is listed in section 15.5. Besides the file created by **susewm** you may also use the configuration file provided by the authors of Fvwm2. This may be found under `/usr/doc/packages/fvwm/system.fvwm2rc`.

Now start up your favourite editor and load the file `~/.fvwm2rc`. We will have a look at some of the options.

What happens when fvwm2 starts

Scroll through the text until you reach the following comment:

```
##########################################################
#                                                        #
#   initialization function head                         #
#   common to all wms                                    #
#                                                        #
##########################################################
```

File contents 15.3.1: **`InitFunction`** in `/.fvwm2rc`

Hereafter you will find anything that is launched at a *restart* of **fvwm2**. Here the banner is loaded (**FvwmBanner**, a couple of xterms and an **xpmroot** is launched. This program simply serves for putting images onto your root windows . For that purpose you can take any program that is capable of writing onto the root window (e. g. **xli**, **xv**, etc.).

Here a sample with **xv**:

```
+ "I"  Exec xv -quit -root -owncmap -maxpect ~/pics/bild13.gif
```

You should list this background picture also in the sections, which are executed with the restart of **fvwm2**, which means you should insert them within **Restart Function**. This is located under:

```
##########################################################
#                                                        #
#   restart function                                     #
#   common to all wms                                    #
#                                                        #
##########################################################
```

File contents 15.3.2: `InitFunction` in `/.fvwm2rc`

Often, the two sections (`InitFunction` and `RestartFunction`) look the same, since they both deal with starting the WM. In the `RestartFunction` function one normally does not include fvwm's banner.

Newer versions of fvwm (Fvwm, Fvwm2, Fvwm95, Bowman and AfterStep in S.u.S.E. Linux5.0 and later) have an additional function `ExitFunction` (besides the usual `InitFunction` and `RestartFunction`). This function enables you to set programs that have to be started *before* the WM is launched, or *before* you exit the WM. So you can (e. g.) remove a background picture before restarting the WM sets a new one.

Colours and fonts

The settings for colours and fonts can be found in the following section:

```
##########################################################
#                                                        #
#   colors and fonts                                     #
#                                                        #
##########################################################
```

File contents 15.3.3: Colour and keyboard settings in `/.fvwm2rc`

Here you can do whatever you like. Select those colours that you like best. You can use any installed colour. Which colours are installed mostly depends on your graphics card and the colour depth. Press the right mouse button in the root menu and go to the item `'system'`; here you change to `'information'`. Far at the end you should see an icon `'colours'`. Start it and it will show you all colours known to your system. All names of these colours can be found under `/usr/X11R6/lib/X11/rgb.txt`.

A few lines below are lines that resemble the following one:

```
WindowFont  -misc-fixed-bold-r-normal-*-13-*-75-75-c-80-iso8859-1
```

This is the systematic description of a font under X11. Every font is classified in this style. It would lead far beyond the scope of this book to explain all those items. For making life easier there are a couple of well known *aliases*. Fonts reside (e. g.) under `/usr/X11R6/lib/X11/fonts/misc`. In this directory there is the file `fonts.alias` which contains all font aliases for this font directory. Some most common fonts:

```
variable  -*-helvetica-bold-r-normal-*-*-120-*-*-*-*-iso8859-1
5x7       -misc-fixed-medium-r-normal--7-70-75-75-c-50-iso8859-1
```

It's much easier to remember these names … `:-)`

Icons

Icons are dealt with in the following environment:

```
#
# others
#

Style  "xterm"          Icon Terminal.xpm
Style  "xosview"        NoTitle, Sticky
Style  "xosview"        UsePPosition
```

File contents 15.3.4: Icons for certain windows

Here you can assign icons to the corresponding applications. You only have to be sure that these icons are in the **`IconPath`** (which is set almost at the beginning of `~/.fvwm2rc`). In principle you can take any icon that resides under `/usr/X11R6/include/X11/3dpixmaps/` (that's an example, there are more). All other icons need their absolute path name.

Now, let's assume you want to assign a ghost icon to **ghostview**. Look in the directory mentioned above and you will see an icon `ghostbusters.xpm`. Just insert the following line:

```
Style   "ghostview"   Icon ghostbusters.xpm
```

That's it. You can give icons to almost all applications in the same way. Most applications do have their default icons (remember application defaults?). Anyway, you have to know the *exact* name of the application[6] since an "xTerm" instead of "xterm" would lead to no success. The name of a certain window can be identified from within (`'Windowmanager'`, `'Module'`)by selecting `'Ident'` (Program name **FvwmIdent**) and clicking on the window of interest.

Cursor

Even the shape and colour of the mouse cursor can be set. Here you have the tool **xsetroot** (which can be used for setting the root window as well in a rather simple way). It is invoked as:

```
newbie@earth: > xsetroot -cursor <bitmapfile>
```

Here **`bitmapfile`** stands for any bitmap file. The bitmaps are located under `/usr/X11R6/include/X11/bitmaps/` by default. Select a suitable one or create your own, e. g.with **bitmap**.

[6] To be precise: the exact name of the application since the WM takes care of your windows.

Focus

An extremely popular feature of the Fvwm-WM-Family is, that one change the behaviour of the windows while changing the active window while the system is up an running. One can change the focus & raising policy of the windows in the configuration file.

With focus we define the properties of the WM, which submit the actual entries and mouse clicks to a certain window. Basically, there are 3 possibilities:

- You have to click on a window for, i.e., doing key entries to the process active in the window. In English, this behaviour is called *Click to focus*. It's a widely spread standard and can also be found under Windows or OS/2.
- You point with the mouse cursor on a certain window and the window receives automatically the focus. This reaction is called *Focus follows mouse*, If the mouse Pointer leaves the window, also the focus leaves the window, even in the case the cursor is located over the background or on another window.
- A more advanced version is the *Sloppy Focus* behaviour, which can be configured with Fvwm2 and Fvwm95. In this case, the focus shows the same behaviour as *Focus follows mouse*, but with the exception that the focus stays with the focused window as long as one does not focus on another window. The focus even stays with window touched if you move the mouse cursor over the background.

With the Fvwm2, you can set the focus behaviour individually for every single window. Like many other settings, the configuration of the focus is done with a style-command:

```
Style "*" ClickToFocus
```

adjusts the focus setting **`ClickToFocus`** for all windows. The same may be done for the settings **`SloppyFocus`** and **`FocusFollowsMouse`**, the latter being the default of Fvwm2.

Autoraise

Autoraising of windows is an interesting feature. Normally the windows keep their position while the focus changes until you click onto the title bar. Autoraise raises the focused windows to the top of the stack. Obviously this only makes sense along with **`FocusFollowsMouse`** or **`SloppyFocus`**. There is no effect if you use **`SloppyFocus`**.

For being capable to use AutoRaising, you can either start the Fvwm module AutoRaising out of the menu (Menu `'Windowmanager'`, `'Modules'`, `'AutoRaise On/Off'`, or, for installing it permanently, you need to insert an entry into Fvwm's configuration file ((`~/.fvwm2rc`) into the functions **`InitFunction`** and **`RestartFunction`**:

```
Function InitFunction
  +     "I"   Module  FvwmAuto 200

Function RestartFunction
  +     "I"   Module  FvwmAuto 200
```

The value `200` sets the delay to 200 milliseconds, before a window is popped to the foreground. If you don't have a delay set every touch of a window leads to a "movement" of windows which will result in permanent flicker.

15.4 KDE – the K Desktop Environment

KDE offers a means of GUI that is easy to configure and standardized. Lots of applications use the Look&Feel of KDE. KDE stands for "K Desktop Environment" and is a project that was founded in 1996.

KDE comes along with a windowmanager (kwm), a filemanager (kfm, which plays a central role), a systemwide helpsystem (kdehelp). This helpsystem supports HTML manpages as well as GNU info. Lots of applications are available either (among others these are mailers news readers, games, system info tools and many more).

Well, please be aware that KDE is still considered beta (as seen in the version). It may happen that it crashes. Luckily this doesn't happen all that often . . .

KDE is totally ☞*URL* based. This in particular means that every path and links to files use a standard form (e. g. links to an HTML page, a file in your filesystem, a help page or an FTP site). Thus it is possible to view different files and formats with the same viewer.

Furthermore KDE is capable of lots of Drag&Drop features (e. g. copying a file from an FTP site to your local system).

A rather powerful application is the KDE helpdesk. All authors are obliged to provide their help pages as HTML pages. Besides the help system itself (this may be launched from the applications itself via the `'Help'` button) it is capable of browsing even GNU info pages or UNIX manpages.

One of KDE's interesting features (among others) is its integrity. It is rather simple to create icons and place them onto your desktop.

This behaviour of KDE is realized by means of the QT widget set. This, similar to Motif, is a set of graphical libraries that enable you to create GUIs on the X Window System. Please have a look at the licence conditions of QT since they differ from GPL (`/usr/doc/packages/qt/LICENSE`).

15.4.1 General

Here we give you a short overview on details of the KDE installation. This refers mainly to path settings location of files and configuration options.

By default KDE is written to `/opt/kde`. Every KDE application (file) may be found below this subdirectory. Setting an environment variable **$KDEDIR** makes it easier to access them (in `/etc/profile`).

The KDE binary path `/opt/kde/bin` is automatically added to your **$PATH** variable when you install KDE.

There are many subdirectories under `/opt/kde`. Here in table 15.1 we only discuss the most important ones.

`/opt/kde/bin`	every KDE programs (binaries)
`/opt/kde/share/config`	systemwide configuration files
`/opt/kde/share/applnk`	Appliaction links (Menu)
`/opt/kde/share/apps`	Files of KDE programs
`/opt/kde/share/doc`	the online help system
`/usr/doc/packages/kde`	additional information on KDE

Table 15.1: KDE – important directories

15.4.2 kdm – graphical login

The KDM displaymanager that comes along with the KDE system is a nice enhancement or your Linux system. This tool replaces the default **xdm** which realizes a graphical login. KDM's default configuration with S.u.S.E. Linux is shown in figure 15.5.

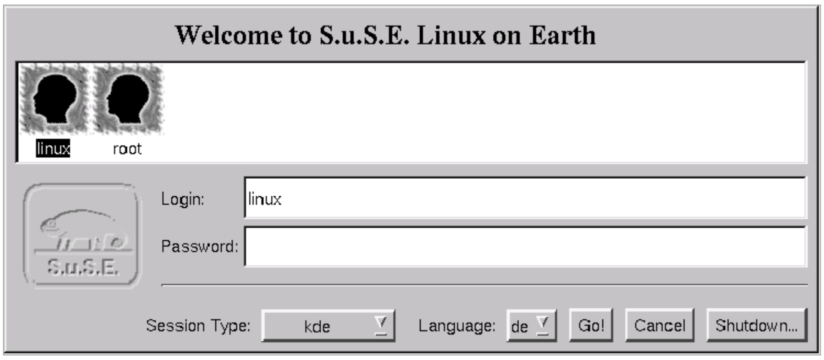


Figure 15.5: The display manager **kdm**

There are buttons for selecting the desired window manager (`'Sessiontype'`) or the language (`'Language'`). An interesting feature (especially for standalone workstations) is the ability to shutdown the machine via the `'Shutdown'` button.

The kdm may be adapted to your needs via either a configuration file in `/opt/kde/share/config/kdmrc` or (since Beta3) from a configuration dialog which may be launched from the KDE menu.

S.u.S.E. Linux provides another enhanced technique. This refers to the configuration of the WMs themselves and the startup mechanism of the kdm.

Here you should create two variables:[7]

- **`DISPLAYMANAGER`**
  Assigns whether the user wants to log into a text console (`console` or empty string this is "runlevel 2") or via `kdm` or `xdm` respectively ("runlevel 3".

[7] These variables are described in page 174

- **`KDM_SHUTDOWN`**
  Assigns which user is permitted to shutdown the machine from within kdm. Here you may enter either `'root'` (`root`) every user (`all`), nobody (`none`) or a user on the local machine (`local`).

These values are written to `/opt/kde/share/config/kdmrc` by **SuSEconfig** and are available with the next start of kdm (you may want to restart the server by entering Ctrl Alt ←). If you plan to make changes yourself please make them in `kdmrc.in` as `/opt/kde/share/config/kdmrc` is created from this file by **SuSEconfig**.

> If you want to get rid of this mechanism just delete or remove `/opt/kde/share/config/kdmrc.in`. **SuSEconfig** will not touch your `kdmrc` anymore

15.4.3 So, what's so special with KDE?

Now that we are able to use kdm's graphical login it is time to go into the details of KDE. We just describe the behaviour of KDE after you have started the server by entering **`startx`** or via the kdm.

Shortly after the actual login a couple of dialog windows appear announcing the missing of certain files and directories. This is not an error. KDE creates similar to the tree under `/opt/kde` its configuration directories in your home directory. These in particular are: **$HOME**`/.kde` and **$HOME**`/Desktop`. The first is for configuration file and the latter for links. Thus you may just hit ↵ on the dialog boxes. They won't appear at the next KDE session.

Thereafter the **kfm** (K filemanager) is launched. The kfm is a basic component of the KDE system. As mentioned above it is able to handle lots of different file formats. The behaviour is totally transparent to the user; be it a archive from an FTP site(URL prefix: `ftp:`) or a manpage (URL prefix `man:`), a GNU info page (URL prefix `info:`) or a HTML page from a WWW site (URL prefix `http:`). It is even possible to view images (via an external tool). A typical KDE desktop is shown in figure 15.6.

The user will get familiar with these features within minutes. Clicking on a symbol with the right mouse button opens a configuration tool for this application.

Every setting of the program itself is written to a file with the `.kdelnk` extension. Changing of settings is done via an property window as shown in figure 15.7. This file is an editable ASCII file. It is recommended to have a look at these files that are located under **$HOME**`/Desktop` (or under `/opt/kde/share/applnk` for the systemwide configuration files). The contents are standard to every KDE applications. So if you know how to write one of them you are set.

The last program to be launched is KPanel the menu bar. Here you may find some elements for the virtual desktops. A special one is the "K" button. Here all available KDE applications are grouped together (see `kdelnk` files!). Here you may also find the KDE system settings via the KDE control centre and many more.

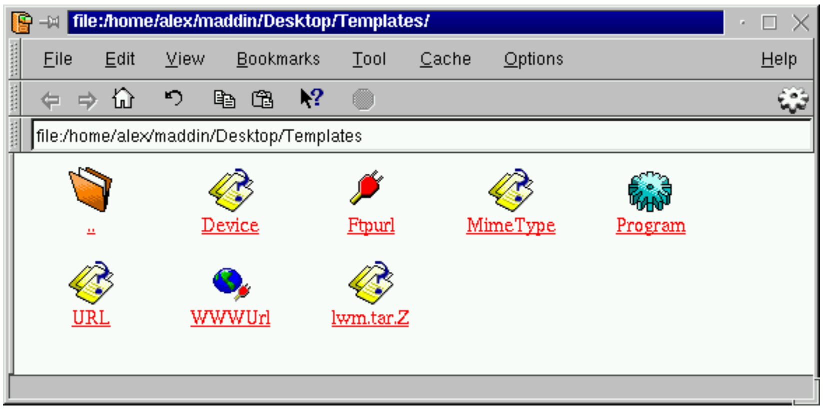


Figure 15.6: The filemanager **kfm**

Well, that's not all KPanel has to offer. You may copy application icons via Drag&Drop onto it and launch them directly from the panel. Moreover KPanel may "swallow" applications. These might be system information applications such as the kwmpager or an **xosview**. These now run within the KPanel.

Another one of KDE's programs the **kdisplay** is shown in figure 15.8.

Of course we cannot list each and every KDE application in this manual. One of KDE's aims is to make the machine more usable and cooperative. Please have a look around at the various applications that come along with KDE.

There is help available with most of KDE's applications. You may reach the kdehelp by clicking onto the background (right mouse button).

But please keep in mind that some of the KDE packages are still considered alpha. This in particular means that they may crash or other strange things may happen. In general this should not happen anymore.

Due to the complexity of KDE and the steady development theses packages are not yet included in our installation support. Anyway we try to provide you with as much information as we can gather on our Support Database. Furthermore there are a couple of mailing lists available on the KDE WWW site (`http://www.kde.org`). Here you may as well find actual information on many topics as well as the latest release of KDE (and applications).

We at S.u.S.E. provide you with specially adapted KDE packages that may be downloaded from our FTP site `ftp://ftp.suse.com/pub/suse_update/KDE`. These packages may be installed by YaST. Please have a look at the README files that are located on the same URL.

15.5 Configuring the windowmanager using susewm

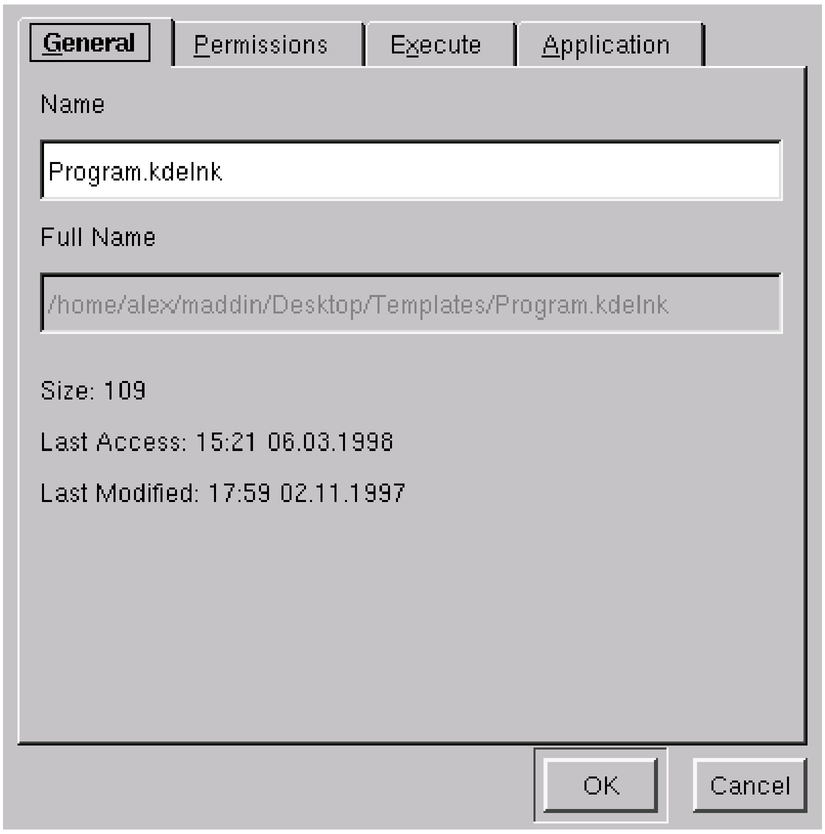


Figure 15.7: The property dialog

What is susewm?

susewm serves for making the handling of the supported WMs (**Fvwm**, **Fvwm2**, **Fvwm95**, **Bowman**, **AfterStep** (`afterstep`), **Ctwm** und **Mwm**[8]) easier[9], **kwm**.

Since **fvwm**, **bowman** and **afterstep** resp. **fvwm2** and **fvwm95** are mainly based on the same WM where **fvwm2** is the descendant of **fvwm**, those five window manager are mainly configured the same way and supply almost identical features.

On the other hand there are considerable differences in configuring those WM's. For letting the user benefit of all common tasks of those WM without having to maintain five different configuration files, **susewm** combines their configuration via a rather abstract macro language.

[8] MWM is part of the commercial **Metrolink Motif** or **Metrolink Motif Runtime Libraries** respectively.

[9] The pseudo-WM **CDEsim** (`cdesim`) should be excluded here. More information on this topic can be found in package `cdesim`, series `xwm`, in directory `/usr/doc/packages/cdesim`.

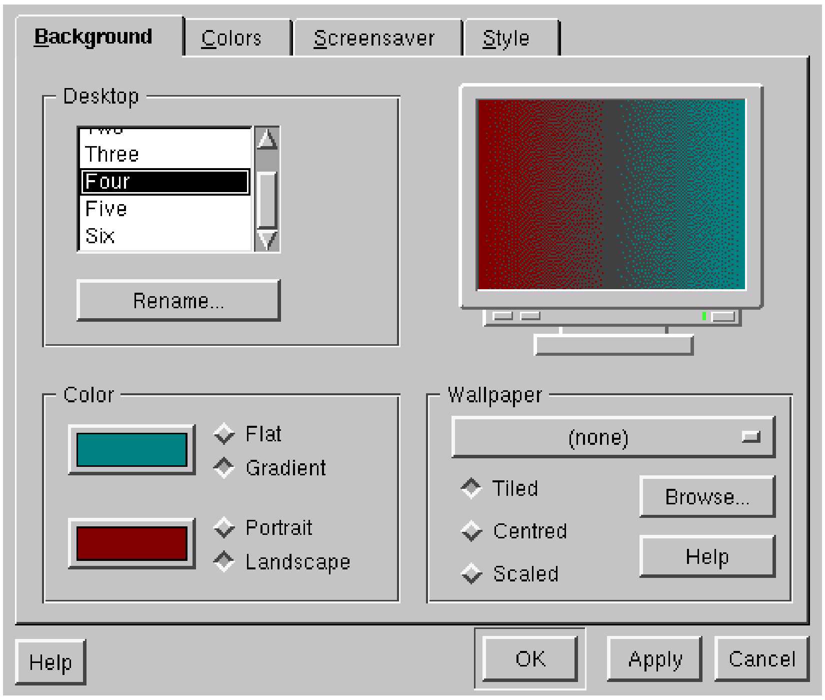


Figure 15.8: Setting up your display with **kdisplay**

And even more, the totally different WM **ctwm**,**mwm**and **kwm** are configured with **susewm** as well. However this only covers the automatically generated menu entries (more in the sections below).

Differences between those single WM's can be taken into consideration through WM specific statements.

One of the most interesting features **susewm** offers is that it creates menus and supported modules depending on the installed software packages. There are no menu entries for packages which are not installed. If you select a menu item you can be sure that the corresponding program is installed and can be launched.

Here are some more of **susewm**ś features:

- Configuration of eight windowmanagers: **Fvwm**, **Fvwm2**, **Fvwm95**, **Bowman**, **Afterstep**, **Ctwm**, **Mwm**, **kwm**
- Central library for all WM's; unified macros for different configuration files.
- Additive configuration files for taking specialities and peculiarities of the WM's into consideration.
- Loading of additional source files[10] in your system via `/etc/rc.config` in automatic configuration.

[10] Source files are WM configuration files in general. For using them with **susewm**, files in **susewm**- format are particular suitable

- Considering source files of the respective user (in HOME directory) during automatic configuration.
- Creation of user-specific configuration files respecting almost any changes the user has made (in older configuration files).
- Unified commands for all eight WM's.
- Integrating of widely used commercial applications, which are not part of S.u.S.E. Linux, while generating menus.
- Switching to and fro between windowmanagers (only for the supported ones) without having to change, e. g., startup scripts.
- Clearly designed **m4** macros.

Technically **susewm** is a set of several shell scripts and m4-macros as well as a database. Details are not covered on this section. If you to know more about **susewm**, please refer to the online-documentation, like described in section 15.5.

That's how to use susewm

susewm is used in two different cases:

- You don't have a configuration of your own file yet but want to adapt your desktop accordingly.
- You already have a configuration file but software has been installed using YaST and the menus are no longer up to date.

Requirements

Here the following is assumed to be installed:

- Your Linux system and the X Window System XFree86™ are installed, configured properly and running.
- You use the **fvwm2** WM (which is the default with S.u.S.E. Linux), but perhaps have another supported WM installed.
- **susewm** is installed on your system (standardsettings).
- You are logged in as normal user (e. g. the sample user defined with the standard installation of YaST), not as `'root'`.
- You have set the language to be `English`. The same applies for german menus.

This is how it is done. If you click onto the root window (which actually is the "background") with your left mouse button a menu appears. This is called the `'Working menu'`. Here the most often used programs needed during your work life are listed (depends on the user actually).

The last entry of this menu (`'Window manager'`) contains a submenu with items on functions and configurations of the actual running WM. If you want to configure another WM just switch to it using menu item `'Other window managers'` and configure it accordingly in the `'Configuration'` item:

You'll find several different items:

- The items of the first window are for reading the documentation on **susewm**. There you can find lots of details that are not handled in this section.
- The items of the second window allow you to change the configuration file of your PC with an ☞*editor*
- The items of the third window help in creating a user specific configuration file in German.
- Also located in the third window is an item for creating a user specific configuration file in English.

This can be seen in figure 15.9 with the 'Working menu' of **fvwm2**

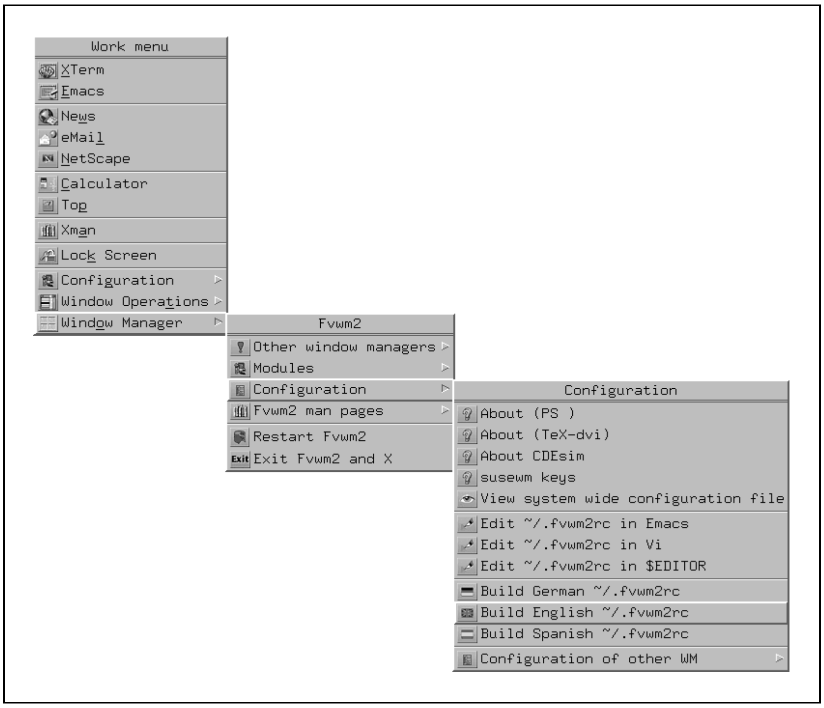


Figure 15.9: Menus for WM configuration

If you select the second or third item a window with **mksusewmrc** (this program is a part of **susewm**) appears. Just read the text and press the appropriate buttons if asked for it.

That's it. Now you can change and edit this freshly created config file in your HOME directory. **susewm** will take over almost everything in the new configuration file.

> If you want to change the settings of the desktops *slightly* you have to create an configuration file of your own first. How this is done is mentioned above. Only after doing so, you can change the file. If you plan to write a *completely different* configuration file, don't it by using **susewm**.

An example

A coloured desktop is nice, but what's the use if the hardware (memory, processor, hard disk) is not fast enough? It's not the point of a WM to use up all resources for animating icons and more.A hint what to do if you think that your machine is too slow if you start up **fvwm95** or **fvwm2**. Most problems are due to the hundreds of icons that are loaded. There are now two possibilities:

- Deinstall the package package `3dpixms`, or package `3dpixm` if you don't want to have the big icons either.
- Set in the YaST menu `'System Administration'`, submenu `'Change the config file'` the variable **`SUSEWM_XPM=no`**, which results in the same effect as mentioned before. In case you have changed the variable directly in the file `/etc/rc.config`, please don't forget after doing so to start **SuSEconfig**.
- Remove icons from configuration file.

The first way is rather simple. Just start YaST, deinstall the mentioned packages and feel fine. YaST in conclusion with **susewm** ensures that the system wide WM configuration file is adapted. If you have a user specific configuration file you have to adapt it explicitly. Use the WM menu as described in section 15.5.

The second way doesn't need to be commented further.

The third way: Let **susewm** create a configuration file for you as described in section 15.5. Then you have to replace all entries that contain icon statements:

```
AddToMenu thiswmpopup "Fvwm2"   Title

+ "Other windowmanagerr%small.warning_3d.xpm%" Popup otherwmpopup
+ "Configuration%small.checklist2_3d.xpm%"      Popup susewmpopup
+ ""                                            Nop
+ "Fvwm2 Restartt%small.restart_suse_3d.xpm%"   Restart fvwm2
+ "Exit Fvwm2 and%small.exit.xpm%"         Function QuitSave

# end popup thiswmpopup
```

File contents 15.5.1: `.fvwm2rc` with icons for menu entries

will become:

Now **fvwm95** or **fvwm2** should start even faster. And, as mentioned before, all personal changes you have applied to your configuration file should persist even if you invoke **susewm** a second time.

15.6 General

As promised above we are now going to tailor the look and functions of your personal desktop – why buy something predefined if we can tailor it ourselves.

First some general information about configuring the desktop – there are two places where you can fiddle:

```
AddToMenu thiswmpopup "Fvwm2"   Title

+ "Other window manager" Popup otherwmpopup
+ "Configuration"         Popup susewmpopup
+ ""                      Nop
+ "Fvwm2 Restart"        Restart fvwm2
+ "Exit Fvwm2 and X"   Function QuitSave

# end popup thiswmpopup
```

File contents 15.5.2: `.fvwm2rc` without icons for menu entries

- The pre-seetings for the applications of the X Window System
- The WM configuration file(s) as already mentioned in section 15.3

15.6.1 X11 defaults for applications

Global settings

Almost any application under X11 has their own default configuration. This configuration will be copied to `/usr/X11R6/lib/X11/app-defaults` at installation time.[11]. Here files as `Xarchie` can be found. As the name already tells us this must be the central configuration file for the application **xarchie**. Just have a look at this file (e. g. with **`less Xarchie`**). Here you can find (among others) the following lines:

`Xarchie.color*background:  powder blue`

Don't bother with the sometimes cryptical names; you don't have to understand them right now. Every program under X11 is built out of "Widgets"[12]. Here there is a *main widget*, which is the root window (it is called first). All other programs are *child widgets* of this parent widget (more or less). That means that every child widget has got exact *one* parent widget and one or more *child widgets*. Each of these widgets can be addressed with an unique name.

Windows and widgets should not be confused. A scrollbar, e. g., is a window of its own (a window which lacks any decoration). A widget can contain a scrollbar, a text field and more.

That leads to the conclusion that any window and its resources can be addressed by a unique name. For our example this means:

- The first word to the first dot (`Xarchie`) is the name of the toplevel widgets of the **xarchie** program (it is a *rule* to write names of application defaults beginning with an uppercase letter).
- After the dot there is a `'color'`. This, of course, is the colour. But which colour?
- Then there is a `'.'`. This dot could have been an asterisk (`'*'` as well).
  - A `'.'` means that this is a direct hierarchy.

[11] It is quite helpful to assign an alias to such abstruse and long names

[12] Think of a "Widget" as a sort of "brick"; it's a shortage for "windows" and "gadget"

- An '*' is a wild card. It indicates that between those two windows may lay 1 or more additional windows.
- The word "background" now shows us which colour is going to be set. Here you can enter a defined colour.

User specific settings

Obviously there is the possibility for every user to make specific settings. For that reason you have a ~/.Xresources file located in the **$HOME** directory. The '.' at the beginning means that this is a hidden file.

In this file, all user specific settings are done. For example you can define now that every window should have a yellow background except the root window which should be red.

Coming back to the example given above this means that you can overwrite system wide settings (app-defaults files) with those in your ~/.Xresources file. So, if you set the following in your ~/.Xresources file:

```
Xarchie.color*background: gold
```

the application **xarchie** is started with a golden background.

Of course you can not only change colours but almost any decoration or setting of your windows. A rather useful program in this context is **editres** (editres stands for EDitRESources). This little tool lets you show all resources of a given program and to change them at runtime (so you can see what you've done).

Here some examples of what you can set:

```
Xarchie.color*background:                 powder blue
Xarchie.color*SimpleMenu*background:      wheat
Xarchie.color*Command*background:         wheat
Xarchie.color*MenuButton*background:      wheat
Xarchie.color*Text*background:            wheat
```

Setting start parameters

There is a third alternative of influencing the application. You can give parameter to it at start time. Of course you can enter those parameters in your WM's configuration file if you start them from there.

For example you can explicitly launch a program with another font and background colour (bg = background, fg = foreground):

```
xterm -bg darkblue -fg white
```

You get a blue xterm with white letters.

How do the configuration possibilities "work"?

Principally, the systemwide settings are set at the start of the X Window System. The settings themselves are administrated by the X-server in the X Resource Database, xrdb. If you want that the changed settings become effective systemwide, it is necessary the read in the resource files once over manually. This can be done with the command:

The various settings for applications are handled in the following order during the build-up of a an intern X-server Resource Database at the X-server start:

Hierarchy of configurations

- First all system wide settings from `/usr/X11R6/lib/X11/app-defaults` are read.
- If you have special settings in your `~/.Xresources` file these will overwrite the settings above.
- If you start an application with certain parameters these settings overwrite the last one. They have the highest priority.

> In case this explicit changes have no influence of the behaviour of the application it might help to vary capitalization of the respective widget name

Chapter 16

Networking Linux

16.1 Preliminary words ...

In the age of communication the number of linked machines grows day by day. Networking becomes a “way of life”. Either on a local ☞*network* (☞*LAN*), on the ☞*internet* or simply connected to a mailbox; one has to communicate. ☞*electronic mail* , news, and participating in world wide discussion forums now becomes a *must* for all who want to be up to date. Linux, being a real child of the internet, offers you all preconditions which are necessary to connect to the multiple networks. All common networking tools are available under Linux thus letting Linux-machines participate in every network.

Here we will give you an overview of all tools needed for installing and maintaining your network connections; the most important configuration files will be presented and discussed and some of the more important tools are shown.

First of all we want to show how a Linux-machine can be integrated in an existing ☞*LAN*, or how to create a network of Linux-machines respectively. Thereafter connections to other hosts via modems are explained. We introduce UUCP, show how you can install an internet connectivity via SLIP or PPP; furthermore configuration of mail and news system is discussed thoroughly.

Almost everything to do with networks can be done by YaST. (see section 4.7, page 82 and section 12.6, page 174); on the other hand configuration of a network can become very complex, that’s why we only show basic mechanisms and all relevant files.

16.1.1 Good neighbourhood – connecting to a LAN

Connecting a Linux-machine to other Unix-machines (of course there can be other Linux-machines among them as well) via a LAN doesn’t impose a problem. There have to be (some) preconditions given, which do not restrict the use of a Linux-machine in a networking environment.

16.1.2 Requirements, preliminary work

At the meantime Linux supports almost any networking cards (Ethernet, Arcnet, TokenRing) and knows almost any known networking protocols (TCP/IP, AppleTalk, IPX). Taking every possible configuration into consideration is, of course, far beyond the aim of this chapter, therefore we will discuss the *generic* case (integrating Linux into a TCP/IP network via an ethernet card). The latest information about other networks is to be found under `Documentation` at your kernel sources directory; moreover the kernel's help-function supplies you with very useful additional information.

The following requirements have to be set:

- Your machine has to have a supported card; whether your card has been started correctly is to be seen through the following command:

  ```
  root@earth:/ # cat /proc/net/dev
  ```

  There should be a line beginning with **`eth0:`**.
- The kernel has to be configured for the net to be used correctly (see section 13.1, page 181 for more details).

> If the kernel support is realized by modules (see section 13.2, page 181) (all S.u.S.E. kernels are configured that way) the name of the module has to be entered in `/etc/conf.modules`. For your first Ethernet card this might look like:
> **`alias eth0 tulip`**
> This is done automatically if you selected the network module in **Linuxrc** at the installation. Later this may be done with YaST. (see section 4.7.1, page 82).

If those preconditions are given there are some things to be thought of before starting configuring your machine:

Hostname	The name of your machine in the network. Hostname should not exceed 8 characters and should not already be used in the local net.
Domainname	Name of the domain your machine belongs to. Domains serve for structuring big networks (e. g.the internet). A host is addressed via its *full* name which is assembled of hostname and domainname. `Gauss.Suse.de` is the machine `Gauss` in the domain `Suse.de`.

Table 16.1: to be continued...

IP-address	Address of your machine in the net. Every machine at least has got one IP-address . This address is a 32 bit word, and in most times is given as a quadruple (e. g.`193.141.17.1`). If you choose an IP-address you should consider whether you plan to connect to the internet in the near future. If this is true it is recommended to use registered IP-addresses from the very start. If you only have a private local network there are some address ranges defined by the pseudo-standard RFC1597 which ensures you that even if one of those addresses "escapes" to the internet it won't hurt anybody or cause any damage. 10.0.0.0 ⇒ 10.255.255.255 (Class A) 172.16.0.0 ⇒ 172.31.255.255 (Class B) 192.168.0.0 ⇒ 192.168.255.255 (Class C)
Gateway address	If there is a Gateway computer in your network (i. e.machine that is connected to more than one network and which transports packages from one network to the other) you can enter the appropriate address while configuring your network.
Netmask	By means of the netmask it is clear which network the machine belongs to. The IP-address is added to the netmask via a logical *AND* thus the "host" part is cut out and gives the network's address as result.
Address of the nameserver	Nameservers provide DNS (DomainNameService) for turning hostnames into IP-addresses. If there is a reachable nameserver on the net and you want to use it by default its IP-address can be entered while configuring the network.

Table 16.1: Values for network configuration

16.2 Configuring using YaST

If all points from section 16.1.2, page 245 are fulfilled you can start configuring your network via YaST.

Select menu item `'System administration'` `'Network configuration'`. YaST first asks you about the hostname and the domainname. Thereafter it wants to know whether only the loopback-device should be used. If this is true all other questions are irrelevant, but this also means that you can't use the machine in a network!

After having entered the IP-address you are asked for a gateway address. If there isn't any gateway installed in your network you should enter your own

IP-address. Next the netmask has to be set. For a class-C network (up to 254 machines in a subnet) this typically is `255.255.255.0`.

After all that you can enter the address of a nameserver, and if it exists enter the appropriate address.

Moreover you can enter a lot of settings directly via our central configuration file; even here YaST supports you (see section 12.6, page 174).

So far for network configuration, really that's all. YaST enters all given settings into the corresponding files (see section 16.3) and quits thereafter. For the changes to take effect you have to reconfigure the changed files and restart the daemons. This can easily be done by:

```
root@earth:/ # /sbin/init.d/network stop

root@earth:/ # /sbin/init.d/network start
```

(see chapter 12, page 169).

16.3 Manual network configuration – where do I find what?

You should use YaST for configuring your network but since (unfortunately) it doesn't cover all parts of networking configuration yet, there might be some work to be done by hand.

Generally all settings should be done in `/etc/rc.config` (see section 12.5, page 173). If you do change this file using YaST you don't have to bother. If you change this file manually don't forget to launch **SuSEconfig** each time you've changed it for the configuration changes to take effect!

16.3.1 Configuration files

This section gives you an overview of the most important configuration files for your network and roughly explains their functions and format.

`/etc/rc.config`
In this central configuration file almost everything concerning your network can be set. After having changed it with YaST or after having launched **SuSEconfig** most of the necessary files are generated automatically. Even boot-scripts are built via settings in this file.

`/etc/hosts`
Here machines (see file contents 16.3.1, page 249) are assigned IP-addresses. If no nameserver is used every machine you want to be able to reach has to be set. There is one line per machine. This line consists of the IP-address, the fully qualified host name (full name), and the unofficial name (nickname). The IP-address has to be at the beginning of the line, entries are separated by blanks, or tab-stops respectively. Comments are started by a '#'.

`/etc/networks`
Here network names are converted to network addresses. The format of this file resembles `hosts` file; here in fact network names are put in front of the addresses (see file contents 16.3.2, page 249):

```
#
# hosts         This file describes a number of hostname-to-address
#               mappings for the TCP/IP subsystem.  It is mostly
#               used at boot time, when no name servers are running.
#               On small systems, this file can be used instead of a
#               "named" name server.  Just add the names, addresses
#               and any aliases to this file...
#
127.0.0.1       localhost
193.141.17.1    gauss.suse.de gauss
193.141.17.42   sofa.suse.de sofa
# End of hosts
```

File contents 16.3.1: `/etc/hosts`

```
#
# networks    This file describes a number of netname-to-address
#             mappings for the TCP/IP subsystem.  It is mostly
#             used at boot time, when no name servers are running.
#
loopback      127.0.0.0
localnet      193.141.17.0
# End of networks.
```

File contents 16.3.2: `/etc/networks`

`/etc/host.conf`

This file is for resolving host names. Here you enter hostnames (or networks); resolving itself is down by the *resolver* library. Comments begin with an '`#`'. The following parameters can be set:

order *bind hosts*	Order in which services for resolving a host name are called. Possible arguments are: *bind*: Using a nameserver *hosts*: Searching `/etc/hosts`
multi *on off*	Determines if a machine in `/etc/hosts` is allowed to have multiple IP-addresses.
nospoof *on*	
alert *on off*	Just influences *spoofing* of the nameserver whithout any other consequences.

Table 16.2: to be continued...

trim <domainname>	The given domain name is cut off from its hostname before being resolved (provided this machinename contains the given domainname). This might come in handy if in `/etc/hosts` there are only local domains which should be recognized even with the full domainname attached.

Table 16.2: Parameter für `/etc/host.conf`

An example for `/etc/host.conf` can be seen under file contents 16.3.3, page 250.

```
#
# /etc/host.conf
#
# We have named running
order hosts bind
# Allow multiple addrs
multi on
# End of host.conf
```

File contents 16.3.3: `/etc/host.conf`

`/etc/resolv.conf`

Same as `/etc/host.conf` this file plays a vital role for resolving machine names via the *resolver*-library.

Here the domain for a given machine is set (keyword **`search`**) and how the address of the nameserver is to be called. There can be several entries for domainnames in here.[1] If a not fully qualified name is resolved entries in **`search`** are appended one after the other to create a fully qualified name.

Several nameservers can be given here, each in a line of its own; those entries have to start with **`nameserver`**. Comments are entered as usual using '`#`'.

YaST (see section 16.2, page 247) enters the given nameserver here!

`/etc/HOSTNAME`

Here the complete name of the machine is given including its domainname (this is called the *fully qualified domain name*. This file is read by a couple of scripts at startup time. It should only contain one line with the machine's name! This file is also automatically generated via settings in `/etc/rc.config`.

[1] The more entries there are the slower resolving of a name will be!

```
# /etc/resolv.conf
#
# Our domain
search suse.de
#
# We use Gauss (193.141.17.1) as nameserver
nameserver 193.141.17.1
# End of resolv.conf
```

File contents 16.3.4: `/etc/resolv.conf`

16.3.2 Startup scripts

Besides those described configuration files there are a couple of scripts that start networking programs at boot-time. These scripts are started as soon as the machine switches to one of the *multiuser-runlevels*.

`/sbin/init.d/network`	This script is responsible for configuring your (network) hard- and software at boot time; it also evaluates IP-address, network address, netmask, and gateway given in `/etc/rc.config` (created by YaST see section 16.2, page 247).
`/sbin/init.d/inetd`	Starts **inetd** if defined in `/etc/rc.config`. This is necessary if you want to login into this machine via network.
`/sbin/init.d/rpc`	Serves for starting the various RPC-servers which are needed (e. g.) if filesystems are exported to other machines via NFS (NFS-server).
`/sbin/init.d/sendmail`	Controls **sendmail** process in dependency of settings in `/etc/rc.config`.

Table 16.3: The network startup scripts

16.4 Routing under S.u.S.E. Linux

16.4.1 Preliminaries

Setting of the routing table under S.u.S.E. Linux is not done via variables in the central configuration file `/etc/rc.config` but via a special script in `/sbin/init.d`. and another configuration file in `/etc`.

After the network has been initialized by the boot-scripts under `/sbin/init.d/ network`, `i4l_hardware` and possibly additional boot-scripts `/etc/route.conf` are searched by `/sbin/init.d/route` routing-table. Then this table is configured for the system.

Any static routes that are needed may be added to `/etc/route.conf`: Routes to a host, routes to a host via a gateway and routes to a network.

Another possibility is to use the program **routed**. Well, this configuration is a bit more tricky. Here we refer to the manpage of **routed**.

16.4.2 How to commence and use it

The rules that apply for `/etc/route.conf` are adapted from the output of **`route`**. If **`route`** is called without parameters the routing table in use is presented. Except for `Flags`, `Metric`, `Ref` und `Use` the entries in `/etc/route.conf` are identical.

Here we give you the rules that apply for `/etc/route.conf`:

- Lines beginning with `#` or blank lines are regarded as comments. An entry consists of one line and 2 up to 4 columns.
- The first column gives the target of a route. Here the IP address of a host, a network, or a *reachable* nameserver may be given. Even the full name is allowed (Fully Qualified Domain Name).
- The keyword `default` marks the default gateway. Please do *not* use `0.0.0.0`.
- The second columns either contains a separator (`0.0.0.0`) or the IP address (or the FQDN) of a host. This host may be the default gateway or a gateway behind a host or network.
- The third column serves for entering the netmask for networks or hosts behind a gateway. For hosts behind a gateway this is `255.255.255.255`.
- The last column only is important for networks (loopback, ethernet, ISDN, PPP, Dummy-device ...) connected to the local machine. Here the device has to be given.

A simple example of `/etc/route.conf` is given in figure 16.4.1. If new entries are added to `/etc/route.conf` entering

```
root@earth:/ > /sbin/init.d/route  stop
```

and thereafter

```
root@earth:/ > /sbin/init.d/route  start
```

sets the routing table with the new entries.

```
# Destination       Dummy/Gateway       Netmask             Device
#
# Net devices
#
127.0.0.0           0.0.0.0             255.255.255.0       lo
204.127.235.0       0.0.0.0             255.255.255.0       eth0
#
# Gateway
#
default             204.127.235.41
#
# Host behind Gateway
#
207.68.156.51       207.68.145.45       255.255.255.255
#
# Net behind a Gateway
#
192.168.0.0         207.68.156.51       255.255.0.0
```

File contents 16.4.1: Simple example of `/etc/route.conf`

16.5 NIS, yellow pages in a LAN

16.5.1 NIS, what it is

As soon as multiple Unix-systems in a network want to access common resources you have to make sure that all users, and groups are identical. The network should be transparent to the user; no matter where he/she is working he will always find exactly the same environment. This is possible via *NIS* and *NFS* services. NFS serves for distributing filesystems over a network and is discussed in section 16.6, page 255.

NIS (Network Information Service) [2] is considered to be a database service which enables access to `/etc/passwd`,`/etc/shadow` and `/etc/groups` over the net. NIS can be used for other, more specialised tasks (e. g. für `/etc/hosts` oder `/etc/services`)which are not mentioned here.

16.5.2 Installing a NIS-client

S.u.S.E. Linux contains all necessary packages for installing an NIS-client. These tools are bundled in package `ypclient`, series `n`. For installing an NIS-client you should proceed as follows:

- Setting the NIS-domain at startup time.

  All you have to do is to set **`YP_DOMAINNAME`** in `/etc/rc.config`. If switching to a (networking) runlevel `/sbin/init.d/network` evaluates those settings and sets the name accordingly.

  NIS-domainname should not be confused with DNS-domainname ; They are not the same even if they do have the same name!

[2] NIS is commonly referred to as YP; this comes from “yellow pages”, the “yellow pages” in the net

- Assigning the NIS-servers. The NIS-server is set via `/etc/rc.config` in the variable **`YP_SERVER`** . **SuSEconfig** now writes the correct values to `/etc/yp.conf` (vgl. file contents 16.5.1, page 254) If you have setup this variable with YaST this step is performed automatically.

  In this file there has to be a line starting with **`ypserver`** followed by the name of the NIS-server. An example of `/etc/yp.conf` can be seen in

```
#
# yp.conf
#
# Legal entries are:
#
# ypserver   <servername>      Define which host to contact
#                              for YP service.
#
ypserver        galois.suse.de
# End of yp.conf
```

File contents 16.5.1: `/etc/yp.conf`

- `helios.cosmos.com` and `heliosII.cosmos.com` have to be set in `/etc/hosts` for being resolved.
- Making sure the RPC-portmapper is started. NIS is realized via RPC (Remote Procedure Calls) therefore starting of the RPC-portmapper is obligatory. This server is started by `/sbin/init.d/rpc`. This as well is done automatically if you have configured it in `/etc/rc.config`.
- Completing entries in `/etc/passwd` and `/etc/groups` .

  For asking the NIS-server about those files after having searched the local ones, a line only containing an `'+'` has to be added. NIS-allows you to set a couple of other informations (e. g. netgroups or local overwriting of NIS-entries. Info about those settings is to be found under the corresponding README file[3]
- Starting **ypbind**. The last step for activating the NIS-server is to launch `ypbind`; this in fact really starts the NIS-client[4]. Even this program is launched automatically if you have configured your network with YaST.
- Activating your changes. Either you restart your system or you enter:

```
root@earth:/ # /sbin/init.d/network stop

root@earth:/ # /sbin/init.d/network start
```

16.5.3 NIS master and slave server

You need to install package `ypserver`, series `n`. How to proceed is given in `/usr/doc/packages/yp/HOWTO`.

[3] `/usr/doc/packages/ypclient/yp-clients-2.2`
[4] Strictly speaking this is no longer necessary but it guarantees the reconnect if, e. g., the NIS-server has been booted

16.6 NFS – distributed filesystems

As mentioned in chapter section 16.5, page 253 NFS serves (together with NIS) for making a network transparent to the user. Via NFS it is possible to distribute filesystems over the network. It doesn't matter at which terminal the user is logged in; he will always find the same environment.

Same as NIS, NFS is an asymmetric service. There are NFS-servers and NFS-clients. Well, a machine can be both of them (it can supply filesystems over the network (export) and mount filesystems from other hosts (import)). Generally there are servers with enormous HD capacity whose filesystems are mounted by other clients.

16.6.1 Importing filesystems

Importing filesystems from an NFS-server is rather simple. The only thing required is that the RPC-portmapper has been started. Starting of this server has already been explained in section 16.5, page 253 in connection with NIS (see page section 16.5.2, page 254). If this is the case other filesystems can be mounted (as long as they are exported by the server) just as easily as local filesystems using the program **mount**. This is the syntax:

mount -t nfs <host>:<remote path> <local path>

So, if user directories from machine `Gauss.suse.de` (e. g.) should be imported this can be achieved via the following command:

```
root@earth:/ # mount -t nfs helios:/home /home
```

16.6.2 Exporting filesystems

A machine that exports filesystems is called an NFS-server . On an NFS-server there are a couple of tools that need to be started:

- RPC-portmapper (*rpc.portmap*)
- RPC-mount-daemon (*rpc.mountd*)
- RPC-NFS-daemon (*rpc.nfsd*)

Those are started by **/sbin/init.d/rpc** at startup time. Starting of the RPC-portmapper has already been discussed in section 16.5, page 253.

Besides starting those daemons it has to be decided which filesystems should be exported. This is done in `/etc/exports` .

One line for each filesystem that is going to be exported is needed; here you can enter which machine may import this filesystem and how (all subdirectories of an exported filesystem are exported as well!). All entitled machines are usually given with their full name (including domainname) but it is possible to use wildcards like '`*`' or '`?`' as well. If no machine is given here any machine is allowed to import this filesystem with the given permissions.

Permissions of the filesystem to be exported are given in brackets after the machine name. The most important options are:

ro	Filesystem is only exported with read-permissions (default).
rw	Filesystem is exported with read-write permissions.
root_squash	This makes sure that the user 'root' of the given machine doesn't have 'root' specific permissions on this filesystem. This is achieved by assigning user-id 65534 to users with user-id 0 (root). This user-id should be set to 'nobody'
no_root_squash	doesn't assign user-id 0 to user-id 65534 (default).
link_relative	Transposing absolute links (those beginning with '/') in a sequence of '../'. This is only useful if the whole filesystem of a machine is mounted (default).
link_absolute	Symbolic links remain untouched.
map_identity	User-ids are exactly the same on client and server. (default)
map-daemon	Client and server don't have matching user-ids. This tells **nfsd** to create a conversion table for user-ids.**ugidd** is required for this to work.

Table 16.4: Permission for exported filesystems

Your exports-file might look like file contents 16.6.1, page 256.

```
#
# /etc/exports
#
/home           helios(rw)   venus(rw)
/usr/X11        helios(ro)   venus(ro)
/usr/lib/texmf  helios(ro)   venus(rw)
/               earth(ro,root_squash)
/home/ftp       (ro)
# End of exports
```

File contents 16.6.1: /etc/exports

File /etc/exports is read by **mountd**. So, if you have changed anything in this file make sure to restart **mountd** and **nfsd** for making your changes take effect. This can easily be done by:

```
root@earth:/ # /sbin/init.d/nfsserver stop
root@earth:/ # /sbin/init.d/nfsserver start
```

16.7 Firewall and Masquerading – Friends and "Friends" on the Internet

This section mainly consists of parts of the "S.u.S.E. Firewall and Masquerading Mini-Howto" from **Bodo Bauer** (if package firewall package is installed; see /usr/doc/packages/firewall/README).

Please note, that for installing a firewall you can't refer to the installation support; (vgl. section 2.1.2, page 10); however we warmly welcome to provide help within the conditions of the business support (vgl. section 2.1.2, page 11).

> There is no warranty that the procedures given here work and do no harm. So, if hacker conquer your system although you did follow the rules, do not blame the authors. We would highly appreciate if you have could send any `feedback@suse.de` instead!

16.7.1 What is a firewall?

A "firewall" serves for protecting a local net from attacks from the internet. This might be achieved by connecting the local net to the internet via a firewall. This configuration is described here. More details can be found in "Firewalling and Proxy Server HOWTO" in /usr/doc/howto/Firewall-HOWTO.gz.

16.7.2 Requirements for installing firewall and masquerading

package `firewall`, series `n1` (network)

You need package `firewall`, series `n` (network) which can be installed by YaST. It contains the following files:

/sbin/init.d/masquerade	Setting up masquerading
/sbin/init.d/firewall	Setting up the firewall
/sbin/redirect	Redirecting packages with this daemon
/sbin/fcii	Redirecting IPs with this daemon
/etc/fw-ssh	List of hosts that have **ssh**-permissions
/etc/fw-friends	List of "friends"
/etc/fw-inout	List of hosts that have access to the Internet
/usr/doc/packages/firewall/*	Documentation on the firewall package

And some links that provide starting up everything at boot time.

Requirements for the kernel

Firewalling and masquerading has to be supported by the kernel. Our S.u.S.E. standard kernels are already configured properly ! Therefore it is not necessary to build a kernel of your own.

Should it however become necessary to build your own kernel, you have to configure the network the right way. In the following, we list the features that have to be enabled. These items refer to kernel configuration using **`make menuconfig`**. For firewalls please also see screen output 16.7.1. The "transparent proxy" may only be activated if the option `'Prompt for` `development and/or incomplete code / drivers'` has been enabled. Here we give you a list of the network options needed:

```
Networking options  --->
  [*] Network firewalls
  [ ] Network aliasing
  [*] TCP/IP networking
  [*] IP: forwarding/gatewaying
  [ ] IP: multicasting
  [*] IP: firewalling
  [ ] IP: firewall packet logging
  [ ] IP: transparent proxy support (EXPERIMENTAL)
  [ ] IP: always defragment
  [*] IP: accounting
  [ ] IP: optimize as router not host
  < > IP: tunneling
  [ ] IP: multicast routing (EXPERIMENTAL)
  < > IP: aliasing support
  --- (it is safe to leave these untouched)
  [ ] IP: PC/TCP compatibility mode
  < > IP: Reverse ARP
  [ ] IP: Disable Path MTU Discovery (normally enabled)
  [*] IP: Drop source routed frames
  [ ] IP: Allow large windows (not recommended if <16Mb of memory)
```

Screen output 16.7.1: Kernel configuration for firewall

If these options are set you can compile your kernel as usual (see chapter 13, page 181).

16.7.3 How a firewall works

The firewall should protect the local net from intruders. This requires at least two networking cards being installed. One is for the local net and the other for the outside traffic. This device can be any networking device such as an ethernet card, an ISDN interface or a SLIP / PPP connection. The only important thing is that it has to be static, not dynamic – the IP address has to be fix. Only the traffic running via the specified device in and going out is checked, the local net remains untouched by the firewall.

Here we give you an example network. The firewall is located between the LAN and the router which enables the connection to the Internet. All examples refer to this example network.

```
 I N T E R N E T
       ^
       |
    Router
       | ...1
       |       Transfernetz (193.141.17.0/30)
       |
       |             WWW/FTP/   Mail/News
       | ...2        DNS-Server und DNS-Server
    Firewall         WWW-Proxy    |
       | ..65            |..66    |..67
   <---+--------+--------+--------+------->lokales Netz
       | ..68   |..69    |..70    |..71   (193.141.17.64/25)
     Host A   Host B   Host C   Host D
```

Only packages that leave the local network via the gateway are controlled. The internal traffic is not controlled by the firewall.

Which variables do I need for firewalls?

The firewall is controlled by a couple of variables. These have the prefix `FW_` and follow the same format. They contain a list of IP addresses or host names separated by blanks. Exceptions are described separately. You have to enter the IPs here *not* the hostnames! During setting up of the firewall there are no nameserver requests as every connection is closed. These variables are set in `/etc/rc.config` (see table 16.5, page 261).

<FW_START>	The firewall is only started if this variable is set to `yes`
<FW_LOCALNETS>	List of local networks. They are protected. only `friends` may access it.
<FW_FTPSERVER>	Adresses of FTP sites that are accessible from the outside.
<FW_WWWSERVER>	Adresses of WWW sites that are accessible from the outside.
<FW_SSLSERVER>	Adresses Secure-Socket WWW sites that are accessible from the outside.
<FW_SSLPORT>	Port where the SSL server expect requests. Here you may only enter one number!
<FW_MAILSERVER>	Adresses of SMTP sites that are accessible from the outside.
<FW_DNSSERVER>	Adresses of DNS sites that are accessible from the outside.
<FW_NNTPSERVER>	Adresses of NNTP sites that are accessible by news feeds.
<FW_NEWSFEED>	Adresses of news feeds that are allowed to conect to the NNTP servers.

Table 16.5: to be continued...

<FW_WORLD_DEV>	Device that should be protected. Here you may enter only one device.
<FW_TCP_LOCKED_PORTS>	TCP portnumbers that should be locked Here you may enter a range that consists of pairs of numbers separated by a colon. Example: `1:6 8:1023` The ports `1` to `6` and `8` to `1023` are locked.
<FW_INT_DEV>	Device for the internal network. Connection to the outside are controlled by this device.
<FW_LOG_DENY>	If this is set to `yes` all violations of the firewall-deny rules are logged to `/var/log/messages`. This means that every try of breaking the firewall is logged.
<FW_LOG_ACCEPT>	If this is set to `yes` all packages that suit the firewall-accept rules are logged to `/var/log/messages`. This means that each packages that passes the firewall (allowed) is logged.
<FW_ROUTER>	Adress of the Internet router. This should only be set if the routers address lies in the range given in <FW_LOCALNETS>.
<FW_INOUT>	If this is set to `yes` `/etc/fw-inout` is read. Otherwise every machine on the local network full access to the Internet.
<FW_TRANS_PROXY_IN>	Here you may enter a list of ports and IP addresses for redirecting packages “OnTheFly” to local ports. (see section 16.7.3). This is for redirecting incoming packages.
<FW_TRANS_PROXY_OUT>	Same as above but for outgoing connections.
<FW_REDIRECT>	This is for redirecting local ports to ports on other machines. This nevertheless is *experimental* and should not be used! (see section 16.7.3).

Table 16.5: to be continued...

<FW_FRIENDS>	If this is set to `yes` the file `/etc/friends` is read. Otherwise no machine on the local net has permissions to connect to the Internet.
<FW_SSH>	This for activating the SSH port (port 22) for those hosts listed in `/etc/fw-ssh`.
<FW_UDP_LOCKED_PORTS>	UDP portnumbers that should be locked Syntax is the same as with the TCP ports. It is recommended to set this top `1:1023` i. e. all reserved ports are locked.

Table 16.5: Variables for the firewall

File `/etc/fw-friends`

Machines that have uncontrolled access to the local net are entered in `/etc/fw-friends`. One line a machine; comments are marked with an ‘#’. This file will only be read if <FW_FRIENDS> is set to `yes`. Otherwise no machine on the local net has access to the Internet.

File `/etc/fw-inout`

Only the hosts listed here have direct access to the Internet. Every machine not included in this list is blocked. Comments are marked with an # (as usual). This file is only read if <FW_INOUT> is set to `yes`. Otherwise any machine of the local network may access the Internet.

Die Datei `/etc/fw-ssh`

Just as above this file contains a list of hostnames (or IP addresses). If <FW_SSH> is set to `yes` in `/etc/rc.config` all listed machines have access to port 22. This means they may access the **secure shell daemon** (**`sshd`**). i. e., sie haben Zugriff auf den **Secure Shell Daemon** (**`sshd`**).

Transparent proxy and IP redirection

This serves for redirecting IP packages. These may be local ports on the firewall machine itself (Transparent Proxy) or ports on other hosts. By setting <FW_TRANSPROXY_IN> incoming IP traffic may be redirected to local ports. It contains a list of quadrupels separated by blanks:

```
Source IP,Target IP,Target Port,Local Port
```

This in particular means that any package that comes from a host with “Source IP” and is aimed for a machine “Target IP” on port “Target port” are redirected to the “Local Port”. <FW_TRANSPROXY_OUT> has the exact same meaning for outgoing traffic. The difference is that `_OUT` on the device given in <FW_INT_DEV> filters whereas `_IN` does the same on <FW_WORLD_DEV>. There may be a daemon on the local port taking care of arriving packages or the may be routed to another host.

That's what <FW_REDIRECT> is needed for. This variable contains a list of triples that resemble the following:

```
Local port,Target IP,Remote port
```

Every package that arrive at "Local port" are redirected to the host given by "Target IP" and "Remote port".

> IP redirection is still in an experimental state and should not be used.

Starting and stopping the firewall

The firewall script accepts four parameters (see table 16.6, page 262).

`start`	The firewall is activated
`stop`	The firewall is shut down; anybody can go anywhere.
`block`	Nothing in nothing out. Resembles unplugging the cable to the internet
`list`	The firewall rules are shown

Table 16.6: Parameters of the firewall script

16.7.4 Example firewall constellation

In section 16.7.3, page 259 you see our example net. There is a local network (`193.141.17.64/25`) as well as a transfer net (`193.141.17.0/30`) which connects the firewall to the router. There are a couple of machines connected to the local network either. Well, only `A` und `B` shoudl be permitted to access the Internet. Machines that are permitted to access the local net are `host.suse.de` as well as `kiste.info.de`. `News.provider.de`. is the news feed machine.

The WWW traffic should be redirected to the local proxy (it listens on port `3128`). Thus we need to setup a transparent proxy which captures every package that goes to the outside via port `80` and sends it to a local port; this port is redirected to port `3128` of the WWW proxy.

Well provided this example network the variables have to be set as given in file contents 16.7.1

<FW_ROUTER> needs to be set for letting machines from the local net accessing the router. If this is not needed you should set this variable to an empty string. The route is not protected by the firewall, so if it is hacked it's rather easy to access the local network.

In `/etc/fw-friends` there should be the following lines: (s. file contents 16.7.2).

Und in `/etc/fw-inout` sind die Hosts `A` und `B` eingetragen (s. file contents 16.7.3).

```
FW_START="yes
FW_LOCALNETS="193.141.17.0/30 193.141.17.64/25"
FW_FTPSERVER="193.141.17.3"
FW_WWWSERVER="193.141.17.3"
FW_SSLSERVER=""
FW_SSLPORT=""
FW_MAILSERVER="193.141.17.4"
FW_DNSSERVER="193.141.17.3"
FW_NNTPSERVER="193.141.17.4"
FW_NEWSFEED="134.222.90.2"
FW_WORLD_DEV="eth1"
FW_INT_DEV="eth0"
FW_LOG_ACCEPT="no"
FW_LOG_DENY="yes"
FW_ROUTER="193.141.17.1"
FW_FRIENDS="yes"
FW_INOUT="yes"
FW_TRANSPROXY_OUT="193.141.17.64/25,0/0,80"
FW_REDIRECT="80,193.141.17.66:3128"
FW_TCP_LOCKED_PORTS="1:1023"
FW_UDP_LOCKED_PORTS="1:1023"
```

File contents 16.7.1: Example configuration in `/etc/rc.config`

```
# /etc/fw-friends
host.suse.de
kiste.info.de
```

File contents 16.7.2: Example configuration for `/etc/fw-friends`

```
# /etc/fw-inout
193.141.17.68           # Host A
193.141.17.69           # Host B
```

File contents 16.7.3: Example configuration for `/etc/fw-inout`

Now you may activate the firewall by entering:

```
root@earth: # /sbin/init.d/firewall start
```

16.7.5 Principle of masquerading

Masquerading means that local addresses are hidden. To the outside there is only one address; the one of your masquerade machine. This serves for hiding the LAN's architecture or for simply assigning internet access to machines without official IP address. Same as with the firewall, there is a device where masquerading takes place.

Kernel configuration for masquerading

Even masquerading needs to be supported by the kernel see screen output 16.7.2.

```
Code maturity level options  --->
  [*] Prompt for development and/or incomplete code/drivers

Networking options  --->
  [ ] Network firewalls
  [ ] Network aliasing
  [*] TCP/IP networking
  [*] IP: forwarding/gatewaying
  [ ] IP: multicasting
  [ ] IP: syn cookies
  [ ] IP: rst cookies
  [ ] IP: firewalling
  [ ] IP: firewall packet logging
  [*] IP: masquerading
  [ ] IP: transparent proxy support (EXPERIMENTAL)
  [*] IP: always defragment
  [*] IP: accounting
  [*] IP: optimize as router not host
  < > IP: tunneling
  --- (it is safe to leave these untouched)
  [ ] IP: PC/TCP compatibility mode
  < > IP: Reverse ARP
  [ ] IP: Disable Path MTU Discovery (normally enabled)
  [*] IP: Drop source routed frames
  [*] IP: Allow large windows (not recommended if <16Mb of
          memory)
```

Screen output 16.7.2: Kernel configuration for masquerading

Which variables do i need for masquerading?

Even masquerading is controlled by a set of variables. The principle is similar to the firewall script, well, there are less variables and they have an `MSQ_` as prefix. These are also set in `/etc/rc.config` (see table 16.7, page 265).

<MSQ_START>	Masquerading is only started if this is set to `yes`
<MSQ_DEV>	Device where masquerading takes place.
<MSQ_NETWORKS>	List of local networks that should be masqueraded.
<MSQ_MODULES>	Modules that are needed for masquerading: There are three modules that may be given: ip_masq_ftp (enable FTP). ip_masq_irc (enable IRC). ip_masq_raudio (enable).

Table 16.7: Variables for masquerading

Special modules

Not all of the protocols accept the replacement of IP addresses in the packages. So, if the server tries to connect to the client this is not possible via a masqueraded router. This is why you may include certain modules for certain protocols (see table 16.8). All modules given in <MSQ_MODULES> are loaded into the kernel during the start of the masquerading.

Modul	Protokoll
`ip_masq_cuseeme`	CU-See-Me
`ip_masq_ftp`	FTP
`ip_masq_irc`	IRC
`ip_masq_quake`	Quake
`ip_masq_raudio`	Real Audio
`ip_masq_vdolive`	VDO Live

Table 16.8: Special modules for masquerading

Starting and stopping masquerading

The masquerade script accepts two parameters (see table 16.9, page 265).

`start`	Activate masquerading
`stop`	Deactivate masquerading
`list`	Display masquerading rules

Table 16.9: The parameters of the masquerading script

Three parameters are known to the masquerade script (see table 16.9, page 265)

16.7.6 Example for masquerading

Here we give an example network. The IP addresses used internally are from a class C network `192.168.100.0`. There is only on official IP address available `193.141.17.1`. FTP as well as IRC should be possible.

```
 I N T E R N E T
       ^
       |
       | 193.141.17.1
    MSQ router
       | ..1        local network (192.168.100.0/24)
    <---+--------+--------+--------+------->
       | ..65   |..66    |..67    |..68
     Host A   Host B   Host C   Host D
```

Now you need to set the masquerading variables in `/etc/rc.config` as given in file contents 16.7.4 zu setzen.

```
MSQ_START="yes"
MSQ_DEV="isdn0"
MSQ_NETWORKS="192.168.0.0/255.255.255.0"
MSQ_MODULES="ip_masq_ftp ip_masq_irc"
```

File contents 16.7.4: Example configuration in `/etc/rc.config`

Now you may activate the masquerading by entering:

```
root@earth: # /sbin/init.d/masquerade start
```

Chapter **17**

Connecting to the world: UUCP, SLIP, and PPP

Apart from the network connections to remote networks (Wide Area Networks, WANs) the use of mailboxes might be of interest.

There are two standards known in the Unix world, UUCP or TCP/IP (via modem or ISDN). While UUCP (**U**nix to **U**nix **C**opy) is mainly designed for transporting mail and news, TCP/IP supplies a *real* network connection which supplies all services from a LAN (which is the basic of TCP/IP).

If TCP/IP is run over a modem most of the time SLIP (**S**erial **L**ine **I**nternet **P**rotocol) or PPP (**P**oint to **P**oint **P**rotocol).

How to connect to a WAN is the topic of the next section. Connection via a modem is explained as well as data transports via UUCP; furthermore we will show you how to configure a SLIP, or PPP connection, respectively and the configuration of a SLIP or PPP server.

17.1 Connecting a modem

Connecting a modem to your machine is exactly how you would expect it to be; it's the same as with other operating systems. The modem is connected to your machine via a serial cable. In YaST you specify which interface to use (see section 12.6, page 174). A link from the modem-device is created to `/dev/modem` thus letting you access your modem no matter where it is connected to.

Normal terminal programs are either **minicom** or **seyon** (of course there are others as well).

17.1.1 minicom

Minicom is a very easy to handle terminal program which resembles the DOS-program **Telix**. This is not an introduction to **minicom** but a short overview of how to configure.

All users that want to use **minicom** have to be entered into `/etc/minicom.users`. Here is given which users can access which modem with what kind of permissions.

You configure minicom (being `'root'`) as follows:

```
root@earth:/ # minicom -s
```

Settings are self-explanatory and don't differ to other operating systems.

17.2 PPP

PPP (Point to Point protocol) enables you to establish a TCP/IP connection via the serial line. PPP offers far more flexibility and options then SLIP. Client and server may communicate while establishing their connection and setup their needed parameters accordingly. The server may configure an IP address as well as a name to the client.

Furthermore PPP is a standard protocol (which SLIP isn't) and is normally the only protocol offered by ISPs.

The **pppd** which is used for communicating with the devices plays a vital and central role in PPP connections. This daemon may either serve as a server or client respectively. The connection itself is done via the program **chat**.

If you plan to setup an "dial on demand" connection you will need the **diald** (dial daemon) as well. This daemon starts a connection as soon as you request services that need a connection to the Internet (such as reading news from another NNTP server). In a nutshell: it establishes a connection as soon as TCP/IP packages need to be send via TCP/IP.

17.2.1 Requirements for using PPP

There are certain things that are required if you want to use PPP on S.u.S.E. Linux:

- The kernel needs to support TCP/IP and PPP. Please make sure if you compile a kernel. (see section 13.4.8, page 195).
- If you want to make use of the **diald** (package `diald`, series `n`) make sure that you have SLIP support compiled into your kernel. (see section 13.4.8, page 195).
- The networking packages need to be installed. The packages that are required are package `nkita` and package `nkitb`, series `a`.
- The PPP base package package `ppp`, series `n`. The **pppd** as well as the **chat** script are included in this package.
- The package `suseppp`, series `n`. This offers you a wide list of pre configured data of some ISPs. This is for automatically setting up PPP with YaST.
- Of course you need to know your login and password at the PPP site.

17.2.2 Customizing PPP

Configuration may be rather easy if the data of your ISP is listed in our configuration scripts. In package `suseppp` there is a list of known scripts (see `/usr/share/suseppp`) which enables YaST to setup PPP according to this ISP. If we don't know the details of your ISP you may start using the default settings (`generic`) and adapt them accordingly.

If you have managed to establish a connection to an unknown ISP (by using the `generic` settings) we would be very grateful if you let us know. This enables us to provide the script in our database making live for future users much easier. You may gather all necessary commands for S.u.S.E. (without the password) by launching the following command:

```
root@earth: # /sbin/conf.d/SuSEconfig.ppp -create-cf
```

see `README` in `/usr/doc/packages/suseppp/`.

> This description assumes that the modem has already been setup as described in section 17.1. It only applies to analogue modems.

YaSTmenu

In YaST please select `'System administration'` as given in section 4.7 and select `'Network configuration'`. First you need to setup the PPP networking device in `'Network base configuration'`. Three steps are needed:

1. Select a device with F5 (`'Modem PPP'`).
2. Configure the device with F6 .
3. Activate the device with F4 .

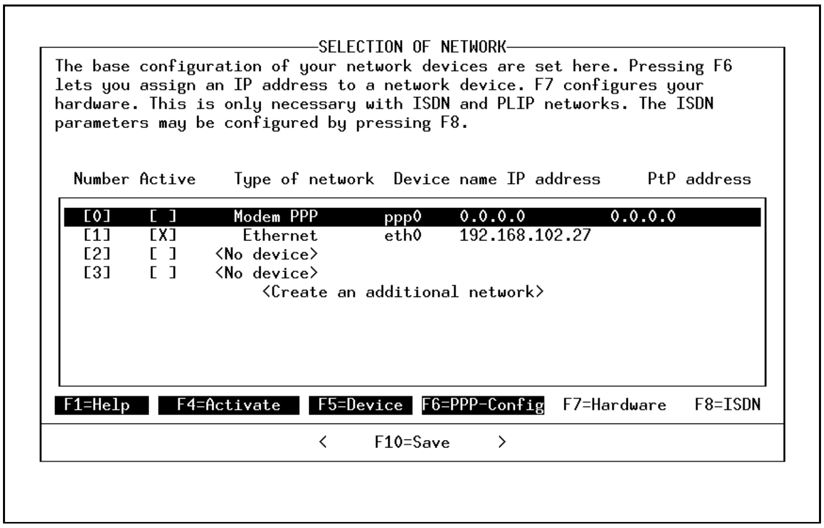


Figure 17.1: YaST: PPP Configuration

If you press F6 you find yourself in `'PPP configuration'`. Here you need to enter your values. You may jump between the entries using ⟼ .

- In **`Provider`** you get a list of the known ISPs. If your provider is not listed please use `generic`. **`Country`** filters only ISPs of this country.
- Enter your login and password in **`PPP Login`** und **`PPP Password`** (you got them from your ISP).
- If you want to connect a standalone host to the Internet via PPP you may activate **`Setup Default route`**

- Activate **`use diald`** (`'X'`) if you like to use the **diald**. This enables you to use your PPP on demand.
- It is recommended to enable **`Debug`** for your first steps and tests (`'X'`).
- Normally you should not change anything in the lower part. If you selected the `generic` template you need to setup this part.
  - **`Telephone number of ISP`**: Enter the telephone number; at the moment you may enter only one number.
  - **`Authentification`**: Here you may select either of `PAP`, `CHAP` or `TERMINAL`. `TERMINAL` is needed if you the remote machine is a "Terminal server". If this is the case you need to adapt `/etc/suseppp/chat-secrets` later on as well.
  - Additional options for **pppd** my be entered in **`PPP options`**. Example:

    ```
    mru=1500 -vj -pc
    ```

    Options as the one given above might be necessary if you have a bad connection. Details are covered in manpage of **`pppd`** (**`man 8 pppd`**).
  - Normally you should set **`Provider nameserver`** as default (`'X'`).
  - The variable **`Provider News server`** should be configured as given by your ISP. If this is set as default (`'X'`) an PPP connection may be established as soon as you invoke a "newsreader" for reading news. The **diald** is required for this to work. If you plan to read news etc. locally you do not have to set this variable.
  - In **`WWW Proxy`** there should be something resembling:

    ```
    wwwproxy.provider.de:<PORT>
    ```

    <PORT> has to be set to the given port.

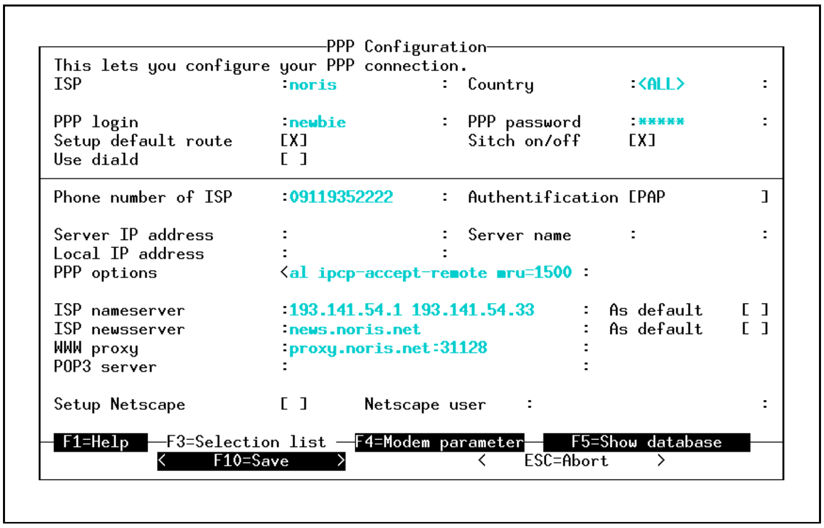


Figure 17.2: YaST: Selection of network (PPP active)

If you leave this menu figure 17.2 by entering F10 a couple of files will be generated in `/etc/suseppp`. These are:
`<provider>.cf`, `<provider>.chat`, `<provider>.options` and `chat-secrets` as well as `chap-secrets` (or `pap-secrets`) in `/etc/ppp`.

`/etc/ppp/options` only is a "Dummy" for making **pppd** happy and satisfied. Now **SuSEconfig** is launched for writing these variables to `/etc/rc.config` (actually this is done by `SuSEconfig.ppp`). There is a documentation on `SuSEconfig.ppp` in `/usr/doc/packages/suseppp`. Anyway every information on package `suseppp` is located in this directory.

Configuration of diald

If the dial on demand doesn't run out of the box the **diald** needs some additional configuration. This is done in `/etc/suseppp/diald/<provider>.diald`. How to setup the **diald** may be seen in manpage of **`diald`** (**`man 8 diald`**).

Here you may (e. g.) set "Timeouts" or the number of tries if a connection fails.

Check the PPP connection

Now you may test your connection by entering:

```
root@earth: # /etc/suseppp/scripts/ppp-up <provider>
```

Of course you have to set <provider> by the name given in **`Provider`**. The error messages are not displayed on screen but written to `/var/log/messages` instead.

by entering

```
root@earth: # tail -f /var/log/messages
```

on a second console you will be able to watch the output. Thus you will see essential warnings and messages.

```
root@earth: # /etc/suseppp/scripts/ppp-down
```

closes the connection. If everything runs smoothly you should disable the **`Debug`** as otherwise `/var/log/messages` will grow rapidly.

17.2.3 Configuration of an PPP server

There is no installation support for setting up an PPP server (see section 2.1.2, page 10). Anyway we offer you this service via our Business-Supports (this nevertheless is not free of charge) (for detailed information please see section 2.1.2, page 11).

Controlling the serial interface on an PPP server is quite comfortable if you use package `mgetty`, series `n` for this task. It is recommended to use **mgetty** even for casual dial ins.

You need to setup one account a PPP client. This can be done by invoking the command **useradd** (more in manpage of **`useradd`** (**`man 8 useradd`**)). The far easier way is to use YaST for this task (detailed information in section 4.7.6).

Invoking **`passwd ppp`** sets a password for the PPP account. The PPP login shell is a small script which in turn launches the PPP daemon:

```
#!/bin/sh
#
# /etc/ppp/ppplogin
#

exec /usr/sbin/pppd modem passive <Local-IP>:<Remote-IP>
```

File contents 17.2.1: `/etc/ppp/ppplogin`

<Local IP> is used for the server whereas <Remote IP> is used for the client.

The PPP daemon is launched as soon as the user `'ppp'` logs in and terminated as soon as he closes the connection.

17.2.4 Further information on PPP

PPP offers far more options and capabilities than we could list here. It would be beyond the scope of this book to mention them all here. If you need more capabilities or options please refer to the corresponding manuals and manpages: `NET-2-HOWTO.gz` and `PPP-HOWTO.gz` in `/usr/doc/howto` as well as in the documentation in `/usr/doc/packages/ppp`.

More detailed information on PPP and its protocols may be found in the corresponding RFCs:

- RFC1144: Jacobson, V. "Compressing TCP/IP headers for low-speed serial links." 1990 February;
- RFC1321: Rivest, R. "The MD5 Message-Digest Algorithm." 1992 April;
- RFC1332: McGregor, G. "PPP Internet Protocol Control Protocol (IPCP)." 1992 May;
- RFC1334: Lloyd, B. Simpson, W.A. PPP authentication protocols."' 1992 October;
- RFC1548: Simpson, W.A. "The Point-to-Point Protocol (PPP)." 1993 December;
- RFC1549: Simpson, W.A. "PPP in HDLC Framing." 1993 December;

17.3 SLIP

SLIP enables you to use TCP/IP via a serial line. This line can be a modem connection to a remote server or a nullmodem as well .

SLIP transports IP-packages via the serial port. These interfaces are configured as *SLIP-devices* (SLIP-devices are named `sl0`, `sl1` and are identified over `ttyS0` and `ttyS1`) For identifying, or for switching them there are two tools; one of them is **dip** the other one **slattach**. We want to present the more comfortable one **dip**. The program **slattach** serves for configuration of SLIP on an existing connection perhaps a permanent line or nullmodem. More information about **slattach** is to be found in [Dawson, 1995].

Before configuring it, some words about requirements:

- The kernel has to support TCP/IP and SLIP (perhaps CSLIP) . To be certain just build a new kernel; this is explained in chapter 13, page 181.
- Networking packages have to be installed; absolutely necessary are package `nkita` and package `netkitb`.
- You have to know your login and password for the SLIP-server
- IP-address of the server and your own IP (for static addressing) have to be known.
- Which SLIP (SLIP or CSLIP) variation should be used? What's your MTU (Maximum Transmission Unit)? In case of doubt you can use an MTU of 296.

17.3.1 Establishing connections with `dip`

Program **dip** (Dial up IP) provides an easy means of establishing and automatizing a connection to a SLIP-server . Program **dip** allows to set serial interfaces to a certain speed, calling a server via modem, doing all login stuff, checking all messages from the server (e. g.for getting to know ones IP-address) and last but not least to switch to the desired SLIP-mode. All those functions can be put together into one script as shown in file contents 17.3.1, page 276.

There are three modes known to dip.

- **`dip -t`** puts you into *command mode*. This is for interactively setting up and configuring a connection. This is mainly designed for testing purposes. More about command mode is to be found in the corresponding manual page or by entering **`help`** in command mode.
- *Dialin-mode* is started with **`dip -i`** or through **diplogin**. This is for building a SLIP-server. More about SLIP-server in section 17.3.2, page 275.
- *Dialout-mode*. A script is executed which describes how to establish a SLIP-connection.

A typical dip-script is shown in file contents 17.3.1, page 276 [1] Establishing a SLIP-connection needs several actions to be taken:

1. Giving the SLIP-server's IP-address. **`get $remote`** <**`Servername`**> sets the server's address . You can either enter the name in quadruples or use the server's name. If you supply the name make sure that it is entered correctly in `/etc/hosts` (see section 16.3.1, page 248).
2. Initializing the serial line **`port ttyS1`** tells **dip** to establish the connection via `/dev/ttyS1`. Here the port where the modem is connected to is given.

   `speed 38400` sets the speed of the interface to 38400 bps.

   `reset` initializes the line. This can cause trouble in some cases. If this happens (i. e.the script hangs) you should omit this command.
3. Initializing of modem and selection of server.

   The command **`send`** sends a string to the in **`port`** given device. You can wait for an input by **`wait`** <Input> <Timeout>". If the expected string

[1] adapting this sample script to your personal needs is described below

(<Input>) is read within the given time (<Timeout>), variable **`errlvl`** is set to the value 0.

The modem is initialized by sending **`ATQ0V1E1S1`** to the interface. If it doesn't answer `OK` within 2 seconds an error is assumed and it jumps to **`error`**.

If initialization was successful the server is selected by **`dial`** <Number>. Command **`dial`** puts <errlvl> to a value according to table 17.1:

`OK`	0
`CONNECT`	1
`ERROR`	2
`BUSY`	3
`NO CARRIER`	4

Table 17.1: Answer of modem ad value of <errlvl>

Thus only value 1 for `errlvl` guarantees a successful connection.

4. Logging in at the SLIP-server

   After connection is up login is to follow. That's why the server is given some time **`sleep`** <seconds> to start the login-program. The modem then sends two ↵ and awaits the prompt `"ogin"`.[2] If this has happened (loginname is send) the password-prompt (`"ord"`) is awaited. Reading of `"ord"` is answered by sending the password.

5. Passing the machine's IP-address, setting of MTU, routing. After both machines are connected, the serial line has to be switched to SLIP-mode. Before this happens MTU and IP-addresses have to be set. For your IP-address there are two possibilities. Either it already exists (static IP) or it is assigned after the connection is up (dynamical IP). Whether your IP-address is static or not depends on the SLIP-server. Small server with small clients normally provide static IPs for their "members" whereas bigger servers search a pool of IPs while the other system is trying to connect.

   - Static allocation of IP-address.

     With static allocation the IP-address is already known. It therefore can be put directly into your dip-script. This is analogue to assigning a server address with **`get $local`** <hostname>. Here the same applies that your hostname has to be entered in `/etc/hosts` or you have to use the IP-address instead of the server's name.

   - Dynamic allocation of IP-address

     If your IP is assigned dynamically the SLIP-server outputs this address on its "welcome" screen after the login. This means that your machine has to "read" the IP-address from the login message.

     dip supports this by providing **`get $locip remote 30`** which reads the local IP-address and sets the variable **`local`** accordingly

[2] it awaits `"ogin"` so that you can type either "Login" or "login"

MTU is set via **`get $mtu`** <**`SIZE`** >. It is important that both sides use the same MTU. You should agree on a certain MTU with your SLIP provider!

If the local machine is not connected to a LAN, you can be sure that all network traffic is done via SLIP you should put this into the kernel's routing table as *default-route*. This is done by: **`default`**.[3]

6. Switching to SLIP/CSLIP mode.

   Last command executed by **dip** is switching the interface to SLIP mode. This happens via **`mode SLIP`**. If the server offers CSLIP instead you have to invoke it as **`mode CSLIP`**.

After having run this script your TCP/IP connection is established. Now you can perform all common tasks within this net (like in a LAN), such as TCP/IP services (e.g. **telnet**, **ftp**, **WWW** . . .)

This script can be used as a template for your scripts. Of course you have to adapt your server-address, interface and modem initialization string, telephone number, login name, password, your MTU and your IP-address as long as you don't have a dynamical one.

> As this file contains unprotected passwords it should not be readable for anybody except '`root`'!

The script is invoked by **`dip`** <scriptname>. If you supply it with the parameter **`-v`** (say **`dip`** -v <scriptname>) **dip** verbosely tells you what it is doing. This can be helpful for debugging.

Shutting down SLIP-connection

A connection established with **dip** can be shut down by **`dip -k`**. The alternative is to send a *SIGINT* to the dip-process.

17.3.2 Configuring a SLIP-server

A SLIP-connection always is established between two machines. Using dip, always one of the two is the server whereas the other is the client. How to configure the client side has been said in the previous chapter, but how to configure a SLIP-server?

Building a SLIP-server can be done in multiple ways. Our mentioned method is rather simplistic and therefore certainly only useful for small servers. Only the base configuration is described, provided that everything else including the network (as far as there is one . . .) is set up correctly. Aspects of experienced network configuration are viewed in brief in chapter section 16.1.1, page 245 and more thoroughly in [Kirch, 1995].

[3] `default` does exactly the same that `route add -net default` <SLIP-Server> does on command line.

```
# sample.dip
#         This script should be an example of how establishing
#         a connection with DIP.

main:

  # Announcing IP-address of the server (Galois.suse.de)
  get $remote galois.suse.de

  # Initializing serial line.
  port ttyS1
  speed 38400
  reset

  # Initialising modem and calling the server.
  send ATQ0V1E1X1\r
  wait OK 2
  if $errlvl != 0 goto error

  dial 49911-3247122
  if $errlvl != 1 goto error

login:

  # Login in SLIP-server.
  sleep 3
  send \r\n\r\n
  wait ogin: 10
  if $errlvl != 0 goto error

  # Send login
  send LOGINNAME\n
  wait ord: 5
  if $errlvl != 0 goto error

  # Send password
  send PASSWORT\n

loggedin:

  # Setting of SLIP-parameters
  get $mtu 296

  # Local machine is hal.suse.de
  # (static allocation)
  get $local hal.suse.de

  # (dynamic allocation)
  # get $locip remote 30

  # Setting SLIP connection as default-route
  default

done:

  # Well, now we can start...
  print CONNECTED to $remote with address $rmtip
  mode CSLIP
  goto exit

error:
  print SLIP to $remote failed.

exit:
```

File contents 17.3.1: Example dip script

Dip as SLIP-server

Our well-known program **dip** can easily be used to configure a SLIP-server as well. We use the *dialin* mode of **dip**. Assigning and adapting clients is done via `/etc/diphosts`.

- Configuring a SLIP-server. For each client a pseudo-user has to be created to login at the server. The only thing special about this user is that it hasn't an interactive *login shell* (as e. g.`/bin/bash` or `/bin/tcsh`) but the program `/usr/sbin/diplogin`.[4] It is important that the appropriate user is allowed to execute `/usr/sbin/diplogin`; and he has to be in the appropriate group ('dialout').

> For serving as SLIP-server **diplogin** needs root permissions. For security reasons **diplogin** of S.u.S.E. Linux*doesn't* have this permission! If you want to design a SLIP-server you have to assign the necessary permissions yourself using the following command:
>
> ```
> root@earth:/root # chmod 4710 /usr/sbin/dip
> ```

A new user (here it is `earth`) can be added with **useradd**:

```
root@earth:/ # useradd -u 6000 -g 300 -c "SLIP by
        earth"
        -d /tmp -s /sbin/diplogin slearth
```

Now your `/etc/passwd` should contain this entry:

`slearth::6000:300:SLIP by hal:/tmp:/sbin/diplogin`

If a password is given to the user (using **`passwd`** <username>) the first step is complete.

- Entering clients into `/etc/diphosts`.

  After having assigned a login for the client a relation between the login-name and the server name has to be set. This is done via `/etc/diphosts`.

  One entry in `/etc/diphosts` consists of one line containing seven fields separated by colons:

  `user:password:remote:local:netmask:comments:prot,MTU`

  Diagram file contents 17.3.2, page 278 shows an entry for `slearth`. Name of the client is `earth`, name of the server is `earth`. Netmask is set to "255.255.255.0" und SLIP is driven at MTU 296.

  For either client or server or both applies that you have to specify either the name or the IP-address.

That's all that's needed for creating a SLIP-server. A client now can communicate with your SLIP-server via TCP/IP.

17.4 UUCP

UUCP has been designed in the late seventies by **Mike Lesk** at **Bell Laboratories** for realizing a simple *Dial-up-network* via telephone lines. **UUCP** works using the *store-and-forward* principle. Messages for another host are

[4] **diplogin** just is a symbolic link to **dip**

user	Login name for the client.
password	This is for assigning another password; generally this is kept empty.
remote host	Name or IP of the client.
local host	Name or IP of the server.(could be empty)
netmask	Netmask for the SLIP device (could be empty.)
comments	Comment.
protocol, MTU	Assigning a protocol (possible values are `SLIP` or `CSLIP`) and MTU).

Table 17.2: Contents of `/etc/diphosts`

```
#
# diphosts      This file assigns Login names to IP-addresses
#               for the dip program.

slerde::erde:galois:255.255.255.0:SLIP by erde:SLIP,296
```

File contents 17.3.2: `/etc/diphosts`

kept until a connection is established. If there is a connection data is transfered and its processing is launched. In the same way data is received and processed. Processing in this context means that mail is delivered and news are sorted, but you can copy any files from one host to the other. Both hosts being directly connected is not required but data is transported via a queue of other hosts before they reach their destination.

17.4.1 Configuring Taylor-UUCP

Along with S.u.S.E. Linux you get the UUCP implementation of **Ian Taylor**, so called **Taylor-UUCP**. The corresponding configuration files are located under `/var/lib/uucp/taylor_config`. Configuration of the UUCP systems is achieved via following files:

`config`:	Central configuration file
`sys`:	Information on systems for communicating
`port`:	Description of available interfaces
`dial`:	Description of available modems

Table 17.3: to be continued...

`call`:	Logins and passwords

Table 17.3: Configuration files Taylor-UUCP

As **Taylor-UUCP** is rather flexible and there are a lot of keywords available it would be beyond the aim of this chapter to explain all of them. Here we just explain the vital options that are necessary for configuring a UUCP-connection.

In our example the name of the local host is `uuearth` which wants to create a UUCP-connection to `helios`.`helios` is regarded the host which delivers `news` and `mails` to `uuearth`.

`config`

This is the central configuration file (see file contents 17.3, page 279). Editing this file can overwrite all hardcoded settings. In general those can be used so that you just have to enter the name of the machine. This is done via the keyword **`nodename`**.

```
#
# config - main UUCP configuration file
#

# UUCP name of the host
nodename        uuvenus
```

Figure 17.3: `/var/lib/uucp/taylor_config/config`

`sys`

File `sys` sets which remote hosts are known to the UUCP-system. Each host description takes its own line which should contain the keyword **`system`** followed by the host name. All settings until the next **`system`** line are regarded as lines for the defined system. Definitions that are made before the separate **`system`** lines are regarded as common entries which apply to all systems, as long as they are not overwritten in the corresponding system line.

Meaning of the used keywords:

`commands`	Allowed commands.
`command-path`	Path where commands are searched.
`call-login`	Setting of login name. Setting `'*'` makes UUCP look in `call` for the appropriate logins.

Table 17.4: to be continued...

call-password	Setting of password. Setting '`*`'" makes UUCP look in `call` for the appropriate passwords.
time	This string is to define when the system (or systems, depending where it is defined) are allowed to be called. This string is an order of substrings which are separated by a vertical bar '`|`' or a comma '`,`'. Each of those substrings has to start with either of '`Su`', '`Mo`', '`Tu`', '`We`', '`Th`', '`Fr`', '`Sa`', '`Wk`' or '`Any`'. The day may be followed by a period of time. This period is given with two times separated by a dash '`-`'. Three examples: **Wk2305-0855,Sa,Su2305-1655** ⇒ working days before 8h55 or after 23h05, saturdays anytime, and sundays before 16h55 or after 23h05. **Wk0955-2205,Su1705-2255** ⇒ working day between 9h55 and 22h05 and sundays between 17h05 and 22h55. **Any** ⇒ No restrictions.
system	Name of the remote system.
phone	Telephone numbers, where the remote system is reached.
port	Port to use for the call. References to an entry in `port` (see below).

Table 17.4: Parameter in `/var/lib/uucp/taylor_config/sys`

An example for `/var/lib/uucp/taylor_config/sys` can be seen at file contents 17.4.1, page 282..

port ¸
This is where the available interfaces (ports) are defined. The structure of this file resembles `sys`. Each interface description starts with with the keyword **port**; global definitions are set before the first **port**- line. Since almost any time there is only one interface available for UUCP, there are only a few entries necessary in `port`. The definitions we used mean:

port	Name of the described port. Is referenced in `sys`!
device	Path to special-device-file; if you have configured your modem with YaST you should enter `/dev/modem`.

Table 17.5: to be continued...

speed	Speed in Bps (Bits per second), for the interface.
dialer	Name of modem that is connected to the interface. Is referenced in `dial`.

Table 17.5: Parameter in `/var/lib/uucp/taylor_config/port`

An example for `/var/lib/uucp/taylor_config/port` can be seen at file contents 17.4.2, page 282.

dial

In `dial` all available modems are defined. Same applies here; all settings before the specific modem settings apply to all entries, which, in this case is the line **dialer** followed by a name.

Besides the name you have to set how your modem is to be initialized, how to call a remote system, which error-codes can occur, and how it should be initialized after a shutdown (or hang-up). This is done via the following keywords:

dialer	Name of modem definition; is referenced in `port`.
chat	Initializing your modem and setting a certain telephone number. Here a couple of strings are given; the first one is sent to the modem while the second one is received by the modem. An empty line is given as "", strings are separated by blanks. Within those strings the following replacements occur: \T Telephone number \r Carriage return \c Suppressing carriage return \d Pausing 1 to 2 seconds \s An empty string
chat-fail	Answer from the modem in case of an error. There can be as many lines as you like.
abort	Modem initialization after connection is established (**complete**), or aborted respectively (**abort**). Syntax of this line is identical to **chat**-line.

Table 17.6: Parameter in `/var/lib/uucp/taylor_config/dial`

An example for `/var/lib/uucp/taylor_config/dial` can be seen at file contents 17.4.3, page 283.

```
#
# sys - Desription of known systems
#

# Globale settings for all systems
commands        rmail rnews
command-path    /usr/lib/news/bin /usr/bin

# Read login name and password from 'call'
call-login      *
call-password   *

#No time restrictions
time            any

# System specific settings

# System 'hal'
system          hal

# Telephone number
phone           0123-123456

# Portdefinition; which port to take
port            serial1
```

File contents 17.4.1: `/var/lib/uucp/taylor_config/sys`

```
#
# port - Description of interfaces
#

# Name of the interface
port     serial1

# Device for this interface
device   /dev/modem

# Speed
speed    38400

# Name of your modem  (references to 'dial')
dialer   generic
```

File contents 17.4.2: `/var/lib/uucp/taylor_config/port`

```
#
# dial - Description of available modems
#
# Name of modem (is referenced in 'port')
dialer generic

#
# Initalization strings
#
chat "" ATZ OK ATDT\T\r\c CONNECT

# Error codes the modem can send
chat-fail       BUSY
chat-fail       NO\sDIALTONE
chat-fail       NO\sCARRIER

# Modemreset after normal shutdown
complete        \d\d+++\d\dATH0Z\r\c

# Modemreset after unexpected shutdown
abort        \d\d+++\d\dATH0Z\r\c
```

File contents 17.4.3: `/var/lib/uucp/taylor_config/dial`

call

Last needed information; this is the name and password for logging into the remote system. This is located in `call`. Each line defines a system and has the following syntax:

<System name> <Login name> <Password>

An example for `/var/lib/uucp/taylor_config/call` can be seen in file contents 17.4.4, page 283.

```
#
# call - Login informations
#

#
# Login name und Password for the given systems
#
# <system> <login> <passwd>
   hal      uusofa  hempel
```

File contents 17.4.4: `/var/lib/uucp/taylor_config/call`

> Since there are unprotected passwords in this file you should ensure that nobody except UUCP has read permissions to this file!

17.4.2 Testing the configuration

For testing your configuration just copy a file to the remote system using the command **uucp** . That's how this is done:

```
root@earth:/ # uucp <file> <system>!<file>
```

(In **bash**, **csh**, or **tcsh** respectively the `'!'` has to be 'masked' with a `'\'`!) Here <file> is to be replaced by the filename to be copied and <system> is the name of the remote system. This system has to be defined in `sys` (**uuname** outputs a list of all machines known to the UUCP-system). **uustat** shows which processes are waiting for being executed:

```
root@earth:/ # uucp testfile erde!/testfile
root@earth:/ # uustat -a
```

halN0002 hal bb 10-24 16:11 Sending /home/user/testfile
(276 bytes) to ~/testfile

Establishing a connection and transferring data is done by **uucico**. The parameter **-S** <system> tells which system is to be called. **uucico** starts a new process which maintains the data transfer.

Parameters **-x** <0-9> are for different debug-levels. Passing `0` means that no debug information is wanted, whereas `9` means that any package transferred is protocolled. Default is `2`. Debug information is written to

`/var/spool/uucp/.Log/uux/<system>` and
`/var/spool/uucp/Log` as well as
`/var/spool/uucp/Stats`

17.5 ISDN Configuration

> At the moment *only* the Euro-ISDN is supported by **isdn4linux**, well, to be precise the passive cards definitely will not work!

Besides its "normal" network capabilities Linux is capable of connecting to your ISP via ISDN. As most of this may be setup with YaST ISDN configuration with S.u.S.E. Linux is rather straight forward and easy.

Our chapter mainly refers to a standard connection to your ISP via ISDN. Of course there is lots more that can be done by **isdn4linux**.

Please keep in mind that some of the procedures mentioned below might be "illegal". Every active cards and their firmware are certified. Passive ISDN controllers are certified if run with the software of the manufacturer. For those who need a certification please use either an active ISDN card or connect the controller to your PBX.

There is one great difference to a modem connection. Once the network has been setup and configured no additional commands are needed. This is called "on demand", as soon as you launch an e. g. telnet session the connection will be established. This normally takes up to three seconds. Thus it is possible to let "normal" users establish a connection. You may set the idle time, this is the period of time when the connections should be cancelled automatically if all processes that depend on the connection are idle.

During configuration of your ISDN system it is recommended to carefully watch messages in `/var/log/messages`. Just start another **xterm** or login onto another console and enter:

```
root@earth: # tail -f /var/log/messages
```

Now you will see every line added to `/var/log/messages`

17.5.1 Overview

S.u.S.E. Linux includes package `isdn4linux`. It includes hardware drivers, network interfaces as well as modem emulation (only digital modems). Furthermore it includes software for e. g. an answer phone.

The ISDN hardware driver is launched by `/sbin/init.d/i4l_hardware` (see chapter 12). Configuration of the ISDN part is done via **isdnctrl** (manpage of **isdnctrl** (**man isdnctrl**)). The network interfaces are configured just as standard ethernet interfaces by **ifconfig** (manpage of **ifconfig** (**man ifconfig**)) and **route** (manpage of **route** (**man route**)). On S.u.S.E. Linux `/sbin/init.d/i4l` performs this task (see chapter 12).

The actions taken are based upon entries in `/etc/rc.config`. These names resemble (where possible) the options of **isdnctrl**.

`/sbin/init.d/route` sets up the routing of the devices given in `/etc/route.conf`.

Establishing an connection is done by **isdnctrl**, or bzw. `/sbin/init.d/i4l` respectively using settings from `/etc/rc.config`. These parameters may be listed by entering:

```
root@earth: # isdnctrl list all
```

as soon as somebody requests an ISDN service (this may be either a user or an application) the connection is established.

17.5.2 Configure ISDN hardware

Requirements

For successfully creating a connection on S.u.S.E. Linux you will need:

1. an ISDN connection
2. a supported ISDN controller
3. S.u.S.E. Linux installed
4. one of S.u.S.E. Linux standard kernels (on CD).

> You *don't* need to compile a kernel! If you want to compile a kernel anyway make sure you use the sources from package `lx_suse`, series `d`!

5. the package `kernmod`, series `a`
6. the package `i4l`, series `n`
7. documentation may be found in package `i4ldoc`, series `doc` (recommended).

What you need to know:

- Type of your ISDN controller
- Controller settings: IRQ, Port address, etc. (depends on the type)
- The ISDN protocol you use:
  - **1TR6**: (old) national ISDN
  - **DSS1**: Euro-ISDN

> Some of the PBX (as opposed in the documentation) still use **1TR6** instead of **DSS1**

What is a MSN/EAZ?

With Euro ISDN you get a MSN (Multiple-Subscribe-Number) which normally is your telephone number without the prefix. If you just subscribed for ISDN you will receive three different numbers. Any of them may be used for your ISDN connection even if you use the same number as telephone number as they may be distinguished by their service indicator.

Normally the ISDN controller is directly attached to an NTBA. You might as well connect another S0 bus to the PBX. If you use Euro ISDN on your PBX the MSN normally is the extension (direct call number).

For 1TR6 you have an EAZ (German: "Endgeräte Auswahl Ziffer = end user selection number") otherwise they are treated the same. The EAZ is a single number. Just select one in the range of `1` to `7`. Don't forget the `0`!

Configure ISDN hardware with YaST

The driver itself is provided by a loadable kernel module. You don't need to reboot your system. Standard ISDN controllers are supported by the **HiSax** driver.

Some controllers e. g. **ICN** or **AVM-B1** (as well as PnP cards) may not be configured with YaST yet. They need a special treatment. Please have a look at the settings for ISDN controllers further down.

Here is how to proceed step by step:

1. Login as user `'root'`
2. launch YaST
3. Now select `'System administration'`, `'Integrate hardware into system'`, `'Configure ISDN hardware'`. The menu structure is shown in figure 17.4.
4. Please enter the following parameters:
   - **`Start I4L`**
     ISDN is only launched if this is activated. Thus you can make sure whether ISDN is started automatically at boot up.

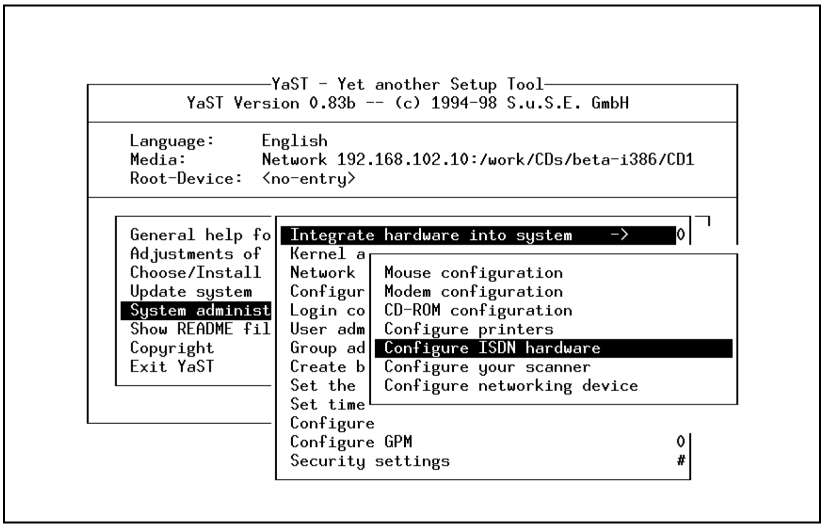


Figure 17.4: Menu structure for ISDN configuration in YaST

- **`ISDN protocol`**
  Here you may choose either the old (national) German ISDN (1TR6) or the default Euro ISDN (EDSS1). Please keep in mind that connections via a PBX often still use 1TR6.
- **`ISDN controller type`**
  Select the supported ISDN controller. Please have a look at `/usr/doc/packages/i4l/README.SuSE`. for PnP and/or PCMCIA controllers.
- **`Controller ID`**
  You should leave this untouched to `Tel0`.
- **`Interrupt`**
  `Memory base address`
  `IO port`
  `ISAC`
  `HSCX`
  Dependent on the card in use some additional settings might be necessary. Only the parameters available for the device are enabled the others are disabled.
- **`ISDN options`**

  This should be empty!

Pressing F1 will give you additional help. figure 17.5 shows you the menu mask.

5. Now please confirm by pressing `'Start'`.
   This is a test: The module will be loaded and the messages in the window tell you whether the card has been setup correctly.

```
                    CONFIGURATION OF ISDN HARDWARE
Configure your ISDN card here (only HiSax driver). Enter the parameters and
test the configuration by pressing 'Start'. If it works, save the
configuration  by pressing 'Save'. For help, press F1!


Start I4L                               [X]

ISDN protocol                           [Euro ISDN (EDSS1)               ]

Type of ISDN card                       [Teledat 150                     ]

Card identification                     :HiSax    :
Interrupt                               :5        :
Memory base address          (Hex)    0x:d2000    :
IO port                      (Hex)    0x:300      :
Value for ISAC           (only PNP)     :         :
Value for HSCX           (only PNP)     :         :

ISDN module options (only needed in special cases!!)
:                                                                        :

<      Save      > <     Start      > <ISDN parameter>  <     Abort      >
```

Figure 17.5: Menu for ISDN configuration with YaST

If OK: Confirm by pressing `'Save'`.

Now your settings are written to `/etc/rc.config` permanently. They will remain active unless you change them. After being tested the driver remains loaded.

If it fails: Check and change the parameters and don't forget to have a look at `/var/log/messages` (you remember you opened it, didn't you?.:)).

Possible problems may be:

- On some boards the IRQs 12 or 15 cannot be used.
- The given address or IRQ are already in use. Remove (just for test) all controllers that are not urgently needed for checking (e. g. sound or networking devices).
- The module has already been loaded. To remove it please change to another console and enter:

  ```
  root@earth: # rmmod hisax
  ```

- The card you use is an PnP device please have a look at `/usr/doc/packages/i4l/README.SuSE`.
- Your card is not supported by **HiSax** (e. g. ICN, AVM-B1). Please refer to `/usr/doc/packages/i4l/README.SuSE`.

6. Exit YaST.
7. configure **isdnlog**

   You should configure **`isdnlog`** before launching the modules. Its task is it to supervise every activity on the S0 bus system.

   Please adapt the following files to your needs:

   - `/etc/isdn/isdn.conf`:

The first parameter is for specifying your country where you use your `isdn4linux`. If this is Germany you should set them as given in file contents 17.5.1, page 289:

```
# /etc/isdn/isdn.conf

[GLOBAL]
COUNTRYPREFIX = +
COUNTRYCODE = 49
AREAPREFIX = 0
```

File contents 17.5.1: `/etc/isdn/isdn.conf`

Here (in the `GLOBAL` section) you also need to enter your area code `AREACODE` (the dialing prefix) without the leading zero. So if your area code is `0911` e. g. you will need to enter: `AREACODE = 911`.

In Germany this is the only part that needs to be adapted. `CHARGEMAX = 20.00` lets you set the maximum amount of money (in Deutsch Marks) that you accept a day. Please do not rely on this feature!

- `/etc/isdn/callerid.conf`:
  Here you may enter every known telephone number. You will see their "names" instead of the MSNs in `/var/log/messages` and if you launch **isdnrep**.
  See our example in file contents 17.5.2, page 289; Your number is `4711`, the number of your ISP is `4712`.

```
# /etc/isdn/callerid.conf

[MSN]
NUMBER = 4711
SI = 1
ALIAS = myself
ZONE = 1

[MSN]
NUMBER = 4712
SI = 1
ALIAS = ISP
ZONE = 1
```

File contents 17.5.2: `/etc/isdn/callerid.conf`

- `/etc/isdn/isdnlog.isdnctrl0.options`: Here you may enter options for **isdnlog** this normally is not necessary.

8. Now enter the commands:

```
root@earth: # init 1
root@earth: # init 2
```

thus restarting all network services. You may as well activate ISDN with YaST or reboot if you prefer.

17.5.3 Testing ISDN on our S.u.S.E. host

The S.u.S.E.ISDN server

This server has been setup for S.u.S.E. Linux users to check your ISDN configuration by logging in using test logins. In the near future we plan to offer more capabilities, this being access on our Support Database (update packages etc). Please have a look at `/home/suse/README` on the the "test" server.

> You will not be able to access the Internet via this server!

Selection of a protocol

This ISDN host offers three different modes:

- Terminal-Login using **X.75**
- **rawip-HDLC**
- **syncPPP**

Of course you may check any of these protocols. Anyway it is recommended to check the protocol offered by your ISP.

Requirements

The ISDN hardware configuration ran smoothly and the ISDN driver has been loaded. Furthermore you need to have your MSN/EAZ at your hand.

Select a protocol (**rawip**, **syncPPP**). Of course you may setup different connections. Our example refers to an **syncPPP** connection. **rawip** is almost the same but much easier.

Well, here are the steps:

1. Please invoke YaST and change to `'System administration'`, `'Network configuration'`, `'Basic network configuration'`. figure 17.6 shows you a screen shot.
2. Now use a number not already used e. g. `4`.
3. Now select the device `'ISDN SyncPPP'` by pressing F5 .
4. Hit F6 (`'IP address'`) and enter ein:
   - IP address of your host: `192.168.0.99`
   - IP address of the PointToPoint partner: `192.168.0.1`
5. Leave this window by pressing `'Continue'`.
6. Hitting F4 activates the network device if you haven't done so already.
7. F8 (`'ISDN'`)lets you enter some ISDN specific options. This is shown in: figure 17.7 .

   Now please enter the following values:

   - Now enter **`Your telephone number (MSN)`**

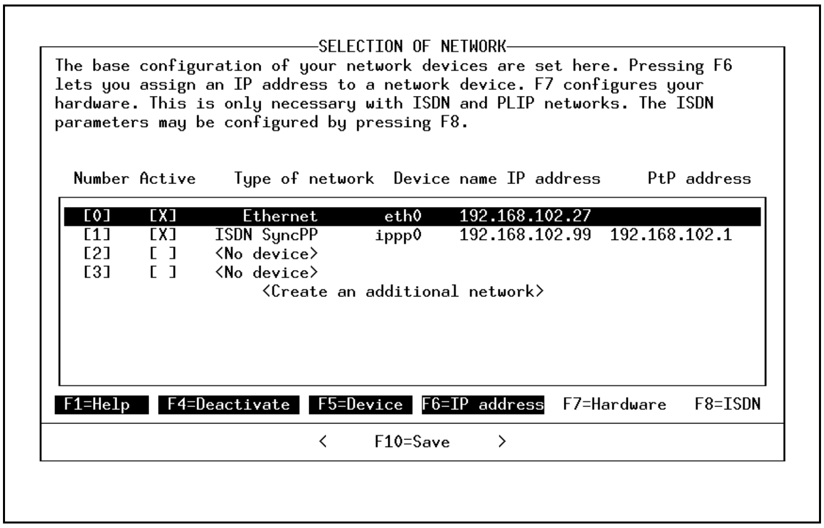


Figure 17.6: Network configuration with YaST

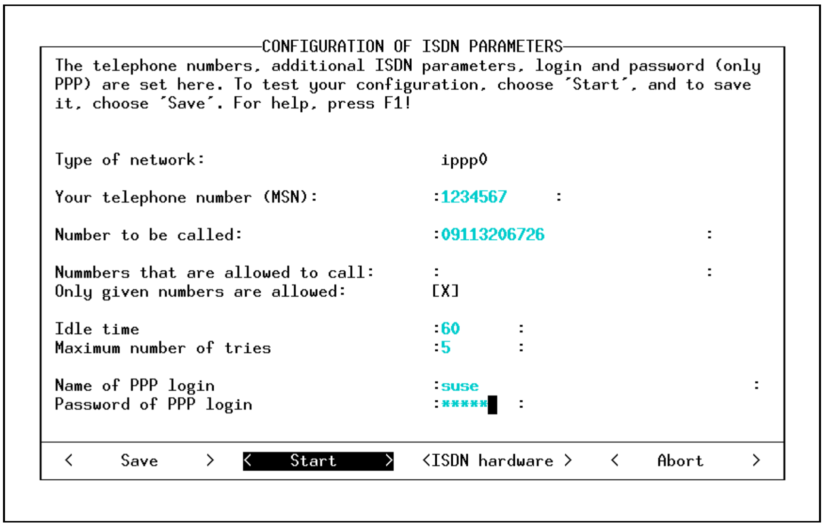


Figure 17.7: Setting of ISDN network parameters with YaST

- **`The number to be dialled:`** `09113206726`
  `09113206726` is the number of the S.u.S.E. test host for **syncPPP**.

> You might need to dial a leading "0" for some PBXs.

- **`Numbers, that are allowed to dialin:`**
  Only needed for dialin servers
- **`Only given numbers are allowed:`**
  Please set this for making sure nobody starts and unwanted connection to your machine!

- **Idle time:**

  After a period of time (idle time) the connection is closed if it is currently not in use:

- **Name of PPP login:**

  For our S.u.S.E. syncPPP test the login name is `suse`.

- **Password of PPP login:**

  For our S.u.S.E. syncPPP test server the password is `linux`. You will not see the actual password while entering it (bus asterisks instead). It is written to `/etc/ppp/pap-secrets`.

Hit F1 to get additional help.

8. Confirm `Start`.

   Now a network test is launched. You will get an error message if anything went wrong. There should not be problems so far.

   If OK: Please confirm by hitting `Save`.

   Now your settings will be written permanently to `/etc/rc.config`. They will remain active unless you change the settings. The settings remain active after the "test" start.

   If it fails: Eventually the ISDN modules were not loaded. Don't forget to have a look at `/var/log/messages`.

9. Now switch to another **xterm** or another console. You don't need to be user `root` to enter the following:

10. Establish an Internet with **syncPPP** mit:

```
root@earth: # telnet 192.168.0.1
```

or if you use **rawip**

```
root@earth: # telnet 192.168.0.2
```

After approximately 2-3 seconds you should see the S.u.S.E. server login prompt. Now you may enter with your account and password (`suse`, `linux`). Thereafter have a look at `README`. e. g. by entering

```
less README
```

`logout` lets you close the connection. After the idle time that was setup previously the connection will be dropped anyway. Every activity may be watched in `/var/log/messages`.

If anything fails:

- Check `/var/log/messages` for "suspicious" messages
- Try the **rawip** connection
- Is the MSN/EAZ setup correctly?
- Perhaps you need a leading `0`?

There are further hints in our Support Database. It may be found on our WWW site at `http://www.suse.de/Support/sdb` or by launching the S.u.S.E. help system (enter: **`hilfe`** or select it from the menu; package `susehilf`, series `doc` and package `sdb`, series `doc` needs to be installed)

11. Presumably the connection was established. Switch to YaST select `'Save'` and exit YaST.
12. Now please adopt `/etc/hosts`. All IP addresses used should be listed here (see file contents 17.5.3, page 293).

```
192.168.0.1   pppserver.suse.de   pppserver
192.168.0.99  pppclient.suse.de   pppclient
192.168.0.2   rawipserver.suse.de rawipserver
192.168.0.98  rawipclient.suse.de rawipclient
```

File contents 17.5.3: Entries in `/etc/hosts` for connecting to the S.u.S.E. ISDN test server

17.5.4 Setup configuration for your ISP

If you have successfully managed the "test" connection you may connect to your ISP in exact the same manner. This might be tricky but only if the ISP uses protocols and switches that are not usual. Some instructions and vital hints may be found along with some scripts on our SDB (Support Database) (see section 2.1.3, page 12).

The method mentioned above connects using either rawip or synchronous PP using PAP as authentification method. This normally is "standard".

Use the settings for the test server to start with and change the parameters listed below:

- ISP telephone number
- User name and password (for syncPPP)
- IP numbers (if known, please see below)
- Enter a nameserver This can be done interactively from YaST in `'System administration'`, `'Configure network'`, `'Name server'`
- Setup your routing section 16.4

Howto use dynamic IP numbers with syncPPP

Your ISP only provides dynamic IP addresses; what do I need to setup?

If you don't have an IP (dynamic IP only for PPP) you should add a private address for yourself and your ISP (`192.168.*.*`). Thus you may just leave the entries as given in our example. After the connection has been established invoking **`ifconfig`** or having a look at `/var/log/messages` will give you the IP addresses used.

Your IP address is not important at all. The IP address of your PointToPoint partner should be a fixed one. Now you may enter this address and thus adjust your gateway.

Each time a connection is established the **ipppd** launches an **`ifconfig`** (man-page of **`ifconfig`** (**`man ifconfig`**)). All routing information goes into the bin as soon as you restart **`ifconfig`** for this device. So if you like to have your "default-route" after the connection is down you need to set it in `/etc/ppp/ip-down`.

There is an example of this file in `/usr/doc/packages/i4l/pppsample`.

17.5.5 Reference of the S.u.S.E. ISDN account

telnet-Login	
User name	'suse'
Password	'linux'

rawip	
Telephone number	09113206728
Client IP address	192.168.0.98
Server IP address	192.168.0.2
Network device	isdn0

syncPPP	
Telephone number	09113206726
Client IP address	192.168.0.99
Server IP address	192.168.0.1
Network device	ippp0
User name	'suse'
Password	'linux'
Authentification	PAP

Modem	
Telephone number	09113247114

Initializing the modem:

```
ATZ
ATS14=0&E123
ATD09113247114
```

In our example `123` is your MSN.

Additional information

Further information on how to establish a connection to the S.u.S.E. ISDN server may be found in:

- File `/usr/doc/packages/i4l/README.SuSE`
- Support Database: `http://www.suse.de/Support/sdb`
- In package `i4ldoc` (e. g. the ISDN-FAQ in: `/usr/doc/packages/i4ldoc/i4l-faq`)
- `/usr/doc/inetcfg` (package `inetcfg`): e. g.: T-Online via ISDN

17.6 Let's write – configuration of email

If a connection to the wide world has been established, let it be it via UUCP, SLIP, or ISDN, it should be used to do something. A rather typical application is ☞*electronic mail*, electronic mail. This section describes configuration of **sendmail**[5] .

sendmail has to decide how to deliver incoming (and outgoing) electronic mail. This might either be via a TCP/IP network using the SMTP protocol or to use another transport agent such as UUCP.

sendmail's main configuration file is called `/etc/sendmail.cf`. If you have a normal and simple configuration you may setup the necessary parameters using YaST. YaST then creates a valid `/etc/sendmail.cf` for you. All settings are written to `/etc/rc.config` and YaST will create the file `/etc/sendmail.cf` for you using the parameters given there.

As **sendmail** configuration is rather complex S.u.S.E. provides two preconfigured settings that should be sufficient in most cases:

If you plan to use **sendmail** within a TCP/IP network make sure to have a valid DNS server. Here you need to setup one entry ("MX record" mail exchange record) per name. The current settings may be checked with the **host** command (in package package `bind`):

```
root@earth: # host helios.cosmos.com
helios.cosmos.com address 192.168.0.1
helios.cosmos.com mail is handled (pri=10) by helios.cosmos.com
helios.cosmos.com mail is handled (pri=100) by mail-relay.cosmos.com
```

If there is no entry for mail you should ask your system administrator for help.

The following electronic mail variables may be set by YaST in `/etc/rc.config` (see section 12.6, page 174):

- **`SENDMAIL_TYPE="yes"`**
  This variable has to be set to `yes` if the **sendmail** configuration file shall be created using the values in `/etc/rc.config`. If you want to create a `/etc/sendmail.cf` yourself `no` is the answer.
- **`SENDMAIL_LOCALHOST="localhost helios.cosmos.com www.cosmos.com"`**
  sendmail needs to know which electronic mail should be stored locally and which needs to be delivered to another host. Only electronic mail to the local host itself is saved locally by default. By givng a list in **`SENDMAIL_LOCALHOST`** you can configure other names that should be considered a local host.

  Example: the name of the machine is `helios.cosmos.com` it serves as WWW site for www.`cosmos.com`. For accepting electronic mail that goes to www.`cosmos.com` you need to enter the following line: **`SENDMAIL_LOCALHOST="localhost www.cosmos.com"`**.

[5] There is an alternative to **sendmail** called **smail** or **qmail**, well these are not covered in this chapter.

- **FROM_HEADER=cosmos.com**
  Normally the local machine's name is used as the from header. This may be set to an indifferent name:
  Example: The machine's name is `earth.cosmos.com`. Anyway you'd like to send electronic mail as `newbie@cosmos.com`. This may be achieved by the parameter:
  FROM_HEADER=cosmos.com
- **SENDMAIL_SMARTHOST=mail-server.provider.de**

  sendmail asks for the DNS names of every mail that is not delivered locally and tries to send the electronic mail via the SMTP protocol. This host might be anywhere on the Internet and eventually has a rather slow connection to our local host. Setting this parameter lets you setup an intermediate host that gets all our outgoing mail. Then this host is responsible of delivering your electronic mail.

  First example: This is for a dialup connection. Thus you deliver all of your electronic mail directly to your ISP:

  SENDMAIL_SMARTHOST=smtp:mail-server.provider.de.
  Second example:
  If you are connected via UUCP you may send all electronic mail that is not local to your UUCP server:
  SENDMAIL_SMARTHOST=uucp-dom:uucp.cosmos.com
- **SENDMAIL_NOCANONIFY=no**
  sendmail tries to resolve each and every email address from the mail header and replaces the name with the "Fully Qualified Domain Names" (FQDN). So if there is no DNS server available (due to a dialup connection perhaps) and you enter the name correctly you may switch this of by setting it to `yes`.
- **SENDMAIL_ARGS="-bd -q30m -om"**

  That's how **sendmail** is invoket at bootup time. `-q30m` tells **sendmail** to check `/var/mqueue` every 30 minutes if there is electronic mail waiting. `-bd` starts **sendmail** in "daemon mode" this enables you to accept electronic mail via the TCP/IP network. So if you only have a dialup connection you may omit the `-q30m` and invoke **sendmail** directly with **sendmail -q**. This might be done via a **crontag** entry once or twice a day. Additionally you may enter the start of **sendmail -q** into your scripts that establish the connection to your ISP. This lets you exchange electronic mail every time you connect to the net.
- **SENDMAIL_EXPENSIVE=no**

  Normally **sendmail** tries to deliver the electronic mail immediately via SMTP. So if you are only temporarily connected this might not be what you need as a connection is established each time you write an electronic mail. If setting this to `yes` the mail is queued in `/var/mqueue` and delivered as soon as you launch a **sendmail -q**.

All locally delivered electronic mail is handled and saved to the local electronic mail folder `/var/spool/mail/<name>` by **procmail**. Please have a look at manpage of **`procmailrc`** (**`man procmailrc`**), manpage of **`procmailex`** (**`man procmailex`**) as well as manpage of **`procmail`** (**`man procmail`**) for a description of this extreme versatile tool.

If you do not deliver remote electronic mail immediately it is saved to the queue directory `/var/mqueue` and delivered at the next run of **sendmail**. You may as well launch **sendmail** directly by entering **`sendmail -q`**.

There are further settings that can be made, e. g. in `/etc/aliases` and some other files in `/etc/mail/`. There are commented examples included. Some files need to be translated to databases using the **makemap** tool. This is invoked automatically if you start **SuSEconfig** or when you leave YaST.

If you need a more complex configuration of **sendmail** you should disable the automatic setup of `/etc/sendmail.cf` by setting **`SENDMAIL_TYPE=no`**. Then you may use `/etc/mail/linux.mc` as a template for your configuration. `linux.mc` is written using **m4** commands.

```
root@earth: # m4 /etc/mail/linux.mc > /etc/sendmail.cf
```

creates a valid **sendmail** configuration by using the macros in `/usr/share/sendmail`

Further documentation my be found in `/etc/mail`, `/usr/share/sendmail` as well as `/usr/doc/packages/sendmail` There is a web site at `http://www.sendmail.org/`. If you need to solve more complex tasks you will need the sendmail boot from O'Reilly[6] This describes **sendmail** in any gory detail.

17.7 Brandnew messages – C News

One of the most important services that is provided by the internet is the transport and delivery of news that are sorted in different groups; this part of the internet is often referred to as the usenet. Only through this media development it was possible to develop Linux at all, and only by means of this high-efficient form of communication a rapid development and removing of bugs was and is possible.

Since a complete description of the news-system with all its thousands of possibilities (e. g. passing news to other machines) would be far beyond the scope of this book; only a basic local system based on the widely spread **C News**-packages is described.

17.7.1 Broadcasting the news

The machine that wants to receive news generally is not on-line all the time (due to telephone costs which are immense for the private user). Usually UUCP (chapter section 17.4, page 277) serves for exchanging news over the net. Those machines that deliver news are called newsfeed. On those machines there is a program running several times a day which packs all news that have to be transferred to other machines and makes them ready to

[6] Vgl. [Costales et al., 1993].

send via UUCP. Those machines that want to receive news have to poll the newsfeed; they call it up once or several times a day and let those prepared packages be sent.

Since this is a regular job it is quite obvious to let **cron** handle this task; thus you don't have to start the job by hand each time you want to poll the news. This can be done by editing the cron table via **`crontab -e`** being `'root'`; you should enter the following line:

```
7 05 * * *       /usr/lib/uucp/uucico -f -s erde
```

File contents 17.7.1: Example cron entry

A machine with this entry in crontab would call the newsfeed `earth` every day at 5h07 for polling news. Of course it is required that **UUCP** is configured correctly and the machine is running at the given time.

17.7.2 Installing a local news-system

For processing those received files it is necessary that package `cnews` is installed properly.

Since permissions are very important for cnews to work properly all configuration should be done as user `' news'`. This can be achieved if you login as `'root'` and change to news by **`su - news`**.

Before starting the configuration you should take the amount of news to be received into consideration. News are saved under `/var/spool/news`; since those files are relatively small you could consider giving it a partition with a small ☞*inode density* (app. 2048 bytes per inode). If the amount is relatively small this is not necessary. But you should ensure that there is enough space for your news.

Now you have to create directories for all these newsgroups; for maintaining news there are several tools available under `/usr/lib/news/bin`. In directory `maint` there is the script **addgroup**; it is designed for adding a newsgroup. Invoking it like

`addgroup de.comp.os.linux.x y`
will create the following hierarchy under `/var/spool/news`: `de/comp/os/linux/x`.

Herein all future news belonging to that special group are sorted.

> Remember to become user `'news'` before adding newsgroups! Otherwise news will not have write permissions for these directories!

After having assigned all relevant groups you have to make sure that those packages received by UUCP are processed correctly and the articles are put into their directories. Typically this also is done via a cron-job. Your cron-tab, which can be seen by typing **`crontab -l`**, for `'news'` should resemble file contents 17.7.2, page 299.

```
#
# Example crontab for the C news Cleanup Release
#
# Sort new news once every 15 minutes
#
0,15,30,45 * * * * touch /tmp/newsrun; /usr/lib/news/bin/input/newsrun
#
# Prepare new news once an hour
#
40 *  * * *   /usr/lib/news/bin/batch/sendbatches
#
# Delete old news once a day
#
59 0  * * *   /usr/lib/news/bin/expire/doexpire
#
# Run newsdaily once a day at 08:10
#
10 8  * * *   /usr/lib/news/bin/maint/newsdaily
#
# Run newswatch once a day at 13:00
#
00 13 * * *   /usr/lib/news/bin/maint/newswatch | mail usenet
```

File contents 17.7.2: A news `crontab file`

The first entry sets that all newly arrived news-packages are sorted every 15 minutes. Of course this can be changed if there is only one user who polls only once a day. In this case it is most recommended to launch **newsrun** directly after polling.

sendbatches is responsible for the local news to be packed correctly and to be passed to the newsfeed at next polling. In our given example file contents 17.7.2 this would be done once an hour; even here you can do this only once a day just before polling.

Since all those articles get older and thus occupy useless disk pace on your system those articles that have reached a certain "age" have to be expired; that's **doexpire**'s task whose actions are set via `/var/lib/news/explist` (see manpage of **`expire`** (**`man expire`**)).

newsdaily and **newswatch** solely serve for supervision purposes. As almost all news programs run in background those tools make sure that nothing happens; if something happens *NEWSMASTER* is informed. If something worth mentioning to the newsmaster is found a report is sent via email to '`news`', '`newsmaster`', or to '`usenet`'. There should be alias entries for those pseudo-users in your mail-system for those mails to reach the user in question (see manpage of **`newalias`** (**`man newalias`**)).

17.7.3 Overview of configuration files

In table 17.7, page 300 we give a short overview of those files used by **cnews**; those files are located under the home directory of the pseudo-user '`news`', this normally is `/var/lib/news`. All files contain useful defaults and should only be changed if you really know what you are doing.

All changes to those files have to be done as user `'news'`!

active	Contains names of local newsgroups and information about existing articles. This file is kept on the latest update all the time and is read among others by the newsreaders for finding out which newsgroups are available.
batchlog	Log file for packing local news. More generations of this file are kept. An `.o` extension is added to older versions.
batchparms	Determines how locally written news are packed and prepared for sending.
crontab	Example crontab table for user `'news'`; this table can easily be installed via **`crontab`**.
errlog	Here all error-messages are logged. Older versions are kept a while before being deleted.
explist	Determines the behaviour or **expire**. Here you can specify which articles are to be deleted after how many days, whether articles should be archived and more.
log	Here sorting of all articles is logged.
newsgroups	List of existing newsgroups including description.
organization	Should contain the name of your organization (e. g.. your company's name). This field appears in every article wrote on this machine.
whoami	Name of the local machine

Table 17.7: **CNews** files

17.7.4 Reading news

For reading news there are a couple of tools available (e. g. **nn**, **tin**, or **trn**). Choice of a newsreader is, same as with editors, a personal decision. All those newsreaders can be configured to use a nameserver or to access local spool-directory. There are pre-configured newsreaders in our S.u.S.E. Linux package (series `e`). If you want **trn** to access the local spool directory you should install the package `trn-spl.tgz`; for use with a nameserver take package `trn.tgz` package. In the latter `/etc/nntpserver` has to contain the name where the NNTP-server runs.

17.8 Faxing with Linux

There are two alternatives if you plan to use your Linux machine as a faxmachine.

- Either you use **mgetty** together with **sendfax**,
- or you install the **HylaFAX** fax server. Here you have a frontend **SuSEFax** which is designed in Java.

> FromS.u.S.E. Linux version 5.0 the package `mgetty` has been separated in two packages (package `mgetty` and package `sendfax`) as some commands of package `hylafax` and package `sendfax` are identical.

The next two sections describe how to setup and configure **hylafax** and **SuSEFax** on S.u.S.E. Linux.

17.8.1 SuSEFax – an HylaFAX fax client

As already mentioned **SuSEFax** has been designed in Java. This in particular means that you need to install the "Java Developers Kit" along with package `susefax`. If you want to use **SuSEFax** on another platform you should install package `susefax` and copy everything from `/usr/lib/SuSEFax` to the destination directory.

The wrapper itself

The program is launched via a small script. This is called a "wrapper" . It in turn invokes the Java interpreter and sets all needed parameters and checks for settings etc. This script is located in `/usr/X11/bin/susefax`. You may setup this program system independently (*System Properties*); table 17.8 shows all properties that have been setup their meaning, default values and more. If you launch the interpreter without any additional options these settings are used. There is only one parameter (normally) that needs to be set. This is `susefax.images`. If you run the program on an operating system with multiuser capabilities this is not needed. This in particular means that each user is assigned a home directory (**UNIX** systems as well as **Windows NT** *not* **OS/2**). If this is not possible on your system you should set the values for
`susefax.setup.path`, `susefax.setup.file` as well as `susefax.phonebook.file` as the tool might not work otherwise. Well, it has not been tested yet!

Property	Default value	Meaning
`susefax.setup.path`	`$HOME`	The directory where the configuration files and the telephone database should reside
`susefax.setup.file`	`.susefaxrc`	Name of the configuration file
`susefax.phonebook.file`	`.susephone`	Name of the address database (telephone numbers)
`susefax.images`	`./images`	Directory where all necessary images are stored

Table 17.8: Die *System Properties* of **SuSEFax**

If you want to change some of these parameters just remove the comments behind the variable names (see file contents 17.8.1).

```
# if you want to store the settings other than
# $HOME/.susefaxrc, then you may place another path and/or
# filename here

SETUPDIR=    # -Dsusefax.setup.path=/where/ever/you/want
SETUPFILE=   # -Dsusefax.setup.file=/what/ever/you/want

# even the phonebook can be renamed to whatever

PHONEBOOK=   # -Dsusefax.phonebook.file=asyoulikeit
```

File contents 17.8.1: Section of the wrapperscript: `/usr/X11/bin/susefax`

Handling – the instructions for use

This is the main application window after the `'Send queue'` has been activated and `'Fetch state'` has been invoked. `'Fetch state'` tells you about jobs that are still queued i. e. faxes that were send within the last minutes. If you did activate the `'Receive queue'` button all faxes that were received within the last days are listed. You may setup the automatic display by clicking on `'Update information'` (enter an interval in the box below and confirm by pressing ⏎). There is a jobs list that tells you what actions are performed on a double click. This depends on which button (`'Receive queue'` or `'Send queue'`) has been activated. In `'Send queue'` you may decide what should happen on a double click; either to get information on the job or to delete it. In `'Extras'` you may select the language (`English` or `German`).

Let's set it up first

Well, before you will get any feedback from the server or launch any activities you need to configure the application. This is done via `'Settings'` in the `'Programm'` menu . All settings will be saved. If you exit the program all settings will be saved either.

Global settings

Here we give you a list of the entries and their meaning:

User name: Here you may enter the name of the user. This is needed for creating fax covers.

EMail: All messages of the fax server is send to this electronic mail address. e. g. if a fax has been removed without having been sent.

User account: The fax server is capable of distinguishing between certain users. You may allow or deny access. That's why you need to enter the account name that is known to the server. You may even assign a password to the user.

Hostname of the faxserver: This is the name of the host where your fax server runs.

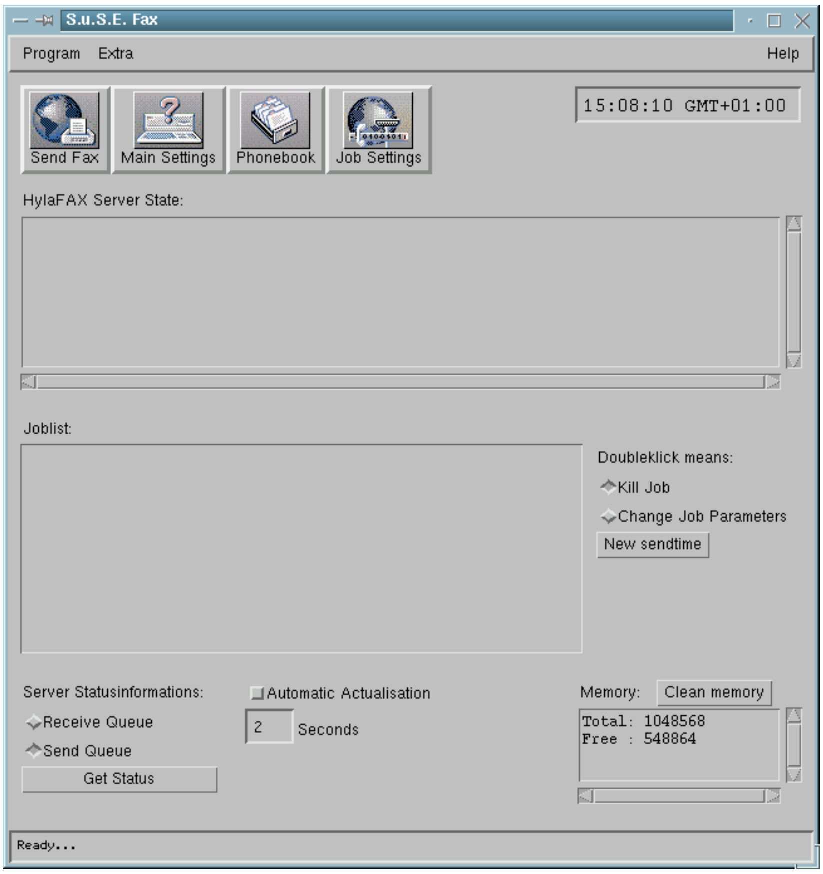


Figure 17.8: Send queue

Automatic sending of faxes: If this button is activated the fax server checks the file given in `'Spool file'` every couple of seconds. If it has changed since you will get a `'Send fax'` automatically. This comes in handy if you print to a file from another application. Thus each application is capable of sending faxes provided it is capable of converting it to Postscript (see section 17.8.3).

Spool file: Here you need to setup the complete path to the spool file if you want to make use of the "automatic fax sending" feature. Clicking on `'Search'` offers a file browser where you may setup this file.

Fax cover: For automatical generation if a fax cover a special PostScript file is needed. Here you need to enter the complete path.

Time zone: This should be the same that you setup on the fax host.

Country: Date and time settings depend on this entry. e. g. on your fax cover.

Job settings

After the global settings have been done you may send your first fax. A fax, or a file respectively is converted to a "job" which waits for being send (perhaps

Figure 17.9: Dialog for global settings

along with some other jobs). There are a couple of job parameters a job. These may be set prior or after queueing the job. Prior to sending the job this may be done in `'Job settings'` which is accessible via `'Extras'`. Here we give a short overview of their respective meaning:

Notification scheme: Here you may set when the fax server should send a notify message to the user whose electronic mail address was set previously. Here there are four different schemes:

- **Never (only errors):** A message to the user is send if an error occurs that sending of a job failed.
- **After sending:** The user is informed after each successful fax sending.
- **After an `'Re queue'`:** The user is send a message if a fax fails if the remote machine is busy e. g..
- **After `'Re queue'` and `'Send:'`** This is a combination of the last two schemes.

Generally the first scheme applies even if you select scheme two.

Resolution: Here you give the resolution of the fax to be send. It is given in lines per inch (lpi).

Priority: Priority of a job in the queue. The default value is set to 127. This is actualizes if, e. g., a job fails (due to busy line e. g.)

Maximum amount of tries: Here you give how often the server should try to send a fax.

Maximum amount of rings: Here you may give how often the server should try to connect to a remote machine (e. g. if the line is busy).

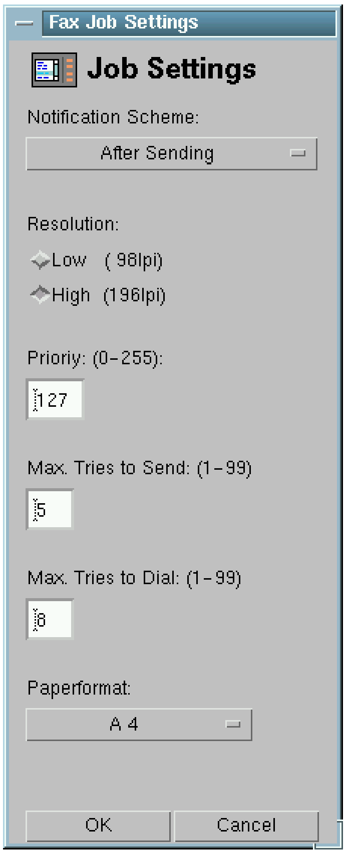


Figure 17.10: Dialog for job parameters

Paper size: At the moment at the moment there are three supported formats: A4, A3 and "North American Letter". This depends on the format of the PostScript document to be send.

If you have activated '`Modify job parameters`' in the main window you will get a dialog window on a double click. Here you may set the parameters '`Notification scheme`', '`Maximum amount of tries`' as well as '`Maximum amount of rings`'.

External viewer

If the job list shows the received faxes you may view the fax using an external viewer. The fax itself uses the `tiffg3`-Format thus your viewer should be capable of displaying this format. This format enables you to combine multiple images within one file. There is a small script that comes with **SuSEFax** that creates a PostScript file (it uses `fax2ps` from **TIFFSoftware** by Sam Leffler [Leffler, 1996b] (package `tiff`)). This is handed to a PostScript viewer. The tool looks for **gv** (Johannes Plass (package `gv`)) if this is not found it uses **GhostView** (by Timothy O. Theisen (package `gs_x11`)) This tool is called `docview` and may be found in `/usr/lib/SuSEFax`

Here are the options for the external viewer

Path to temporary files: Here **SuSEFax** stores the `tiffg3` from the server and replaces the `$F` with the images complete pathname.

> The user who invoked **SuSEFax** needs to have read and write permissions in this directory!

Invoking the viewer: Here you need to enter the *full* pathname to the program that should display the file. It has to accept the files name as a parameter.

Sending a fax

Well, after having set the most important settings you should make sure whether you are able to fetch the status settings of the fax server (see page 302). If not you will not be able to send a fax. You might need to check the configuration of **SuSEFax** and/or **HylaFAX**. Anyway, if everything was setup correctly you should see a dialog when clicking on `'Fax send'`. Here are the different items and their meaning:

Telephone number of the remote host: The telephone number of the remote machine. You may select an item from the telephone book if you click on `'From telephone book'` (of course you will need to assign them first)

Document to be send: Here there should be the full pathname to the document you'd like to send. If you select `'Search'` you may select a file in the browser.

Use fax cover: This may only be activated if you you have setup a fax cover file in the global settings. A fax cover is generated and send.

Do not send immediately: If this is activated a dialog window pops up where you may enter the time for the fax to be sent. If you leave this dialog by clicking `'Cancel'` the fax will be sent immediately after having left with `'Send fax'`.

For time and date settings please consider the following:

- If you have made a selection please confirm by pressing ↵. Now the program checks whether this is an existing value. If not so it is converted to a valid date.

Name of sender: Here the name that has been set in global settings is used.

Name of recipient: No comment...

Regarding: Regarding..

An Firma: This should be obvious...

Comment: No comment...

Selection of `'Poll fax'` assumes that you have setup a telephone number of the fax polling server. Of course you need to give the name of a file that should be sent.

Meaning of the job list

As already mentioned above you may switch between the contents of the send queue as well as the contents of the received queue. Here we give a short overview on the items:

Reception queue

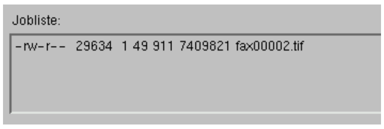


Figure 17.11: Example of an reception queue

Well, from left to right: first there are the permissions, the size (in bytes) number of pages, the TSI as well as the name of the received fax. The TSI (engl. *Transmission Subscriber Identification*) is an identification format the user has setup on his machine. This must not be a telephone number. It might be a company's name instead.

> Faxes received may only be viewed by a double click if the fax server has been setup for every user to read them. For this to work you need to set the value `0644` for **`RecvFileMode:`** in `/var/spool/fax/etc/config.`*`device`* (see section 17.8.4)

Send queue

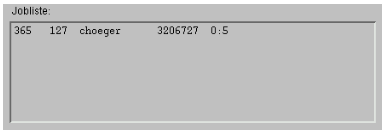


Figure 17.12: Example of an send queue

From left to right: first there is the job ID, the priority of the job, the user account, the target telephone number, the time and/or number of tries and the maximum number of rings. If an error occurred you will see the correspondent error message. The job ID is assigned automatically by the fax server. You may set a priority a user. Anyway the server may correct this setting. The user account tells you which user has launched the job. He is the only one who can delete the job or change its parameters.

The telephone book

The telephone book (see figure 17.13)is for administering and maintaining your personal phone numbers. You may list them sort them according to names, surnames, fax numbers as well as company. Double clicking an entry will launch the editor. You may now select another item and its entries will be presented. `'Accept entry'` add the item to your list required you filled out the following entries: `'Surname'`, `'Name'` und `'Telephone number'`.

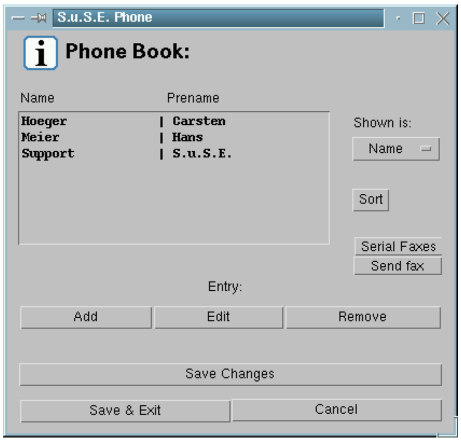


Figure 17.13: The telephone book

If you have activated the `'View entry'` mode double clicking on `'Send fax'` will invoke the `'Send fax dialogue'` using the entries `'Surname'`, `'Name'`, `'Telephone number'`, `'Company'` and `'Comment'` as defaults for the cover page (see page 306).

Of course the cover will only be send if you have activated the item `'Use fax cover'`. If you have launched an activity that disables the selection bar (e. g. by `'Sort'`) you may reactivate it afterwards by selecting an entry.

A new entry is added as soon as you click `'Add'`. `'Delete'` removes the currently selected entry. The list is only sorted if you explicitly activate the `'Sort'` button. The order is given in `'Sort order'` (order is increasing alphanumeric). The buttons `'Save changes'` and `'Save & Exit'` are disabled as soon as you invoke `'Accept entry'` in either `'Sort'` or `'Delete'`.

The telephone book may be used standalone too. Just enter **`susephone`** and the wrapper will be called. Obviously you may not send faxes with this tool.

> Do not invoke the telephone boot twice a user. If so, please make sure not to save it simultaneously.

The faxed form letter

The form fax dialog enables you to generate a serial fax list from the telephone list. For adding (or removing) an item it has to be marked. You may select either by mouse of by one of the toggle buttons. If `'⇐Toggle'` is activated all selected entries will be marked as disabled and vice versa. Same applies to the list. `'Send faxes'` causes all faxes in the list to be send. You cannot create fax cover for serial faxes.

17.8.2 Automatic generation of the fax cover

As mentioned above you will need a PostScript template for automatic generation of fax covers. This by itself is no PostScript file but a template that includes certain holders that are inserted automatically during creation of the cover. Creation of a template may be time consuming. If you are familiar with LaTeX you may use the package `latex-cover`[7]. Here there is an easy to handle TeX style for creating covers with TeX. The cover of used for this package and for package `hylafax` has been created with this versatile tool.

If you don't want to use it you will have to create a "normal" PostScript file and insert the entries by hand.

Which options are known to SuSEFax?

If you're up to modify the TeX document please be aware that **SuSEFax** only replaces the following macros:

`\toperson`
`\from`
`\regarding`
`\tocompany`
`\todaysdate`
`\comments`

If you want to check your self-made template you may use the **faxcover** tool which is included in package `hylafax`. This will create a PostScript file out of the template. Now you may print it or view it. You may as well use the Java binary **FaxCovergen.class**. from package `susefax` Just enter the following:

```
newbie@earth:/home/newbie > java -classpath
    /usr/lib/java/lib/classes.zip:/usr/lib SuSEFax.FaxCovergen
```

Now you should see:

```
Command: FaxCovergen sourcecover.ps docname.ps targetcover.ps
```

The source cover is your template. `docname.ps` is the document that will be send. It will be saved under `targetcover.ps`. Now you may view either of them.

17.8.3 Fax spooling on UNIX/ Linux

The spooling mechanism of **SuSEFax** has originally been designed for the use with **Windows**. Anyway you may use it on **Linux** as well. For this to work you will need to install package `faxprint`, series `n` (Network support).

Now, if you convert `/etc/passwd` to an PostScript file by entering the command `a2ps -nP /etc/passwd | lpr -Pfax` there should be a file in `/tmp` called `fax_`*`accountname`*`.ps`. *accountname* is just your login. If this file exists you may enter it as a spool file as given in section 17.8.1 and activate the `'Automatic fax'` button.

[7] It will be installed in `/usr/doc/packages/hylafax` when installing package `hylafax`

> The spooling mechanism only works if **SuSEFax** is up and running. If this is the case it checks the time stamp `Lastmodified` of the spool file regularly and opened if it has been changed.

17.8.4 HylaFAX – distributed faxes

Function

Installation and configuration of **HylaFAX** is not covered by our installation support (see section 2.1.2, page 10); you may contact our business support team. This service is not free of charge (see section 2.1.2, page 11).

figure 17.14 shows how the fax server works and how to interact with the client. As you may see there are three different ways for communicating with the server. The protocol that is used on port 4557 is still used due to compatibility reasons with older versions of **HylaFAX**. **WinFlex** by Peter Bentley e. g. runs on **Windows** and still uses this protocol. New clients should stack to the new protocol behind Port 4559. This protocol is based on *File Transfer Protocol, RFC959*. The third available protocol is SNPP *(Simple Network Paging Protocol, RFC1861)*.

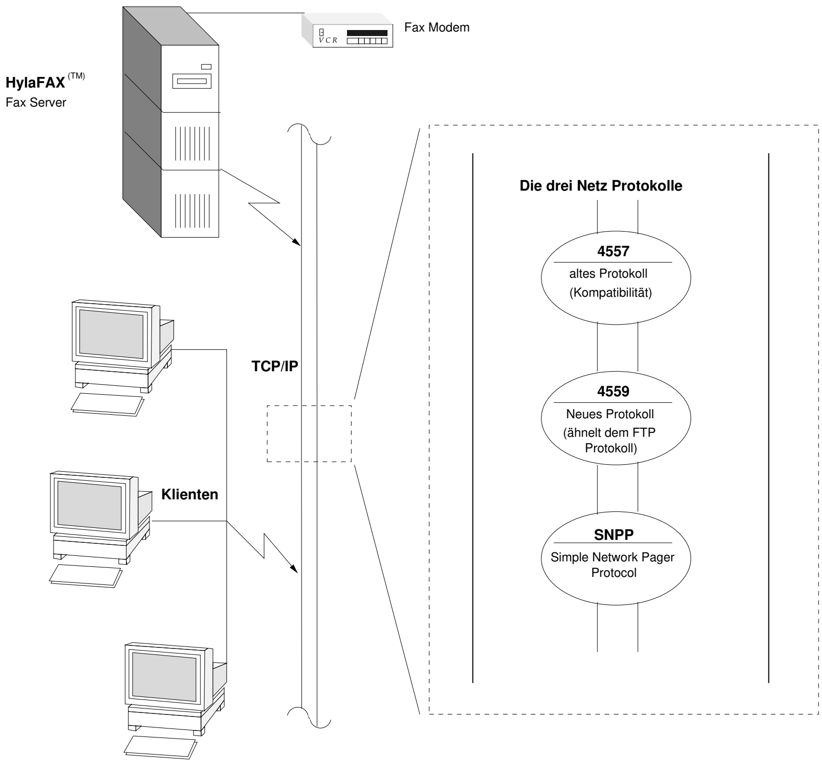


Figure 17.14: Functioning of **HylaFAX**-Servers

The server itself consists of three different daemons. Each of them is responsible for a certain task:

hfaxd This is the protocol server. It is responsible for the communication between the client and the server. It may be launched standalone e. g. via the **Init** process or by the **Inetd**. It uses a file "FIFO special file" with the **faxq** process itself.

faxq This is the so called "Queueing Agent". It is responsible for maintaining in and outgoing faxes as well as the job queue. This process runs all the time. Make sure there is only *one of them* running.

faxgetty This tool is responsible for the communication between the server and the modem. As an alternative for **faxgetty** you may use the **faxmodem** tool (if you want to send but not receive faxes!). Furthermore you might administrate it via a FIFO file[8].

Directory structure

The server itself runs in a "Change-Root" mode. By default this *Server-Root* lies under `/var/spool/fax`. All processes and the *Server-Root* directory itself belong to `'uucp'`. table 17.9 shows a list of all directories that may be found under *Server-Root* and gives a short introduction.

`archive`	Here jobs are archived if *job archival support* was activated.
`bin`	Here all scripts that are used by: `faxq`, `faxsend`, `pagesend` and `faxgetty`.
`client`	This is for the FIFO files that serve for communicating with `faxq`.
`config`	Configuration, permissions and user accounts may be found here.
`etc.`	
`dev`	Since the whole system runs in `chroot` you will find all character devices needed (`null`, `socksys` und `tcp`).
`docq`	This as well as `tmp` are used for pre checking jobs e. g...
`doneq`	Jobs that were done but neither archived nor deleted.
`etc`	see `config`.
`info`	This is for general information on hosts that are already known to **HylaFAX**.
`log`	Here you will find the logs of sender and reception
`pollq`	This is for documents that will be get by polling the server.
`recvq`	Outgoing faxes...
`sendq`	Incoming faxex.
`status`	Status information of the server itself.
`tmp`	see `docq`.

Table 17.9: The *Server-Root* directories and their function

In addition there are some FIFO files. The file `/var/spool/fax/FIFO` itself as well as one `/var/spool/fax/FIFO.devname` per modem that is main-

[8] A FIFO file (FIFO = First In First Out)

tained by **faxgetty**. `devname` stands for the device where the modem is connected too.

Configuration

The configuration itself is divided into 2 up to 2+n configuration files. Here `'n'` gives the number of modems used. In `/var/spool/fax/etc` you will find the files `config` and `config.`*`device`*. The latter configures the modem attached to *device*. So, if there is a modem connected to `/dev/ttyS0` the name would be `config.ttyS0`.

General settings are entered in `config`. These are needed by the scheduler process `faxq` itself. Modem specific setup may be found in `config.`*`device`*. Settings for the queueing service may be found in `config`. The files for the protocol server are in `/usr/lib/fax/hfaxd.conf`. These configuration files are created automatically if you invoke **`faxsetup`** after installation.

Example configuration

Here you see an example session with **`faxsetup`**. We will use the data mentioned in table 17.10:

Telephone number	(0)49(0)911-3206728[9]
Modem Fax-Class	2.0

Table 17.10:

The **bold** letters are user entries.

Configuration of the scheduler

- `Should an entry be added to /etc/inetd.conf [no]?` ↵
- `Country code [1]?` **49**
- `Area code []?` **911**
- `Long distance dialing prefix [1]?` **0**
- `International dialing prefix [011]?` **00**
- `Dial string rules file (relative to /var/spool/fax) ["etc/dialrules"]?` ↵
- `Tracing during normal server operation [1]?` **527**
- `Default tracing during send and receive sessions [0xffffffff]?` **527**
- `Continuation cover page (relative to /var/spool/fax) []?` **etc/cover.templ**
- `Timeout when converting PostScript documents (secs) [180]?` ↵
- `Maximum number of concurrent jobs to a destination [1]?` ↵
- `Define a class of modems []?` **"any"**

[9] Of course you should use your telephone number and enter it correctly in `faxsetup`.

- `Time of day restrictions for outbound jobs ["Any"]?` ↵
- `Pathname of destination controls file (relative to /var/spool/fax) []?` ↵
- `Timeout before purging a stale UUCP lock file (secs) [30]?` ↵
- `Max number of pages to permit in an outbound job [0xffffffff]?` **30**
- `Syslog facility name for ServerTracing messages [daemon]?` ↵

After the data has been entered you will get a conclusion of what you just entered as given in screen output 17.8.1.

```
The non-default scheduler parameters are:

CountryCode:            49
AreaCode:               911
LongDistancePrefix:     0
InternationalPrefix:    00
ServerTracing:          527
ContCoverPage:          etc/cover.templ
MaxSendPages:           30
ModemClass:             "any"
SessionTracing:         527
```

Screen output 17.8.1:

Configuration of the server

After the scheduler has been configured **`faxsetup`** requests if you want to setup your modem using **`faxaddmodem`**. Obviously you should reply `yes`. Now the serial line has to be entered (without entering the full pathname) just `modem` if it is located in `/dev/modem`.

- `Country code [49]?` ↵
- `Area code [911]?` ↵
- `Phone number of fax modem [+1.999.555.1212]?` **+49.911.3206728**
- `Local identification string (for TSI/CIG) ["NothingSetup"]?` **"S.u.S.E. GmbH"**
- `Long distance dialing prefix [0]?` ↵
- `International dialing prefix [00]?` ↵
- `Dial string rules file (relative to /var/spool/fax) ["etc/dialrules"]?` ↵
- `Tracing during normal server operation [1]?` **527**
- `Tracing during send and receive sessions [11]?` **527**
- `Protection mode for received facsimile [0600]?` **0644**

- `Protection mode for session logs [0600]?` ↵
- `Protection mode for modem [0600]?` **0666**
- `Rings to wait before answering [1]?` ↵
- `Modem speaker volume [off]?` ↵
- `Command line arguments to getty program ["-h %l dx_%s"]?` **"-r -b -s %s %l"**
- `Pathname of TSI access control list file (relative to /var/spool/fax) [""]?` ↵
- `Pathname of Caller-ID access control list file (relative to /var/spool/fax) [""]?` ↵
- `Tag line font file (relative to /var/spool/fax) [etc/lutRS18.pcf]?` ↵
- `Tag line format string ["From %%l|%c|Page %%p of %%t"]?` ↵
- `Time before purging a stale UUCP lock file (secs) [30]?` ↵
- `Hold UUCP lockfile during inbound data calls [Yes]?` ↵
- `Hold UUCP lockfile during inbound voice calls [Yes]?` ↵
- `Percent good lines to accept during copy quality checking [95]?` ↵
- `Max consecutive bad lines to accept during copy quality checking [5]?` ↵
- `Max number of pages to accept in a received facsimile [25]?` ↵
- `Syslog facility name for ServerTracing messages [daemon]?` ↵
- `Set UID to 0 to manipulate CLOCAL [""]?` ↵

The summary given in screen output 17.8.2 is created according to your settings entered above.

So far for configuration of the scheduler and the server.

Now you may set if you want to start **`faxmodem`** for each of the configured modems. This serves as an alternative to **`faxgetty`** which is only capable of "send only". You may choose either of them.

Adaptive Answer Support

A rather handy feature of the fax server (**faxgetty**) is the so called "Adaptive Answer Support". It enables the server to launch any **getty** depending on the type of incoming call. This might be a data call as well. For this to work make sure the entry in file contents 17.8.2 is configured (see page 312).

Here `%s` is a substitute for the DTE/DCE between machine and modem. This is set to 38400 bps (bits per second) by default. Some types manufactured by **USRobotics** cannot support this rate (see docu [Leffler, 1996a]). Thus transceiving errors may occur. This may be avoided by decreasing this

```
The non-default server configuration parameters are:

CountryCode:             49
AreaCode:                911
FAXNumber:               +49.911.3206728
LongDistancePrefix:      0
InternationalPrefix:     00
DialStringRules:         "etc/dialrules"
ServerTracing:           527
SessionTracing:          527
RecvFileMode:            0644
DeviceMode:              0666
RingsBeforeAnswer:       1
SpeakerVolume:           off
GettyArgs:               "-r -b -s %s %l"
LocalIdentifier:         "S.u.S.E. GmbH"
TagLineFont:             etc/lutRS18.pcf
TagLineFormat:           "From %%l|%c|Page %%p of %%t"
MaxRecvPages:            25
```

Screen output 17.8.2:

```
GettyArgs:               "-r -b -s %s %l"
```

File contents 17.8.2: Entry for the Adaptive Answer Support

<ModemRate> to 19200 in the appropriate modem configuration file. The default **getty** is **mgetty** (package `mgetty`, series `n` (Network support)). Well for this to work you need to adapt your mgetty configuration file ((`/etc/mgetty+sendfax/mgetty.config`)) by the entries given in file contents 17.8.3:

```
port modem
direct y
toggle-dtr n
```

File contents 17.8.3: Entry in `mgetty` configuration file

The key **`modem`**[10]. This is your modem's device name. Make sure that **faxgetty** as well as **mgetty** use the same device.

Fax Dispatching

An interesting feature is the redirection of incoming faxes to an given electronic mail address. You will need to create a file `etc/FaxDispatch` in the *Server-Root* directory. file contents 17.8.4 shows an example configuration:

[10] If it is `/dev/modem` refers to a link to `/dev/ttySx`

```
case "$SENDER" in
*0815*) SENDTO=newbie;;
*)      SENDTO=FaxMaster;;
esac
```

File contents 17.8.4: Example of `etc/FaxDispatch`

The incoming faxes are identified by their TSI. In our example every fax with `0815` in their TSI will be automatically forwarded to the user '`newbie`' via electronic mail(as PostScript attachment)[11]. Furthermore every incoming fax is redirected to '`FaxMaster`'.

If you encounter any problems while configuring **HylaFAX** please have a look at our SupportDatabase (Help system package `susehilf`, series `doc` (Documentation)). Enter the keyword "fax"and you will find lots of useful information.

[11] Of course you may change the file format by modifying **bin/faxrcvd**

Chapter 18

Samba PC Server

Using **samba** developed by **Andrew Tridgell]** from Australia one can convert any Unix machine into a powerful file and print server for DOS and Windows machines. Since its start in 1991 Samba has proven to be a stable and reliable product which made its way into companies and serves as supplement or even as a replacement for **NetWare**- or **Windows NT** servers.

In the meantime Samba has become a rather complex product. We can not cover all details here in our manual but only an overview on its functionality. In `/usr/doc/packages/samba` many a documents can be found that help you building complex network solutions with Samba. Samba's reference file `/etc/smb.conf` can be found on an extra man page (manpage of **`smb.conf`** (**`man smb.conf`**))

18.1 Introduction

Concepts of PCs and Unix differ considerably. That is why we want to give a short overview of PCs architecture with **NetBIOS**.

18.1.1 NetBIOS

NetBIOS is a software interface (API) which has been designed for communication between machines. Here a so-called name service is provided. This enables machines connected to the net to reserve names for themselves. After reservation these machines can be addressed by their names. There is no central process that checks names. Any machine on the net can reserve as many names as it wants provided the name is not already in use.

This dynamic architecture has its origin in PC networks where installation of a new network node had to be made as easy as possible. Configuration of a machine was reduced to giving it a name. Problems with unique names, maximum length of names is 16 characters, could be ignored as the networks were not so extended.

Besides the name service there are services for communication itself. There are secure and insecure data streams. These might be compared to TCP and UDP protocols in the Unix world. The higher protocols such as the SMB protocol are the layer on top of them.

The NetBIOS interface now can be implemented for different network architectures. An implementation works relative close with network hardware and is called **NetBEUI** which is often referred to as **NetBIOS**

For addressing the single packages NetBEUI works together with the hardware address of the adapter. On the contrary to **IPX** or IP addresses you can not gather routing information from it. Thus it is not possible to transfer NetBEUI packages via routers and a net running NetBEUI is reduced to the range that can be reached by repeaters and bridges.

Further network protocols that have been implemented with NetBIOS are IPX from **Novell** and TCP/IP. The protocol to lay NetBIOS onto TCP/IP is described in RFC 1001 and 1002. RFC 1001 furthermore contains a rather good and understandable introduction in NetBIOS' concepts which helps while trying to understand services as **WINS**[1].

The NetBIOS names that are given away via TCP/IP have nothing in common with names as in `/etc/hosts` or are given via ☞*DNS*. NetBIOS is a name space of its own. Well, it is recommended to use names that correspond to DNS hostnames for making administration easier. This is the default Samba uses.

18.1.2 SMB

The SMB protocol (Server Message Block) serves for making file and print services in the Windows and **LAN Manager** available. Then SMB protocol is based on NetBIOS services thus it is comparable to NFS. Here there is no difference to other protocols as NetWare Core protocol. Amazingly Microsoft has released the specifications of the SMB protocols so that others can now support SMB as well.

Microsoft has extended the SMB protocol recently and renamed it to **CIFS** (Common Internet File System). Microsoft now tries to establish CIFS as the internet standard for gaining more support for this protocol[2]

Well, Samba is a server that implements the SMB protocol under Unix. Samba can turn any Unix machine into a file and print server for most PC operating systems. Samba has been ported to any "intelligent" operating system. It runs on the real-time QNX as well as on a Cray. Novel has ported Samba to NetWare 4.1 and calls it "Migration Toolkit" providing an easy means for Windows NT users to switch to NetWare.

18.1.3 Clients

Except for DOS and Windows 3.1 any current PC operating system supports the SMB protocol for importing and exporting. Windows for Workgroups also supports SMB; in its standard installation only via IPX and NetBEUI. For using Samba which can only provide SMB via TCP/IP additional software has to be installed which (if needed) may be obtained from `ftp://ftp.microsoft.com/bussys/Clients`.

SMB servers provide hardware space to their clients by means of so called shares . Here a share includes a directory and its subdirectories. It is exported

[1] WINS is nothing more than an extended NetBIOS name server and *not* an idea of **Microsoft**; only the name is new!

[2] For those of you who are interested in CIFS specifications have a look at `ftp://ftp.microsoft.com/developr/drg/CIFS/cifs`; in `/developr/drg/CIFS` there are the older specifications of SMB

by a name and can be accessed by its name. Of course the Share name can be set to any name. It does not have to be the name of the to export directory. A printer also is assigned a name; clients can access the printer by its name.

18.1.4 Permissions

An NFS server is configured via `/etc/exports`. Additional restrictions are only possible a single machine. In NFS this makes sense since it was designed for Unix workstations that themselves check permissions and authentification. Nowadays in times of Windows and OS where any user can gather "root" permissions the NFS protocol is no longer suitable. NFS clients for DOS must be regarded as immense security gaps!

The SMB protocol comes from within the DOS world and directly covers the security issues. Any access to a share can be protected by a password. SMB now has two alternatives for providing this:

- Share Level Security

  A password is directly assigned to any share. Anybody who knows this password can access the share.
- User Level Security

  This variant introduces the user concept in SMB. Every user has to login at the server using his login and password. After successful login the server can now grant access dependent on the users permissions.

Distinction between share and user level security has to be set for the server as a whole. It is not possible to export some shares as share level security whereas others are exported as user level security.

Per default Samba is set to share level security. Here the home directories of the users are protected by the normal user password. For other shares a user has to be entered by giving the `user` option whose password is used as protection. The parameter **`security`** may be given as **`security = user`** in `smb.conf`. Then users are validated as usual under Unix using `/etc/passwd` and `/etc/group`. Samba offers also a third possibility by providing **`security = server`**. If this option is activated Samba validates the user on another (NT) server which has to be set by the option **`password server`**.

18.2 Installation of the server

Almost anything that can be configured is done in `smb.conf`. This file resembles a Windows .INI file. It is separated in different sections which by themselves contain certain parameters. Generally a share is described per section whose name is given by the sections name. Furthermore there are three special sections as well. These are **`[globals]`**, `[homes]` and **`[printers]`**. In **`[globals]`** parameters are set that are not specific to a certain share. If the option `[homes]` is set any user on the server can access his/her home directory without having to define a home-share for each user. Same applies to printers; all printers in `/etc/printcap` are accessible without setting them separately.

18.2.1 smb.conf

A simple example file can be seen in file contents 18.2.1.

```
[global]
   workgroup = workgoup
   guest account = nobody
   keep alive = 30
   os level = 2
   security = share
   printing = bsd
   printcap name = /etc/printcap
   load printers = yes

[homes]
   comment = homedirectory
   browseable = no
   read only = no
   create mode = 0750

[printers]
   comment = All Printers
   browseable = no
   printable = yes
   public = no
   read only = yes
   create mode = 0700
   directory = /tmp
```

File contents 18.2.1: Example for `/etc/smb.conf`

This `/etc/smb.conf` provides access to the home directories of the users as well as all printers listed in `/etc/printcap`.

- **`workgroup = workgroup`**
  Any Windows machine Samba is assigned just like a work group where it can be seen in the "network environment". "Work group" is the default setting for Windows for Workgroups.
- **`guest account = nobody`**
  Samba needs a user that is listed in `/etc/password` that only has limited permissions for certain tasks. If public shares (Parameter **`public = yes`**) are defined all operations are executed with this user ID. Even if no public share is defined the account **`guest account`** has to be defined for the Samba machine to appear in the networking environment.
- **`keep alive = 30`**
  Windows machines tend to crash from time to time. If they leave open connections behind it could happen that the server recognizes this very much later. If you do not want Samba to waste resources you can tell it to have a look whether the client is still alive by giving the option **`keep alive = 30`**.

- **`os level = 2`**

  This parameter **`os level = 2`** assigns that Samba provides browser services to WfW and Windows 95. If there is a NT machine on the net Samba will not provide these services to it but use the NT machine itself.
- **`security = share`**

  See section in permissions.

The following three parameters serve for reading `/etc/printcap` and to export any mentioned printer. The section **`[homes]`** assigns parameters for the home directories. These directories are reachable via the users name.

- **`comment = homedirectory`**

  Any share can be given a comment with SMB servers which classifies the share.
- **`browsable = no`**

  This setting prevents the share `homes` being visible in the networking environment. Any user can reach its home directory via its name.
- **`read only = no`**

  Per default Samba prohibits write access on exported shares. Logged in users should have the possibility to write in their home directories anyway, thus **`read only = no`** has to be set.
- **`create mode = 750`**

  Windows machines do not know the concept of Unix permissions. That is why the cannot assign permissions while creating a file. The parameter **`create mode`** assigns which permissions should be used while creating a new file.

18.3 Installation of clients

I consider DOS, Windows for Workgroups, and Windows 95 to be important clients.

First of all it should be mentioned that clients can access Samba only via TCP/IP. NetBEUI and NetBIOS via IPX are not available at the moment. Since TCP/IP becomes more and more "popular" even on Novell and Microsoft it is not certain whether this is going to change in the near future.

18.3.1 DOS and Windows 3.1

DOS has not been designed for networks. For making DOS able to access Samba a protocol stack from card drivers to network redirect has to be loaded. This protocol stack has been written for "naked" DOS and makes plenty use of memory below 1 MB. There is not very much memory left for applications. For those of you who want to continue using DOS programs it is recommended to switch to Windows 95 or Windows for Workgroups and run these programs in a DOS box. These two operating systems enable you to use TCP/IP without reducing DOS' conventional memory. This is achieved by a client software that makes use of the protected mode and directly accesses memory above the 1 MB limit.

If this is not possible either as a 80386 processors (or less) is used or Windows is not available or the applications do not run in a DOS box, **mars_nwe** from Novell and its client software should be used. The Novell client software has been designed not to use an immense amount of memory. You can get the NetWare client either from an existing NetWare installation or get a 2 users **NetWare 4.1** test license from a bookstore. Many books on NetWare 4.1 also contain this CD.

But even DOS cooperates with Samba. The software required can be obtained from `ftp://ftp.microsoft.com/bussys/Clients/MSCLIENT`. These are two self extracting archives that have to be unpacked on a DOS disk each. On the first disk there is the setup program for automatically initializing the protocol stack. This setup program first selects NetBIOS vis IPX as protocol. This has to be replaced by adding the TCP/IP protocol first and thereafter remove IPX.

The setup program configures the networking software for usage of an DHCP server which might not be available under Linux. Therefore you have to set the parameters explicitly in the TCP/IP protocol settings and disable automatic configuration by setting **`Disable Automatic Configuration`** to `1`. If automatic configuration is not disabled the machine will hang at the next reboot while waiting for the DHCP server. You should keep in mind that MSCLIENT does not give IP addresses as usual (as `192.168.0.20`) but as `192 168 0 20` (without the dots).

18.3.2 Windows for Workgroups

Windows for Workgroups already has the SMP support built in. This is called a client for Microsoft networks and normally runs on NetBEUI or IPX. TCP/IP connection has to be installed additionally. The TCP/IP protocol stack is freely available at `ftp://ftp.microsoft.com/bussys/clients/WFW`. `TCP32B.EXE` is a self-extracting archive that has to be unpacked onto a disk in advance.

Now for the installation. First you have to enter `'not listed or actualised protocol'` in `'drivers'` (network setup) which is on disk.

After the contents of the disk has been copied the TCP/IP parameters have to be set which is done under `'Settings'`. Here you need (same as with Linux) the IP address of the machine, a netmask and perhaps an existing gateway. More options can be set under `'Extended'`. Thereafter you should set TCP/IP as standard protocol or remove IPX/SPX and NetBEUI. This reduces the amount of memory being used and accelerates the start of Windows. If there are other machines that rely on this protocol you may, of course, not remove them!

After installation a restart is obligatory. Thereafter you can access the Linux machine via Samba the same as with other machines.

For using a printer on the Samba server one should install the general postscript printer driver of Windows for Workgroups and connect it to the Linux queue which includes the apsfilter recognition possibilities.

18.3.3 Windows 95

Windows 95 has the TCP/IP support already built in. Same as with Windows for Workgroups it is not installed by default. For adding TCP/IP you have to choose `‘Add’` (in `‘System’`), `‘Protocols’` TCP/IP from Microsoft. For the rest the same applies as for Windows for Workgroups.

Chapter 19

First steps with Linux

Since **Unix** is rather a complex system we can only cover the most important aspects. This book is not aimed at replacing existing literature on Linux (or UNIX)— this is impossible and this book is not regarded as replacement.

Anyway we recommend that ex DOS-users consider buying a book (see page 341). If you call some Unix literature your own (not Linux specific) this might be enough to start, since most of Unix' features are known under Linux as well.

Some books that are published under **GPL** are located on CD 3 as `.dvi` files in directory `/usr/doc/books`. You can view those files using **xdvi** (under X11) and print them using **dvips**. (Hints can be found in the manual pages).

or in ⟶ Why LaTeX (Page page 380)).

Well, until you have your book at hand this chapter should give you a short overview and as well support you while "experimenting" with Linux for the first time.

Being a Unix novice you should log in with your `'user name'` (*not* as root!). First, there are a lot of settings that already have been dealt with for you and second, you're only responsible for your private *home* directory. This only serves for security purposes; to avoid deletion of system-relevant data.

There is *no* undelete under Linux (as under DOS); so, if you delete system files by accident you probably have to reinstall the whole system.

The first steps are just a little complicated since you are the system administrator as well, which is not an easy task for a novice user.

19.1 Logging in, "root", adding users

Since Linux is a multiuser system you have to login on the ☞*console* anytime you want to use your system. This is called *login*. You enter your username (e. g.`'newbie'`) and enter your password (only Enter at the first login).

```
login: newbie ↵
```

If this was succesful you will find yourself in your "own" *home* directory (e. g. `/home/newbie` for the user `'newbie'`).

If you want to leave, you log out using **`logout`**; that's all.

`'Root'` is the *System administrator* (☞*sysadmin*)he is allowed to *do everything*. All important system information files can only be changed by `'root'`.

Therefore you should only log in as `'root'` if you plan to configure something or to execute system specific tasks. Never log in as `'root'` if it is not necessary (most novices forget this)! Thus you protect your system against unwanted damages!

Some things only `'root'` can do:

- ☞*mounting* filesystems (such as CDs, floppies, installation of software etc). This right can be given to certain users by adding the option `user` for the corresponding device in `/etc/fstab`.
- Adding and deleting of users.
- Installation of a new kernel.
- Configuring the system.
- Shutting down the system.

19.2 Shutting down and booting

> *IMPORTANT!* You should never turn off the machine while it is running nor press the *reset button* (☞*reset*) If you switch it off without bringing it down properly loss of data and damage of your ☞*filesystem* is almost certain!

For shutting down your system properly there are the commands showed in table 19.1 :

`shutdown -h now`	halts the system and (when you see the output: `"the system is halted"` you can switch off your machine)
`shutdown -r now`	reboots the system immediately

Table 19.1: Commands for halting your Linux system

Commands **`shutdown`** can only be invoked by `'root'`.

For shutting down your system log in as `'root'` and enter the commands **`shutdown -h now`** or **`shutdown -r now`**.

Alternatively you can use Strg + Alt + Entf (Ctrl + Alt + Del) (which you might know from booting DOS). This combination does not work if you are currently running X11; nevertheless this method can be used by any user on any virtual console.

19.3 Virtual consoles

Linux is a multiuser and ☞*multitasking*- system. You will appreciate these features even on one-PC systems:

In textmode there are 6 virtual ☞*consoles* available; you can switch to either of them by using Alt + F1 to Alt + F6 . The seventh console is reserved for X11.[1]

If you want to switch to a console from X11 without leaving X11 you should use either of: Ctrl + Alt + F1 to Ctrl + Alt + F6 . Alt + F7 brings you back to X11.

19.4 Adding and deleting users

You can add users by using the **useradd** command. Again, this can only be done by 'root'. A good alternative for invoking this command by hand is to use YaST for adding users (see page 86).

Example:

```
root@earth:/ # useradd -u 300 -g users -d /home/newbie \
    -s /bin/bash -m newbie
```

adds user 'newbie' with its home directory /home/newbie, user-id 300. 'newbie' belongs to group 'users' and uses **bash**. Now *root* can give a password to *Emil* by:

```
root@earth:/ # passwd newbie
```

'root' gives a password to the user 'newbie' which he uses for logging in. 'newbie' can (and should!) change his password after he is logged in for the first time.

The 'root' password is given in exact the same manner.

If you are adding a user all files from /etc/skel are copied into the user's home directory providing minimal system-wide configuration to all users.

Of course each user can adapt those files to his/her personal needs.

For comfortably maintaining users it is recommended to use YaST.

19.5 Directories and filenames

The character for separating directories ('\' under DOS) is a '/' under Unix. Thus a path being a string where all directories are separated by '/'. One single '/' stands for the topmost directory, the ☞*"root directory"*.

Upper- and lower case are significant under Unix, meaning that Emil names another filename as emil; separating a filename into its *name* and its *extension* is not necessary, but there are some programs that expect a certain name (e. g.LaTeX).

A nice feature is the *globbing* feature of the (**bash**) shell; just enter the first significant word of a file or command and press ⟼ . This string is now completed by the shell. Pressing ⟼ twice shows all possibilities (if it was not significant).

[1] you can assign more consoles via /etc/inittab

19.6 Working with directories

After having logged in you'll find yourself in your `home` directory. You can check this by typing **`pwd`** (print working directory):

```
newbie@earth:/home/newbie > pwd
  /home/newbie
```

For changing into another directory there is the **`cd`** command (same as under DOS). Typing

```
newbie@earth:/home/newbie > cd /usr/bin
newbie@earth:/usr/bin >
```

changes into directory `/usr/bin`,

```
newbie@earth:/home/newbie > cd latex
newbie@earth:/home/newbie/latex >
```

changes to `latex` provided there is a directory `/home/newbie/latex` in '`newbie`'.

If you invoke **`cd`** without any argument you will be brought back home. This can be reached by typing a tilde ('`~`') as well.

So, typing

```
newbie@earth:/home/newbie > cd ~/latex
```

changes to `latex` in your current `home` directory. Same as under DOS, '`.`' signifies the current directory whereas '`..`' stands for the parent directory.

You can create new directories with **`mkdir`** (make directory);

The command

```
newbie@earth:/home/newbie > mkdir texts
```

creates `text` under the current directory. Empty directories can be removed using the command **`rmdir`** (remove directory).

19.7 Working with files

Until they are replaced by objects or symbols (in future days) files play a vital and central role while working with a computer. Therefore a huge variety of file commands exist under Linux.

19.7.1 Information on files

The command **`ls`** (list) outputs the contents of your current directory. A list of all filenames and directories is presented. Directory names end with a '`/`'. Of course you can see any directory if you enter a directory:

```
newbie@earth: > ls /usr/bin
```

Executable programs do not have any extensions as `.exe` or `.com`, but an asterisk '`*`'. You don't have to set it, it is set by the attribute "e**x**ecutable" behind the filename. (see ⟶ Infos on files (Page page 332))

A rather nice option is `-l`. This gives you a *l*ong list of filenames including their owner, permissions, groups and size.

```
newbie@earth: > ls -l
```

This will create an output as screen output 19.7.1.

The meaning of the entries in screen output 19.7.1 is explained in table 19.2

```
drwxr-xr-x  6  newbie users   1024  Mar 21  12:39 ./
drwxr-xr-x  4  newbie users   1024  Mar 21  17:13 ../
drwxr-xr-x  2  newbie users   1024  Nov 6   16:19 bin/
-rwxr-xr-x  1  newbie users   4160  Mar 21  12:38 check*
drwxr-xr-x  2  newbie users   1024  Nov 6   16:23 etc/
drwxr-xr-x  2  newbie users   1024  Nov 6   16:19 sbin/
drwxr-xr-x  12 newbie users   1024  Nov 6   18:20 usr/
-rw-r--r--  1  newbie users 185050  Mar 15  12:33 xvi.tgz
-rw-r--r--  1  newbie users  98444  Mar 14  12:30 xvnews.tgz
```

Screen output 19.7.1: Output of **ls -l**

Permissions	The first character indicates the file type. 'd' stands for directory, 'l' for a link and '-' is a normal file. The next 9 characters indicate permissions for the user, the group and all other users (three characters each). 'r' stands for read, 'w' for write, and 'x' for execution. For example, '-rw-r--r--' refers to a file which can be read by the owner, the group and all others, whereas it can only be changed by the owner. See manpage of **chmod** (**man chmod**).
Owner	The owner of the file. See manpage of **chown** (**man chown**).
Group	The group the file belongs to. See manpage of **chgrp** (**man chgrp**).
Size	The file's size in bytes.
last change	Date when the file has been changed last. Files that have been changed more than a year ago are marked with the year instead of hours:minutes.
Name	The file or directory's name.

Table 19.2: Explanation of UNIX file attributes

19.7.2 Wildcards

Unlike DOS those ☞*wildcard* interpreting options of the shell (e. g.**bash**) are fascinating. Wildcards are not only reduced to just '*' and '?'.

For example:

```
ls *a???.?
```

gives all files in the current directory having an 'a' as the sixth from the last and the second last one being a '.'. Instead of a single character you can give a whole range of different characters. For example letters 'a', 'b', 'c', 'd', 'e', 'f'. This would be done like:

```
ls *[a-f]???.?
```

You can even search in non alphabetical order:

```
ls *[1,3-5,M-P,a,k]???.?
```

19.7.3 Contents of files

You can view the contents of a file with **less** and **more** page by page. **more** lets you only scroll downwards. Scroll forward with [SPACE] and backwards with [b]. A quite similar command is **cat** (concatenate) which takes multiple files as arguments and writes the contents directly to standard out (normally your screen). Since most files are bigger than just one screen **cat** is mainly used to concatenate files via piping standard out. Entering

```
newbie@earth: > cat one two > oneandtwo
```

does not show the contents of `one` and `two` but writes the result into the file `oneandtwo`

19.7.4 Hidden files

A special sort of files are the hidden files. Those files begin with a dot and are only seen if you pass the shell the special option `-a`. In your home directory just enter **ls -a**; now you should see all files, even these hidden ones (like `.profile` or `.xinitrc`). Hidden files are protected from an otherwise hazardous **rm *** (see section 19.7.5) those files have to deleted separately with **rm** <**.filename**>.

> Entering **rm .*** deletes all hidden files of the current directory! If the option `-r` is added (recursive; **rm -r .***) all files of the parent directory are deleted as well (they are named '`../bla`' which is represented by '`.*`' as well)!
> So be extremely careful with `-r`!

19.7.5 Copying, renaming and deleting of files

The command for copying files under Linux is **cp**:

```
cp source target
```

Now, for copying file `/etc/XF86Config` in one's own home directory you would use:

```
newbie@earth: > cp /etc/XF86Config ~
```

Files can be removed using the **rm** command. A very handy option is `-r` which deletes recursively (removing all subdirectories and their files as well; this is rather comparable to DOS' **deltree**). Entering:

```
newbie@earth: > rm -r bin
```

deletes directory `bin` and all files and subdirectories within. Please use this option with extreme precaution, since there is absolutely no means of recovering (restoring) a deleted file!

You can move files by typing **mv**; syntax is identical to **cp**:

```
newbie@earth: > mv xvnews.tgz XVNEWS.tgz
```

Moves file `xvnews.tgz` to `XVNEWS.tgz` which is not more than just renaming it. It's going to be more interesting if you move whole directory trees:

```
newbie@earth: > mv bin ~/latex
```

This moves the directory `bin` (if there is one) to `~/latex`. All files that used to be under `bin` are now to be found under `~/latex`. Even this command should be used with care since it is very easy to move whole trees to places that you do not remember afterwards . . .

Moving of a complete directory tree only is possible within one filesystem (one partition).

19.7.6 Searching and grepping files

This leads to another useful command: **find**. For searching all subdirectories of the current directory for the file `emil` you should enter:

```
newbie@earth: > find . -name "emil"
```

First argument gives the directory where the search should commence. The option `name` expects a searchstring (wildcards are allowed). Thus, for searching all files containing the string ‘`emil`’ you would have to change the line to:

```
newbie@earth: > find . -name "*emil*"
```

As with all commands we suggest you look at the manual pages for more information. There is a fast way anyway; try finding files with **locate**. Have a look at the corresponding manual pages too.

If you do not want to search for a file but for a string inside a file instead, you should use the **grep** command. This commandline searches for the string ‘`detective`’ in the file `emil`:

```
newbie@earth: > grep "detective" emil
```

Thus you can search vast amounts of data for certain strings very quickly. Any number of filenames can be entered, even searching using wildcards and regular expressions can be used. **grep** sends its results to standard output; it gives you every location where the string was found. **grep** knows many options; please have a look at manpage of **grep** (**man grep**).

19.7.7 Symbolic links

By means of symbolic links you can give another name to a file. This name “points” to the corresponding file. Imagine that you want to keep different versions of a file but the version actually used should be obtainable by the same name (always). The solution is called *symbolic linking* which points to the currently in use version. Symbolic links behave just like the file they link too, thus being executable if the “source” file is executable. The entry

```
newbie@earth: > ln -s check.2.4 check
```

creates a symbolic link `check` which points to `check.2.4.1`. In your directory this looks like figure 19.7.2:

```
lrwxrwxrwx 1 newbie users 1024 Mar 21 17:13 check -> check.2.4*
```

Screen output 19.7.2: A symbolic link using **ls -l**

Links can be removed by **rm**.

> Here only the link is removed not the source file!

19.7.8 Archiving data and saving

For creating and unpacking archives there is the command (tape archive). Normally compressed archives have the extension `z` or `tar.gz`; uncompressed archives the extension `.tar`. Most important uses of **`tar`** are:

1. Unpacking archives (e. g.from CDs)

   ```
   newbie@earth: > tar xvfz archive-file.tgz
   ```

   `tar` unpacks (`x`) the compressed (**`z`**) archive `archivdatei.tgz` (`f`) and assigns necessary subdirectories as well. Each file that is extracted is put to standard output (**`v`**).
2. Creating archives

   ```
   newbie@earth: > tar cvfz archive-file.tgz file1 dir1
   ```

   `tar` creates (`c`) the compressed (**`z`**) archive **`archive file.tgz`** (`f`), where `file1` and all files in `dir1` are contained as well. Each file that is packed is put to standard output (**`v`**).
3. Viewing the archive

   ```
   newbie@earth: > tar tfz archive file.tgz
   ```

   `tar` shows a table of contents (`t`) of the compressed (`z`) archive archiv file.tgz (`f`).

Flag `z` tells **tar** to create/unpack compressed archives using **GNU zip** (**`gzip`**).

```
newbie@earth: > tar xvf archive file.tar
```

unpacks the uncompressed `archivefile.tar`. More information can be found by using

```
newbie@earth: > info tar
```

19.8 Permissions

Only the user `'root'`, being the system administrator, has got unrestricted permissions to all files. Permissions are structured as follows:

- Owner permissions
- Group permissions
- "All others" permissions

Each of this categories is represented by three characters; in conclusion with the first character (file type: d, l, or -) this results in a total length of 10. Each flag is represented by a certain character. The possible flags for all three categories are the same. Reading `'r'`, Writing `'w'` and Executing `'x'`. If a flag is not set there is a `'-'` instead of the flag. Let's have a look at the file *linux.info*:

```
-rw-r-xr--   1 newbie users  29524 Jun 29 13:11 linux.info
```

Well, this file can be read and written by the owner (`newbie`); all members of the group `users` are allowed to read and execute it whereas all other users are only allowed to read it. `'-'` indicates that this is a "normal" ☞*file*.

-	r	w	-	r	-	x	r	-	-
Type	Owner			Group			Others		

Figure 19.1: Overview on file permissions

The same applies to directories; if the given file is a directory there is a `'d'` in front of those 9 characters (d stands for directory). This might look like:

```
drwxr-xr--   3 newbie users  1024 Jun 29 13:11 info/
```

`'x'` in this case means that one is allowed to cd into this directory.

19.8.1 Changing permissions

You can change permissions with the **`chmod`** command (change mode). The main options **chmod** needs are the permissions to be changed and a filename.

- permissions that are to be changed
- a file name.

Those three categories are represented by `'u'` (user), `'g'` (group, and `'o'` (others) followed by the corresponding permissions. A `'+'` or a `'-'` means adding or removing the corresponding flag. Our example sets permissions of the file `linux.info` to readable, changeable and executable for group-members:

```
newbie@earth: > chmod g+rwx linux.info
```

If permissions for all categories should be set, then giving the permissions to be changed is sufficient. This command sets permissions to pathlinux.info so nobody has write permissions:

```
newbie@earth: > chmod -w linux.info
```

Those permissions concerning reading and executing are not involved. You can give permissions and remove them within a single command line. This sets the permissions of `linux.info` to executable, not readable, and not changeable:

```
newbie@earth: > chmod u+x-rw linux.info
```

If you look at the result:

```
newbie@earth: > ls -l linux.info
  ---xr-xr--   1 newbie users  29524 Jun 29 13:11 linux.info
```

This is an interesting command in connection with **`chmod`** are **`chgrp`** (change group) and **`chown`** (change owner).

19.9 Manual pages

Information about commands, configuration files, and C-libraries can always be found by using the corresponding table 19.3.

- **`man -k`** <**`headword`**> searches the headword <**`headword`**> and gives a list of manual pages found.
- **`man -f`** <**`headword`**> searches all sections of manual pages for <**`headword`**> and lists the manual pages found.

- **man** <**headword**> invokes the corresponding manual page for <**headword**>.
- **man** <**section**> <**headword**> invokes the manual page for <**headword**> from section <**headword**>. Thus the command **man 1 man** invokes **man** 's manual page.
- If you have S.u.S.E. help installed, all manuals are available here too; if you are using the X-Window System you might find the program **xman** of some use. Anyway **man** is much faster.

man -k <**keyword**>	searches for <keyword> and lists the man page found.
man -f <**keyword**>	searches for <keyword> in all man page sections and lists the man pages found.
man <**keyword**>	launches the man page for <keyword>.
man <**section**> <**keyword**>	launches the man page for <keyword> from <section> (e. g.**man** from section 1).

Table 19.3: Invoking the **man** command

In the following sections you can find:

1	Describes user commands. Some of them are built in commands (**bash**- und **tcsh**); i. e.manpages of **bash** or **tcsh** give you help.
2	System calls of libraries.
3	C-library functions.
4	Description of configuration files.
5	Syntax of important files.
6	Description of games.
7	Anything that covers text, text formats etc.
8	System Administrators commands.
9	Description of Linux kernel routines.
n	**n** supposed to derive from “new”, here other man pages are listed that belong to another section but have been placed here by reasons of tradition or which do not fit in one of the other sections.

Table 19.4: Man page sections

Please notify that there is *not* a manual page for every command. Perhaps you can find more information under `/usr/doc` (e. g.`/usr/doc/howto`, `/usr/doc/howto/mini`, or `/usr/doc/packages`))

19.10 System information

Sometimes it is important to gather information about the system's state. The commands **`df, free, top, ps`** should help.

19.10.1 The `df` command

`df` outputs information on the used and occupied hard disk space. Output is given as shown in screen output 19.10.1

```
Filesystem 1024-blocks    Used Available Capacity Mounted on

/dev/sda4       699392  659258      5165      99% /home
/dev/sda1       102384   23955     73310      25% /
/dev/sdb1      2097136 2070485     26651      99% /usr
/dev/sda3       126976  106908     20068      84% /opt
```

Screen output 19.10.1: Output of **`df`**

19.10.2 The `free` command

`free` gives information on usage of RAM and swap.

```
             total  used  free shared buffers cached
Mem:         30900 29272  1628  25608    1012   6412
-/+ buffers:       28260  2640
Swap:        66524   176 66348
```

Screen output 19.10.2: Output of **`free`**

19.10.3 The `w` command

`w` shows all currently active users on your system. This command outputs a rather informative list on certain issues. It shows you all users, how long the system has been up and running, its load and the user's current action.

19.10.4 The `du` command

`du` gives information on the space that is used by subdirectories and single files. More can be found under manpage of **`du`** (**`man du`**).

```
11:19pm up 9 days, 11:13, 13 users,  load average: 3.26, 2.80, 2.67
User    tty   from             login@ idle JCPU PCPU what
root    tty2                    2:09pm 9:10              -bash
root    ttyp1 :0.0              2:11pm    2    4    2  xdvi -s 3 Li
root    ttypa :0.0             11:19pm                    w
root    tty1                    2:07pm 9:08   50        (startx)
newbie ttyp0 earth.cosmos.com 11:37am   11    2    2  -bash
root    ttyp3 :0.0              3:24pm         4        rlogin helios
newbie ttyp2 earth.cosmos.com  3:22pm    1   46    2  -bash
root    ttyp4 :0.0              3:27pm 1:48    8        bash
root    ttyp5 :0.0              5:40pm    5    1    1  telnet earth
newbie ttyp6 venus.cosmos.com 3:53pm    3    5    5  -bash
root    ttyp7 :0.0              4:25pm 6:05              bash
newbie ttyp8 helios.cosmos.com 9:37pm 1:30    1        telnet earth
newbie ttyp9 helios.cosmos.com 9:50pm 1:27             -bash
```

Screen output 19.10.3: Output of **w**

19.10.5 The `kill` command

Sends signals to currently active *processes* (☞*prozess*) Expects a process number (PID) as an argument. This PID can be obtained by **ps** (see section 19.10.6). **kill** is invoked:

root@earth: # kill <pid>

If the corresponding process catches the signal it can be killed by giving the optional parameter `-9`

Entering

root@earth: # kill -9 <pid>

kills PID <pid> definitively.

19.10.6 The `ps` command

ps (process status) shows the processes started by the user. More information on this command can be obtained in manpage of **ps** (**man ps**). **ps** shows information on running processes of the other users as well. Giving the process id (1. row) it is possible to kill running processes (see section 19.10.5).

19.10.7 The `pstree` command

pstree shows the whole *process tree*. This is shown in screen output 19.10.4.

19.10.8 The `top` command

Outputs all running processes and their loads and much more. This list is being updated within a given frequency. You can end it using q .

19.11 Filesystem types under Linux

There are a couple of ☞*file systems* available under Linux. This is shown in table 19.5.

```
init-+-bash---startx---xinit-+-X
     |                       `-sh---fvwm-+-FvwmPager
     |                                   |-FvwmWinList
     |                                   |-GoodStuff
     |                                   |-xclock
     |                                   `-xeyes
     |-color_xterm---bash---xdvi.bin---gs
     |-2*[color_xterm---bash---vi]
     |-color_xterm---bash---pstree
     |-coolmail---coolmail---xterm---pine
     |-cron
     |-gpm
     |-inetd
     |-kflushd
     |-klogd
     |-kswapd
     |-5*[mingetty]
     |-4*[nfsiod]
     |-sh---master---slipto
     |-syslogd
     |-update
     |-xload
     `-xosview
```

Screen output 19.10.4: Output of `pstree`

`ext`	*Extended Filesystem*: Ancestor of the `ext2` file systems and without any significant meaning nowadays .
`ext2`	*Second extended Filesystem.* Linux' standard filesystem.
`fat`	*File Allocation Table* usually this is DOS' filesystem but is used by a couple of operating systems nowadays.
`ffs`	*Fast Filesystem* a filesystem used on Amigas.
`hpfs`	*High Performance Filesystem* this is **IBM OS/2**'s standard filesystem. It is only supported in read-only mode.
`iso9660`	Standard file system on CDROMs.
`minix`	This filesystems has its origin in academical projects on operating systems. It was the first file system used for Linux. Nowadays it is used as filesystem for floppy disks.
`ncpfs`	Filesystem for mounting Novell-volumes.
`nfs`	*Network Filesystem.* Here data can be stored on any machine in a network and access may be granted via a network.
`proc`	Process filesystem (virtual)
`smb`	*Server Message Block* is used by products as **WfW**, **Windows NT** und **Lan Manager** and servers for making files accessible over a network.

Table 19.5: to be continued...

`sysv`	Used on **SCO-Unix**, **Xenix** and **Coherent** (commercial Unix systems for PCs) .
`ufs`	Used by **BSD**, **SunOS** and **NeXTstep**. Is only supported in *read-only* mode.
`umsdos`	*Unix on MSDOS* Is applied on top of a normal `fat` filesystem. It achieves Unix functionality (permissions, links, long filenames) by creating special files. It is rather slow. It is required for the demo-mode.
`vfat`	*Virtual FAT* Extension of the `fat` file system (supports long filenames).
`xiafs`	An old filesystem that is hardly used anymore.

Table 19.5: Filesystem types under Linux

19.12 Mount and unmount filesystems

Via the command **`mount`** (which can only be invoked by '`root`'), a storage media can be linked into the Linux filesystem tree. Two arguments are required by **`mount`**;

- Name of the device (corresponds to the device's name; e. g. `/dev/hda3`)
- And, of course, where it should be attached (mounted).

Option **`-t`** <**`filesystem type`**> passes the type of the filesystem (see table 19.5).

```
root@earth:/ # mount -t msdos /dev/hda2 /dosa
```

makes DOS partition `/dev/hda2` available under `/dosa`.

Passing the **`-r`** option mounts it read-only. Now writing on this filesystem is not allowed.

`mount` protocols the mounted partitions. This protocol can be found in `/etc/mtab`; if **`mount`** is invoked without any argument the contents of this list is printed to screen. This corresponds to all mounted filesystems.

19.12.1 umount

`umount` removes a partition from Linux' filesystem[2].

You can pass either the name of the device or the name of the directory where it is mounted as an argument to **`umount`**. So, for removing `/dev/hda2` which is mounted under `/dosa` you can enter either of:

```
root@earth: # umount /dosa
root@earth: # umount /dev/hda2
```

[2] This command used to be called `unmount` but the '`n`' got lost somewhere in Unix history.

It is important (if you have mounted a disk) to execute **umount** *before* you remove the disk since not all files are necessarily written to disk at this time; you can lose your data that way!
If there are opened files on this device **umount** will try to write them otherwise it produces an error message. Furthermore you have to leave the directory you want to unmount.
If there are still users working (e. g.in a shell) on this path the filesystem can not be unmounted either.

19.13 The mtools

Especially for using MS-DOS filesystems either on disk or HD **mtools** (in package `mtools`, series `ap`)are at your disposal. All of those little programs try to emulate the corresponding DOS-original as good as possible. All **mtools** commands are named after their DOS pendant.

You can only use **mtools** commands if the disk (or HD) is *not* mounted!

DOS filenames are normally constructed by a device-letter followed by a colon, a subdirectory (evtl .), and the respective filename.

For separating only the `'\'` is used. For separating directories you can either use the `'/'` as well as the `'\'`.

If you use `'\'` or ☞*wildcards* they have to be put into quotation marks since otherwise they will be misinterpreted by the commandline interpreter (e. g.**bash**).

An asterisk `'*'` corresponds to DOS' `'*.*'`. Instead of using a `'/'` for passing parameters you have to use the `'-'`.

The standard device for **mtools** is `'A:'`. If another one should be used please change to it by typing **`mcd`**. Please do not forget to go back to the "root" directory of the device since otherwise no directory tree can be read.

Up to now the following commands are supported:

`mattrib`	Change DOS file attributes (`hidden`, `system`, etc.).
`mcd`	Change to another directory.
`mcopy`	Copy from DOS to Unix. Remember not to forget the target.
`mdel`	Deleting a DOS file.
`mdir`	Listing a DOS directory.
`mformat`	Assigning a DOS filesystem to a *low level-formatted* disk. (Low-level formatting is done via the command **fdformat**)
`mlabel`	Renaming a DOS device.
`mmd`	Creating a DOS subdirectory.
`mrd`	Deleting a DOS subdirectory.
`mread`	Reading (*low-level*) a DOS file into Unix.
`mren`	Renaming an existing DOS file.

Table 19.6: to be continued...

mtype	Shows contents of a DOS file.
mwrite	*Low-level* copy of a Unix file to a DOS filesystem.

Table 19.6: package `mtools`'s commands

Device `A:` is set to a 3.5" device, and device `B:` to 5.25". Diskformats that are supported are: 720 kB, 1.44 MB, or 360 kB, and 1.2 MB, respectively.

Those default settings can be changed in `/etc/mtools`. They are structured into one line as follows:

- Name of the drive (under DOS), e. g. `a:`
- Device file (Linux), e. g. `/dev/fd0`
- ID (12 floppy disk drives, 16 hard disk drives),
- Number of tracks, heads and sectors.

By changing those parameters you can use two 5.25" disk devices. You should not enter the same DOS device-letter or device-name twice.

19.14 Overview

cd directory	changes to given directory
cd ..	changes to parent directory
cd /directory	changes to directory
cd ~	Changes to home-directory of the user.
cp file_from file_to	copies files
ln [-s] source linkname	creates the symbolic link `linkname` in the current directory to `source`; `linkname` gives the path where the file can be found. Only symbolic links can be created across different filesystems. Even directories can be linked via "symbolic links" .
ls [directory]	lists contents of `directory` (short)
ls -l [directory]	lists contents of `directory` (long)
ls -a [directory]	lists even hidden files. (e. g.`.xinitrc` in your home-directory)
mkdir new_directory	creates a new `directory`
less file	pages through a file (page up using space, page down using b)
mv file_from file_to	moves or renames a file
rm file	deletes file (including links !);
rm -r directory	recursively deletes `directory`.
rmdir directory	deletes `directory` (if empty)

Table 19.7: Overview of the most important UNIX commands

`'find . -name' "file"`	searches all subdirectories of the actual directory for `file`.
`'find . -name' "*emil*"`	searches all files containing the string `'emil'`.
`'man command'`	gives you the manual page of **`command`**.
`'grep string files'`	searches all `files` for the given `'string'` which can, of course, contain ☞*regular expressions* as well (see section 19.7.2 or **`man regexp`**).

Table 19.8: Overview of "search" commands

19.15 And now?

The general directories for executables under Linux are:

- `/bin`
- `/sbin`
- `/usr/bin`
- `/usr/sbin`
- `/usr/X11R6/bin`

More commands are to be found under

The command **`man`** <**`command`**> gives you detailed information on the given command (see section 19.9) and programs (provided a manual page exists). Output is directed to standard output (which normally is your screen);

you can redirect output by means of *pipes* (`'|'`) to a printer or to a file.

This is achieved by entering:

```
newbie@earth: > man -t ls | lpr
```

Thematically sorted introductions about manual pages are found in hypertext-help system of your S.u.S.E. Linux system. Just enter **`help`**. Here other links to manual pages can also be found (hypertext).

19.16 Literature recommendations

Æ. Frisch *Essential System Administration* Sebastopol Calif., O'Reilly & Ass. 1991

Gilly, D., et.al. *UNIX in a nutshell*, Sebastopol, Calif., O'Reilly & Ass. 1992

H. Herold *Unix-Grundlagen*, Addison-Wesley 1992

Kofler, M. *Linux*, Bonn, Addison-Wesley 1995

G. Perlman *Unix For Software Developers*, Prentice Hall 1994

K. Pugh *Unix For The MS-DOS User*, Prentice-Hall 1994

G. Todino, J. Strang, J. Peek *Learning the UNIX operating system*, Sebastopol, Calif., O'Reilly & Associates 1993

S. Hetze, D. Hohndel, M. Müller, O. Kirch, et. al.
Linux Anwenderhandbuch, LunetIX Softfair 1994

Chapter 20

Emulators

20.1 The DOS-emulator

20.1.1 Introduction

Having a DOS-emulator which enables you to run MS-DOS- programs under Linux is not only of interest for former DOS-users.

This short introduction should give you the ability of installing the emulator on your system and give you an overview of how it can be configured. This project is still being developed and the frequency of the patches can grab hold with those of the kernel. We encourage you to experiment with it but please remember that some features are not yet implemented and others are still buggy.

An interesting way has been chosen to implement the DOS-emulator; some words of introduction are appropriate.

The DOS-emulator is not really an emulator but a BIOS-emulator which enables you to really "boot" DOS under Linux. That's why the user needs a DOS-license since the emulator just offers you a virtual machine. This enables you to run any DOS that runs on a PC under Linux as long as it doesn't switch to protected mode. Windows 3.0 only runs in real-mode.

Although the DOSEMU (since version 0.66.4) includes the FreeDOS kernel it has not reached the reliability and compatibility yet you might expect. Programs running under protected mode under DOS can be run if they make use of the DPMI. VCPI based DOS extender cannot be used up to now. Windows 4.1 applications use this VCPI whereas Windows 3.1 itself does not. That is why Windows 3.0 only runs in real mode under DOSEMU. The possibility of running the OS/2 Windows kernel (described in certain news groups and mailing lists) is only recommended under reservation (license problems).

> DOS-emulator make extensive use of kernel functions! Using it excessively can cause a freeze of the whole system! Don't use the emulator on really important machines, only use it if you're absolutely sure nothing can happen!

This description is based upon 2.0.30 and the DOS-emulator version 0.66.5 as they are on your CD. Due to the lightning-fast development of this package there might as well be another version on CD which was not available by time of going to press of this book.

20.1.2 Installation of binary package

If you have installed the binary-package via YaST you can skip the compiling section.

Please be aware that the DOS-emulator recently needed a function which was not available in kernel code, so there was a special module which supplied the necessary functions. Since kernel version 2.0.28 this is not needed anymore, as the kernel itself now provides this functionality. Furthermore it is no longer needed to recompile the DOS-emulator after a kernel reinstall.

Now please go on reading section configuration.

20.1.3 Compiling

Install the sources onto your system with YaST.

If you want to compile log in as `'root'` and change to the source directory of DOS-emulator (if you have installed the package using YaST this is `/usr/src/packages/dosemu/dosemu-0.64.3.1`).

You start compilation by entering:

```
root@earth: # cd /usr/src/packages/dosemu
root@earth:.../dosemu-0.66.5 # pkgmake extract dosemu-0.66.5
root@earth:.../dosemu-0.66.5 # pkgmake compile dosemu-0.66.5
root@earth:.../dosemu-0.66.5 # pkgmake install dosemu-0.66.5
```

Please keep in mind that some modules require quite an amount of memory (you should have at least 16-20 MB of swap; RAM and swap-partition). Compilation should be over after a couple of minutes.

Please note that not the version of the currently running kernel is of interest but the contents of `/usr/src/linux/include`, since `/usr/src/linux/include` defines the kernel interface during compilation.

So make sure that the DOS-emulator suits the actual installed sources.

> For this version at least kernel version 2.0.28. Older versions do not work!

Furthermore the programs **bison** and **flex** have to be installed.

20.1.4 Configuration

For being able to use the emulator it has to be configured; this is achieved by two central configuration files, `/etc/dosemu.conf` and `/etc/dosemu.users`. Personal settings for each user can be put into local `.dosrc`. Furthermore it is possible to configuration statements via commandline (option –I).

For your orientation there are are sample files `dosemu.conf` and `dosemu.users.*` under `/var/lib/dosemu/etc`. You should copy this file to `/etc` and create a link from `dosemu.users.easy` to `dosemu.users`.

```
root@earth: # cd /var/lib/dosemu/etc/
root@earth: # cp dosemu.config dosemu.users.* /etc
root@earth: # cd /etc
root@earth: # ln -sf dosemu.users.easy dosemu.users
```

Now you have a good position to start with. If you have installed the binary-package everything will already be in the proper place. It is recommended to leave this file untouched until you definitely know what you do (especially video-parameters).

Please do not use the `/usr/src/dosemu/Configure` script; the current version is faulty!

20.1.5 Booting DOS under Linux

For starting the DOS-emulator for the first time you will need a running DOS on one of your partitions or an bootable disk (apart from system only `command.com` is needed). Now `setup-hdimage` creates a bootable HDimage (virtual drive) of your DOS[1]

```
root@earth: # cd /var/lib/dosemu
root@earth: # ./setup-hdimage
```

First the script needs to know where to get DOS from. All partitions are checked for DOS and a list of suitable partitions is displayed.

```
checking you partitions, wait...
...done
I've seen the following bootable DOS partitions:

     /dev/hda1
     /dev/hdc1

Which one do you want to use as input?
Type in the full device name or ENTER to cancel
```

Screen output 20.1.1: DOSEMU searches for bootable DOS-partitions

Please enter the device of the partition (or `/dev/fd0` for floppy).

```
/dev/hda1
```

The script now starts detecting your DOS and recognizes the following types (manufacturer):

- MS-DOS (MicroSoft)
- Windows 95 (MicroSoft)
- PC-DOS (IBM)
- DR-DOS (Novell, Digital Research)
- OpenDos (Caldera)
- FreeDos (FreeDos Team)

If Windows 95 is selected `command.com` is taken from `\windows` otherwise from the root directory. If neither of the DOS versions mentioned above was found you are asked to enter the name of the DOS images `io.sys` and `msdos.sys`. This is shown in screen output 20.1.2.

Please enter, e. g.:

```
Your /dev/hda1 seems not to contain one of the DOSes that
I know.

A normal MSDOS system has io.sys and msdos.sys, these are
the files that are needed to boot your system. These
apparently are different for your system, but I need to
know their names.

What is the name of the file that corresponds IO.SYS?
```

Screen output 20.1.2: DOSEMU searches for `io.sys`

```
What is the name of the file that corresponds MSDOS.SYS?
```

Screen output 20.1.3: DOSEMU searches for `msdos.sys`

```
ibmbio.com
```

Please enter, e. g.:

```
ibmdos.com
```

Some DOS users use command interpreters (e. g. `4dos.com`) that differ from `command.com`, that is the background for the following question:

```
A normal MSDOS system has command.com as shell.
If you have a different one, please enter the name,
else just type ENTER
```

Screen output 20.1.4: DOSEMU searches for DOS-interpreters

So, if you want to use `command.com` just press ↵, otherwise enter the name of your favourite interpreter.

If the script is done you will have an preliminary hdimage file `/var/lib/dosemu/hdimage.first`; this file is a virtual disk for booting DOS. Configuration of DOS is rather simplistic and does only contain programs that belong to DOSEMU. Among others these are drivers for EMS CD-ROM, mice and many more. This disk is activated by:

```
disk  image "/var/lib/dosemu/hdimage.first"
```

in the file `/etc/dosemu.conf`.

Later on if the emulation runs smoothly and you got used to the configuration you might as well try drivers you are used to under DOS. For editing the configuration files `config.sys` and `autoexec.bat` on this virtual disk there is a small editor available (`ezedit`) for doing so.

[1] The package `mtools`, series `ap` has to be installed

You may even copy files from or to the hdimage by use of the **mtools** thus enables you to edit files under Linux. For accessing this hdimage use `N:`. This is already configured in `/etc/mtools.conf`. Please have a look at manpage of **`mtools`** (**`man mtools`**).

Now start the emulator by entering:

```
root@earth:/ # dos
```

After a while you should see the DOS bootprompt on your screen. Please be patient, this might take longer than on normal DOS machines (without emulator). Do not be afraid (or switch off the machine) if the BIOS messages of your graphics device appear; after all DOSEMU is an BIOS emulator and starts at a the lowest level.

If booting fails just switch to one of the other consoles by entering either of Ctrl + Alt + F1 to F6 and enter:

```
root@earth:/ # dosdebug
```

You should see the debugger screen output 20.1.5

```
DOSEMU Debugger V0.3 connected
- type ? to get help on commands -
Reading map file
1900 symbol(s) processed
highest address 0814ac90(_end)
system state: running
AX=0100 BX=00c0 CX=0000 DX=1814 SI=010d DI=a8b3 SP=0a96 BP=0000
DS=0133 ES=0133 FS=0000 GS=0000 FL=7246
CS:IP=fdc8:84f0       SS:SP=0133:0a96

fdc8:84f0 C3          ret
```

Screen output 20.1.5: The DOS debugger

Enter "kill" and both processes (debugger and the emulator) are interrupted. Just repeat the start-up process.

> The DOS-emulator expects a resolution of 25 lines by 80 columns. If you have started you Linux in another text-mode (e. g.`vga=extended`) perhaps beginning of lines are shifted diagonally. Especially while booting DOS 7.0 (Windows 95) the logo (in 320x300 graphic mode) destroys screen geometry set by DOS-emulator. In this case you can activate graphic mode in `/etc/dosemu.conf` later on leading to the correct graphic mode being set.

Leaving the DOSEMU is done by entering:

```
C:> exitemu
```

There also is a "rude" method for leaving the emulator: just enter Ctrl + Alt + Page ↓ .

> Be careful! This can lead to a blocked keyboard and screen. First try **exitemu**, then the debugger and as a last resort the Ctrl + Alt + Page ↓ method.

If you want to switch to another console you should use Ctrl + Alt + F1 to F6 as the "normal" switching Alt + F1 is only used for the emulation itself[2] not for switching consoles.

If invoking the DOS emulator should fail you might pass parameters so that it presents you with its debug messages. Just invoke it using:

```
root@earth:/ # dos -D+a 2> /tmp/debug
```

After the emulator has finished you may analyze the file `/tmp/debug` for finding out what went wrong. Normally this will not happen so that you can start adapting DOSEMU to your system.

20.1.6 Booting from a virtual drive

After having made your virtual drive bootable you again start the emulator with:

```
root@earth:/ # dos -D-a 2> /dev/null
```

If you change to drive D: (yes, you are right, this exists) you will notice that you find yourself in your Linux' root directory. Well, those long filenames look quite "funny".

Nevertheless, if you have mounted your real DOS drive (`C:`) to `/dos` under Linux you can access this drive at `D:\DOS` and start programs. Using an editor you can now edit `config.sys` and `autoexec.bat`. It is recommended to replace the virtual drive by the real-mounted drive in `autoexec.bat` (using **LREDIR**). You will need two identical copies of `autoexec.bat`; one on your virtual drive and one on your real drive otherwise **command.com** will not find the next line while switching. Here is an example:

```
c:\lredir c: linux\fs\dos
c:\lredir del d:
c:\lredir d: linux\fs\dos-drive-D
```

File contents 20.1.1: Accessing DOS-drives from the emulator

If you, e. g., want to map **/usr/dos** to **F:** you enter:

```
c:\lredir F: linux\fs\usr\dos
```

Normal / are to be replaced by DOS-typical backslashes ('\'). Filenames are case-insensitive. Now you have similar conditions as under "native" DOS: C: is C: and D: is D:; including the advantage that either Linux or DOS can access your drive.

[2] If `rawkeyboard` is set in the configuration file for using any combination while working with the emulator.

20.1.7 Virtual disks

An alternative for booting your drive is to virtualize the boot-floppy. Change to `/var/lib/dosemu` and enter the command:

```
root@earth:/var/lib/dosemu # dd if=/dev/fd0 of=diskimage
```

This writes a original copy of your disk to `diskimage`[3]

Now you have to "tell" the DOS-emulator that you plan to boot this virtual disk. In your `/etc/dosemu.conf` there are (directly after the greeting message) the following lines:

```
## comment out the below line, if you want to boot from a
## virtual floppy
#define vbootfloppy
```

File contents 20.1.2: Defining a virtual floppy

Just remove the '`#`' in front of the statement **`define vbootfloppy`**; now the necessary statements in `/etc/dosemu.conf` are activated.

For testing you may now start the emulator again. This should be possible without the inserted disk. Please do not hand `diskimage` to anybody since it contains a bootable DOS!

For using your boot-drive as a real drive you should run **`BOOTOFF`** under DOS. The counterpart **`BOOTON`** sets your virtual disk to on again.

20.1.8 User management and security precautions

From version 0.66.5 onwards the DOSEMU inherits an enhanced configuration system which improves security and provides quite some flexibility. Configuration variables may be checked in `/etc/dosemu.conf` which are set in `/etc/dosemu.users` as well. The file `/etc/dosemu.users.secure`, e. g., defines a more secure environment than the beginners configuration described above (`/etc/dosemu.users.easy`.

```
root c_all
nobody guest
guest guest
all restricted
```

File contents 20.1.3: Secure DOSEMU provided by `/etc/dosemu.users.secure`

The first word in every line defines a user (login); all other words in a line are considered configuration variables and are set. The user '`all`' refers not to a login but to *all* users. The first line that matches is used. So if you have set **`all`** at the bottom you can handle all other special users above.

[3] Command **dd** is especially useful if you want to create 1 : 1 copies of a disk.

Configuration statements are combined in classes (Prefix `'c_'`) that forbid setting not allowed statements *outside* of `dosemu.conf` (e. g. in `.dosrc` of the user).

You can decide what is allowed by defining the variables in `/etc/dosemu.users`. In our example the user `'root'` is allowed to perform everything, all other users are restricted. Further information can be found in `/usr/doc/packages/dosemu/README.config`.

For a normal user (besides root) to be able to access files on DOS partitions you have to make sure the permissions are already set while mounting the DOS partition. Since DOS does not know about permissions they are globally defined for the whole partition.

That is why S.u.S.E. Linux defined a group `'dosemu'` and creates the appropriate entries in `/etc/fstab` if you have assigned a DOS partition to a mountpoint in your Linux system. If you want to add additional partitions this should be done like:

```
/dev/hda1 /dosc msdos defaults,gid=30,umask=002 0 0
```

All users that have permissions set to access this partitions in `/etc/dosemu.users` have to be added to `'dosemu'` as well. This may be done by YaST in `'Group administration'` in `'Administrating the system'`.
After having entered the group name, `'dosemu'`, add the users into the list **`Group member list`**

20.1.9 Further configuration options

Besides linking a DOS partition as virtual drive it is possible to give the DOS-emulator read/write permissions to your drive. You have to uncomment the appropriate line in `/etc/dosemu.conf`. e. g.:

```
disk  partition "/dev/hda1"
```

Generally speaking the order in `/etc/dosemu.conf` is important. This is how drives are specified. (`C:`, `D:`, `E:`, etc.). Thus a bootable disk has to be listed first!

Use of one partition (`/dev/hda1`) is preferable to (`/dev/hda`, use of a wholedisk); if you want to access compressed drives (**Stacker**, **Dblspace**) you have to use “wholedisk”.

> Attention! Never use both methods of linking a DOS-partition at the same time! If you assign a DOS-partition as virtual drive you should not give the DOS-emulator read/write access to this partition since this undergoes kernel filesystems and a loss of data will be inevitable if you write to disk by two different methods!

If you have made it so far you should have got a feeling for the emulator's configuration and be able to adjust further configuration yourself. For an in-depth description of the emulator a book would be necessary so that we give only a few more “hints & tricks”.

By correct configuring video subsection at the beginning of `dosemu.conf` it is possible to run programs which need direct access to your graphic device;

even high-resolution modes and higher colour-depths are supported. Use of a mouse and connectivity to an existing Novell network are supplied. You can even start DOS under X11; start the emulator with **xdos**.

If other users (except 'root') should be allowed to use the DOS-emulator they have to be entered into /etc/dosemu.users. Enter one line per user; each user not mentioned here is not allowed to use the DOS-emulator.

Furthermore there are many features provided or implemented, such as use of EMS and XMS, direct access to certain hardware components, and the use of the normal printer spooling from within the emulator. Even dual monitor configurations are possible (for CAD-programs or debuggers). There is detailed information in /etc/dosemu.conf about all of these options. Furthermore you are directed to /usr/doc/dosemu for detailed information.

/var/lib/hdimage contains a complete DOS-image. It is possible that you will need a *new* version (e. g.if you plan to install a new DOS version). If this is the case just invoke the script setup-hdimage (make sure you made a backup of hdimage.first). All necessary files are still under /var/lib/dosemu/.

20.1.10 DEXE, directly executable DOS applications

From version 0.66.5 onwards special pre-configured DOS applications can be started directly. These are commands that contain a virtual drive where everything the DOS application needs is installed, but nothing more. Additionally there are configuration parameters (not accessible from DOS) that enables you to adapt the application.

One of the advantages is that it can be launched rather easily (the S.u.S.E. kernels recognize this format and launch the DOSEMU directly). Another is the security: The application runs isolated on the hdimage. Even viruses cannot burst into the frame of DEXE files. Thus your Linux is protected against such intruders. Well, packing of DEXE files is only for the experienced DOSEMU user. We only want to point out to the installed binary packed tetris clone **fallout**. It only runs under X and is invoked by entering:

```
root@earth: # fallout
```

If you have an (non-S.u.S.E.) kernel installed that has not been configured for **binfmt_misc** (section 13.4.4) you can invoke the DEXE as follows:

```
root@earth: # dosexec fallout
```

In /usr/doc/packages/dosemu/README.dexe there is additional information on DEXE. Here it is described how to create such files. We at S.u.S.E., LLC in cooperation with the DOSEMU team plan to establish an DEXE archive that will contain quite some of the good old DOS public-domain software that is not ported to Linux.

20.1.11 DOS-emulator and Windows 95

To put one thing straight: Under DOS-emulator you cannot run Windows 95 GUI, only DOS-mode (DOS 7.0). For not letting Windows 95 automatically start the GUI you should proceed as follows:

Under Windows 95 you click on ‘`Start`’, ‘`Shut down`’, ‘`Restart in MS-DOS-mode`’. If you have left the GUI and got presented the DOS prompt you should make `msdos.sys` (this is a text file) “visible” using the command **attrib** and replace **`BootGUI=1`** by **`BootGUI=0`**. Furthermore you should append a line **`BootDelay=0`**.

You can attach the key F8 to the DOSEMU to get into the Windows¸95 boot-menu. This is done as follows:

```
dos -D-a -I "keystroke \F8;"
```

Otherwise booting under DOS 7.0 using DOSEMU is exactly the same as described above.

20.1.12 STonX – An ATARI ST emulator

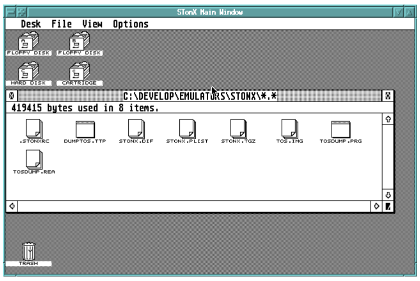


Figure 20.1: The ATARI ST desktop; TOS 2.06

Overview

STonX is a software emulator which runs on Unix workstations under the X-Window system.

Same as with the DOS-emulator STonX is designed to use already existing programs furthermore. The emulators concept is to enable — up to a certain degree — direct hardware access. The following system components are emulated:

- M68000 CPU
- 4 MB or 14 MB Main memory
- Blitter
- MFP (partially)
- IKBD (partially)
- ST-Shifter (STE partially)
- YM2149 Sound Chip
- Serial and parallel interfaces

Installation

Installing the STonX binary package is done automatically via YaST. Well, the emulator still needs the ATARI operating system[4], which is not available for free. If you have an ATARI with the adequate **TOS** you can use the

[4] Version 2.05 oder 2.06 (1.x only after recompiling, 3.x and 4.x do not run yet.

program **tosdump.prg** to save the contents of your ROM into a file. This program is located on our CD in `dosutils/atari`. Under `ftp://pc42-c603.uibk.ac.at/pub/TOS` there is a **TOS** 2.06. Please keep in mind that you are only allowed to use it if you have a registered original **TOS** 2.06.

If you invoke this program on an ATARI you will get a file `tos.img`. This file now has to be copied onto disk using **`mcopy`**[5]

```
root@earth: > ei -e stonx -d a:
```

reads this image and installs STonX accordingly. Assume that `tos.img` is located under `/tmp` this has to be done as:

```
root@earth: > ei -e stonx -p /tmp.
```

A detailed instruction on **ei** can be found in section 20.5.

The emulator itself now is installed. Now you can start STonX by typing **`stonx`**. On our CD additional ATARI software can be found. This is included in package `stonxsft` and is automatically installed to `/var/lib/stonx/software`. This software can be found on `C:` of the ATARI desktop.

Configuration

STonX has lots of options that can be configured. An overview on these options is listed in table 20.1. An example:

```
newbie@earth: > stonx -disk a:file -color -noaudio
```

would use `/dev/fd0` (floppy a) as drive A:, boot in colour mode and not emulate ATARIs sound.

-q	Quiet (n output)
-size <W>x<H>	emulated resolution: Width x Heigth
-disk <X>:<file>	Use file <file> as drive <X>:
-fs <X>:<Dir>	Use directory <Dir> as drive <X>:
-colo[u]r	Colour emulation
-mono	Monochrome emulation
-usec <X>	<X> sets the microsecond interval timer
-vbl <X>	Sets <X> * usec VBL Interval
-timer-c <X>	Sets <X> * usec / 2 Timer C Interval
-refresh <X>	Sets <X> * VBL refresh Interval
-para <file>	<file> as parallel interface
-serial <file>	<file> as serial interface
-noaudio	No sound emulation
-noshm	Do not use XShm extensions
-warmboot	Simulated warm reboot
-vdi	Enable Xlib VDI drivers
-private	Use own colourmap
-help	This List :-)

Table 20.1: List of valid Commandline parameter for STonX

[5] IN package package `mtools`

You can also add all these parameters to `.stonxrc` in your home directory. This file is automatically created for the sample user. Please keep in mind that the “-” prefixes are no longer needed (so, only fs instead of -fs).

> The `size` parameter is not fully implemented yet. At first it is recommended only to try 800x600 resolution. In `.stonxrc` `noaudio` and `noshm` are set; if you already have a kernel with sound support you can remove noaudio. For test purposes you can remove the `noshm` as well as this speeds up the emulator; if the system crashes, well, then insert it again!

Also be aware that STonX only runs with 8bit colour depth and under X11. More documentation on the package can be found in `/usr/doc/packages/stonx`. Brand-new information on STonX actual state of development can be get via the WWW from `http://www.complang.tuwien.ac.at/nino/stemu.html`.

20.1.13 UAE – Unix Amiga Emulator

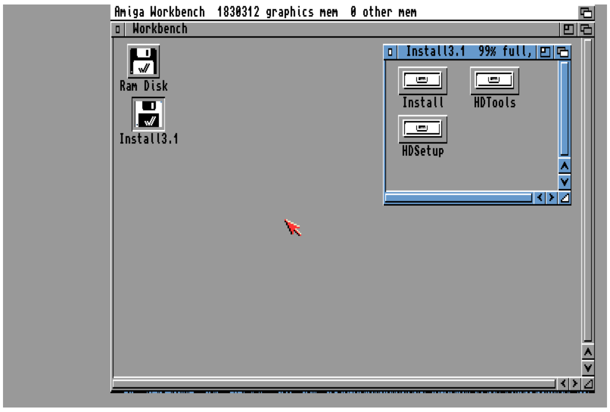


Figure 20.2: Amiga Workbench

Overview

UAE emulates an A5000 almost perfect. All kickstart ROM versions are supported, provided they have not been compiled for use on a 68020 CPU. The following system components are supported:

- max 11,8 MB RAM (2 MB Chip- and up to 9,5 MB FastRAM)
- complete OCS functionality.
- partially ECS.

Installation

For running the Amiga emulator you need a Kickstart ROM as file `kick.rom`. For copyright reasons this file can not be shipped together with S.u.S.E. Linux. It has to be created using the tool **transrom**. This program can be found on CD in `dosutils/amiga`. Comfortable installation is provided by **ei**.

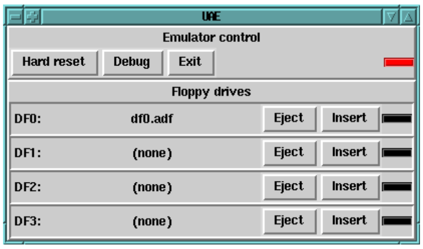


Figure 20.3: GUI of UAE for controlling media.

Configuration

There are many configuration options or UAE. table 20.2 gives you an overview.

-h	Overview of parameters.
-m LW:<mount>	Mount filesystem at <mount> as AmigaDOS drive LW:
-M LW:<mount>	like -m, but read-only
-a	Do not mount hard disk file automatically; Comes handy for test purposes.
-s n	Emulate n*256 KB slow-RAM at 0xC00000
-c n	Emulate n*512 KB chip-RAM at 0x000000
-F n	Emulate n*1 MB fast-RAM at 0x200000
-f n	Set number of frames.
-D	Launch the included debugger
-i	Announce invalid memory access
-[0123] <file>	Use <file> instead of `df[0123].adf` as disk.
-r file	Use <file> as ROM-image instead of `kick.rom`.
-J	Emulate joystick via number block.
-S	Disable sound-support (if configured)
-x	Use Cross-hair cursor
-l lang	Keyboard language DE, SE, US, FR or IT
-p command	Command for sending output to the printer
-I device	Name of serial interface (/dev/ttyS1)
-d mode	Select resolution (see table 20.3).
-H mode	Number of colours.
-C	Use correct proportions for pages.

Table 20.2: List of valid commandline parameters for UAE

Valid resolutions	0 (320x200); 1 (320x240); 2 (320x400);
	3 (640x480); 4 (800x600)
Valid colour modes	0 (256 colours); 1 (32768 colours);
	2 (65536 colours)
	3 (256 colours using dither)
	4 (16 colours dithered); 5 (16 millions of colours)

Table 20.3: List of valid graphic modes for UAE

> Please be aware that UAE ignores some resolution/colours options since the video hardware is not designed for any combinations.

20.2 Z80 emulators

Z80 of **Zilog** is the best-selling 8 bit micro processor of the world .

On June, 3rd 1996 the ZX80 celebrated its 20th birthday which makes it more than a methusalem in computer history. Anyway it is still used nowadays in a couple of devices and even new realisations of products are planned with the **Z80**.

In the mid-eighties it dominated the market of 8 bit home computers and is still used nowadays in “toys” as the gameboy. Emulator based on this system on S.u.S.E. Linux are:

- ZX81 (see section 20.2.1)
- ZX Spectrum (see section 20.2.2)
- Coleco Vision (see section 20.2.3)
- Gameboy (see section 20.4.2)

20.2.1 Z81 – Sinclair ZX81 emulator

The Sinclair ZX81 (in the US Timex TS1000) is the predecessor of Sinclair ZX Spectrum. In 1981 the ZX81 Sinclair was released either assembled or as kit. Its look resembles an ancient (prehistoric) computer species: 1 KB RAM, foil-keyboard, black/white block graphics. Nevertheless this machine initiated the “home computer area” and lots of programs were created for this machine.

These programs now can be viewed under Linux as well. You can also enter your own programs or try the early ZX80 programming (using its assembler programs). The underlying operating systems has kindly been given to us by **Amstrad**. Amstrad still holds the copyright for this software.

Figure 20.4: One of over 4000 games for ZX Spectrum

20.2.2 XZX – ZX Spectrum Emulator

Overview

XZX is the emulator for the ZX Spectrum computer of **Sinclair**. A Sinclair ZX80 emulator. A Spectrum 48 K, 128 K and a +3 are emulated. The following system components are supported either:

- Z80 CPU
- 16 K, 48 K, 128 K RAM
- Interface 1
- Sound (Beep)
- Kempston Joystick

Installation

XZz needs — dependent on its mode — certain ROM files. These are included, thanks to the permission by **Amstrad**, in the package. Anyway **Amstrad** keeps the copyright of these files.

On `http://www.ludd.luth.se/users/alfredo/spectrum.html` there are some of the "game classics" in SNA format. Please be aware that those files might be copyrighted and only may be used if you are a registered customer of their product.

> The files on this FTP server are protected by copyright! Any use without permission of the manufacturer and/or possession of the original software is considered illegal!

Configuration

-help	this table

Table 20.4: to be continued...

-version	actual version
-visual <visual>	Use given X-display <visual>
-font <fontname>	Use given X Font
-geometry <g.spec.>	Give window position
-romName <fname>	Use <fname> as 48K ROM image
-rom128[01] <fname>	Use <fname> as 128K ROM image
-rompl3[0-3] <fname>	Use <fname> as +3 ROM image
-libDir <path>	Assign library paths
-mitshm <bool>	Use MIT-SHM
-flashing <bool>	Activate flashing [true]
-rrShm <n>	Refresh screen every <n> interrupt
-rrNoshm <n>	Refresh screen every <n> interrupt
-mono	Use monochrome
-private	Private colour map
-intStyle [signal—counter]	Art of interrupt implementation
-iFreq <n>	Interrupts every n commands
-kempstUp <keysym>	Kempston up key [q]
-kempstDown <keysym>	Kempston down key [a]
-kempstLeft <keysym>	Kempston left key [o]
-kempstRight <keysym>	Kempston right key [p]
-kempstFire <keysym>	Kempston fire key [space]
-sound <bool>	Sound on/off [false]
-if1 <bool>	Interface 1 on/off [true]
-if1Rom <path>	Give interface 1 ROM
-crlf <bool>	CR/LF translation for RS232 input
-strcr <bool>	Remove CR at RS232 output
-m<n> <path>	Microdrive cartridge file in <n>

Table 20.4: List of valid commandline parameters for XZX

20.2.3 Adamem – Coleco Vision/ADAM Emulator

The company **Coleco** and their products **ColecoADAM** and **ColecoVision** presented a game console as well as a PC like successor (using 64 KB RAM and a Z80 processor). The Linux emulator understands both modes

Even nowadays the "Adam" and its 5.25" disks and 20 - 40 MB hard disks is used by freaks. Since its introduction in 1981 lots of programs were written that now can be used under Linux.

Installation

Adamem needs — depending on its mode — different ROM files which normally are included in the source package. Since copyright of these programs could not be figured out you have to get those from the original package.**ei** is a great help provided you have internet connectivity. Here the archive with the original sources from `http://www.komkon.org/\~{}dekogel/adamem`.

Figure 20.5: One of thousands of games available for the Coleco System

`html`. The necessary ROM files are extracted and installed on your system. This process is launched by entering

```
root@earth:/root > ei -e adamem -s
```

Software

`\footnotesizehttp://www.sponsor.net/\~{}gchance/ColecovisionStuff/ Docs/docs.html` gives an overview of available software for this system.

20.3 CBM-8Bit-Emulation – Vice

Commodore Business Machines (CBM) has produces several 6 bit machines since the early eighties. The most famous one certainly is the Commodore C64. Now, **vice** is four emulators in one. It emulates C64, C128, VIC20 and Commodore PET.

20.3.1 Vice-installation

Of course you need the appropriate ROM for the different types. These ROMs can be found on `ftp://sunsite.unc.edu/pub/Linux/system/emulators/ commodore` or a mirror. There the `vice-roms.tar.gz` can be found. **ei** expects the file for installation.

Please be aware that the precompiled versions only run under X and only support a 8 bpp colour depth. The different options can be gathered by

8 bit colour depth

```
newbie@earth:/home/newbie > <Emu> -help
```

All options can be abbreviated as long as they remain unique. Meaning of all these options can be looked up in package `vice`'s documentation which lies under `/usr/doc/packages/vice`. On an average machine these emulators reach 100% of the original machines. If required you can speed them up (see figure 20.6).

Figure 20.6: Elite on C64 running in "mach 2" :-)

20.3.2 C64-emulation

After a correct installation of package `vice` and installation of ROMs (see above) the program can be invoked by **`x64`**. The emulator is started using

```
newbie@earth:/home/newbie > x64
```

The screen shot shown under figure 20.6 has been achieved as follows: You have to get a "disk-file". This may be done by manually transferring the original disk (or cassette; see vice documentation) or by access on certain internet pages. This file should be app. 174 KB in size[6]. The start command for achieving the above result is:

```
newbie@earth:/home/newbie > x64 -floppy ELITE.D64 -speed 200
```

20.3.3 C128-emulation

The C128 was the C64 compatible successor. The emulator emulates "only" the C128 specific features, since there is the x64 if you want to emulate an C64. Launching **`GO 64`** therefore is not possible. The emulator is started via

```
newbie@earth:/home/newbie > x128
```

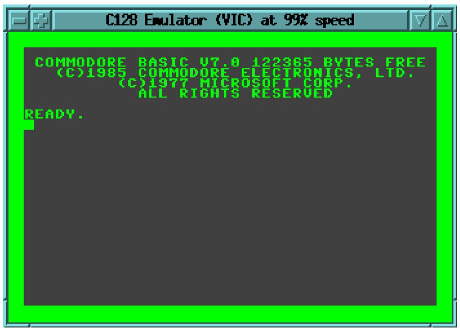


Figure 20.7: Message screen of C128. You must see the copyright...

There is a wide variety of additional information, programs and data that can be found in the internet. Some URLs of interest:

[6] mostly with an .D64 extension

- `http://www.cucug.org/c64128.html`
  The C64/128 Web Directory. The most well-known pate on C64 and C128. Contains numerous lins to other pages.
- `http://www.hut.fi/\~{}msmakela/cbm/`
  The Commodore 8-bit WWW server. Becomes the main source for information on this architecture.
- `ftp://ftp.funet.fi/pub/cbm`
  Finnish FTP server with “tons” of software for **CBM**’s 8 bit computers.

20.3.4 VIC-20-emulation

This machine was the **CBM**’s first real “home computer” since all other machines (PET) were designed for commercial purposes and were not affordable to the public. The VIC-20 is the predecessor of C64 and C16. A vast amount of information on this model can be found under `http://www.mds.mdh.se/\~{}dal95acn/vic20.html`. The emulator is launched as

```
newbie@earth:/home/newbie > xvic
```

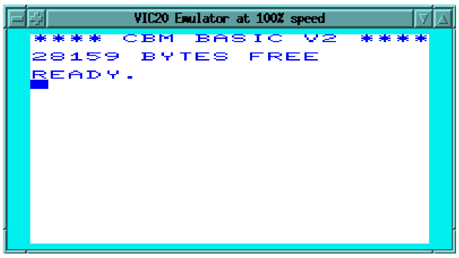


Figure 20.8: Welcome screen of VIC20. 80 characters /line were a rarity on PCs .

20.3.5 PET-emulation

CBM’s first computer[7]. For lots of computer freaks this was the first contact with a PC. Further information on this type can be obtained under `http://www.ncsc.dni.us/fun/user/tcc/cmuseum/pet.htm`. The emulator is invoked using

```
newbie@earth:/home/newbie > xpet
```

It emulates 40 as well as 80 characters (see options **`xpet -help`**).

20.4 Emulators for game consoles

20.4.1 Stella – ATARI VCS 2600 emulator

Before **ATARI** started creating “real” computers they were famous for their game consoles. A quite early and well-known type was **Video Console System** VCS 2600 which came out in 1977. All games for this console were only available as ROM cartridges. The successors were 5200 und 7800; at the moment there is no emulation availabe for these machines.

[7] if KIM-1 does not count

Figure 20.9: One of the multiple games for the VCS 2600 system

Installation

The original console does not contain an operating system. After installation of the emulator it can be started immediately. Under `/usr/lib/stella/ROMS` there is a little test program (`TEST.BIN`).

by entering

```
newbie@earth:/home/newbie > cd /usr/lib/stella/ROMS
newbie@earth:/usr/lib/stella/ROMS > xstella TEST.BIN
```

it can be launched. It shows psychedelic waves `:-)`

Software

There are more than 450 games that run under the VCS 2600, which can be run with Stella. These ROM images can also be found in the internet.

These ROM images are protected by copyright. Using withoud permission is regarded illegal!

ei knows all the URLs of software that belong to this emulator. Provided the URL still exists — and you are connected to the internet — the software can get by invoking **ei** like:

```
root@earth:/ > ei -e stella -s
```

We have to point out that you *have* to delete files if you do not own the original files!

20.4.2 VGB – A virtual gameboy

Gameboy of **Nintendo** is not a very powerful but widely distributed game console. Now there is an emulator available for Linux.

This emulator provides high comatibility. According to the author any known game is supported. **Vgb** does not provide network and sound support up to now. Well, the first is actually being worked on. Interesting pages in connection with this program are e. g.`http://www.freeflight.com/fms/VGB/` as well as a beta Linux port (with sound and joystick-Support) on `http://www.geocities.com/SiliconValley/Vista/8340/`.

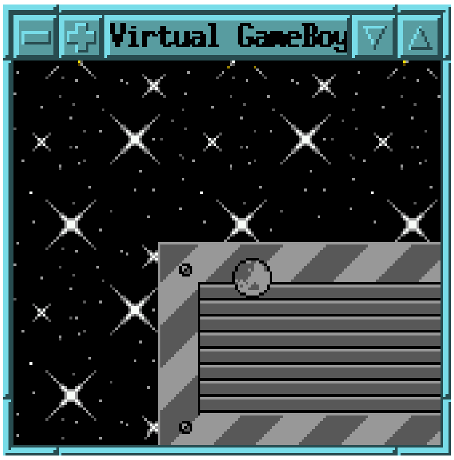


Figure 20.10: A gameboy that runs under Linux

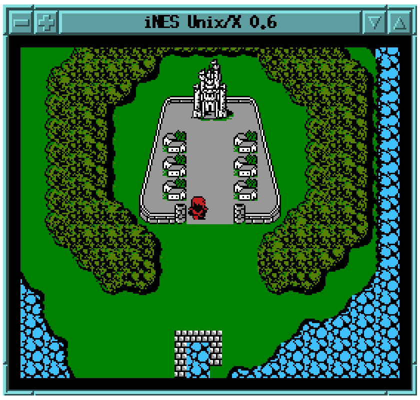


Figure 20.11: Nintendos Entertainment System under Linux

20.4.3 iNES – Nintendo Entertainment System Emulator

Nintendo Entertainment System (NES) is a little game console for home use and is designed for attaching it to a TV. iNES provides this for Linux in a window . . .

The emulator can be launched immediately. This means that you do not need an operating system. Only the ROM images (in form f *.NES files) have to be there.

iNES is invoked as

```
newbie@earth:/home/newbie > ines <filename>
```

Here <filename> is the filename of the *.NES file; the games ROM image.

20.4.4 Where to get ROMs?

This is the most common question. Furthermore it also is the most inconvenient question that can be asked by a user of emulator software. It does not matter where you ask, be it the S.u.S.E. support or a news-group. Do not forget that ROMs are proteced by copyright and nobody wants to commit a crime by sending you copies. Here some information on how to proceed:

- Internet search engines
  The first (and often the only) choice for finding files. A clever request to one of the big engines (**Lycos**, **Excite** etc.) often gives a useful result (e. g. Kickstart AND Amiga). A couple of search engines is given on `http://home.netscape.com/escapes/search/`.
- Suitable WWW server
  We already gave some hints where certain ROM files can be found. Some web-servers are a good start for this search. The acknoledged Node 99 can be found on `http://206.101.78.248/node99/`.

What you should *avoid*; what you should never ever do:

- Ask the author of the emulator for ROMs. The authors will never fulfill this request and (at best) ignore your question. Some authors announce you to an "abusers" list where your electronic mail address is presented to public. It already happened that authors stopped writing an emulation *because* thousands of people pledged them for a ROM.
- Ask S.u.S.E. support for ROMs.

20.5 ei – Installation of emulator-firmware

The instalation script **ei** serves as an interface for installing the needed ROM images of the emulators. These images normally contain the operating systems of the emulated machine. Since we can not provide these ROM s due to copyright restriction you have to get these files your own.

Most of the data can be found onthe internet. We point out that this is only for those customers that have a registered license of the software (because it is not easy to port 15 year old software from proprietary machines onto modern machines.)

This documentatin as well as **ei** (emu-install) should help you installing everything needed easily.

20.5.1 Invoking ei

Whenever you launch **ei** without parameters or with the **`-h`** option this message appears:

The options in detail:

- **`ei -e uae -d a:`**
  Expects `kick.rom` in drive a: incl. the AMIGA Kickstart ROM.
- **`ei -e stonx -p /tmp`**
  Expects ATARI TOS (`tos.img`) in `/tmp`. Please keep in mind that STonX has been precompiled for versions 2.0.x.
- **`ei -e vice -d a:`**
  Expects `vice.tgz` in floppy drive A: which contains ROM files for CB machines.

```
Emuinstall - an installation program for various emulators under linux
           Copyright (c) 1996 S.u.S.E GmbH, Fuerth Germany.
               Written by Richard Jelinek (rj@suse.de)

Arguments:
ei [-h] -e MODE -p EI_PATH | -d EI_DISK
  -d,--disk     use disk instead of list of paths.
                DISK is either a: or b:
  -h,--help     this help
  -e,--emu      which ROM to install. Possible values for MODE:
                amiga, uae, atarist, stonx, commodore, vice
                spectrum, xzx, all
  -p|--path     directory where to search for ROM-images
```

Screen output 20.5.1: ei help

- **`ei -e adamem -s`**
  If there is a connection to the internet this gets the ROMs needed for the Coleco-emulator.

Chapter 21

The vi editor

Using **vi** is a little difficult to get accustomed to. It is preferred to other editors since it is installed on every standard Unix system and comes along with Linux' standard installation. On the other hand its use is well defined, thus this can't lead to misunderstandings.

This little help shall enable you to edit your configuration files using **vi**. See [Lamb, 1990].

Concept:

There are three modes available in **vi**:

- Command mode
  Every pressed key is interpreted as part of a command.
- Insert mode
  Pressed keys are interpreted as text.
- Last line mode
  For more complex commands that have to be edited in the last line.

The most important commands of the *command mode* are:

`i`	Changes to input mode (Characters are entered at the actual cursor position)
`a`	Changes to input mode (Characters are entered after the actual cursor position)
`A`	Changes to input mode (Characters are appended at the end of line)
`R`	Changes to input mode (overwrites old text)
`r`	Changes to input mode for overwriting one character.
`s`	Changes to input mode (character the cursor is own will be overwritten)
`C`	Changes to input mode (rest of line is replaced by the new text.)

Table 21.1: to be continued...

`‘o’`	Changes to input mode (*after* the actual line a new line is inserted)
`‘O’`	Changes to input mode (*before* the current line a new line is inserted)
`‘x’`	Deletes the current character
`‘dd’`	Deletes current line
`‘dw’`	Deletes to the end of the current word
`‘cw’`	Changes to input mode (rest of the actual word is overwritten by the input)
`‘u’`	Undoes the last command
`‘J’`	Appends line to follow actual line
`‘.’`	Repeats the last command
`‘:’`	Changes to last line mode

Table 21.1: Simple commands of **vi**

All commands can be preceeded by a number; this number sets how many times the given command should be executed; thus entering `‘3dw’` deletes three words at once and `‘10x’` deletes 10 characters, `‘20dd’` deletes 20 lines respectively.

The most important commands in *last line mode* are:

`‘:q!’`	quits **vi** without saving changes
`‘:w [file]’`	saves in [file]
`‘:x’`	saves changed file and quits
`‘:e [file]’`	edits [file]
`‘:u’`	undoes the last editing command

Table 21.2: Complex commands of **vi**

Typing ESC in input mode changes to command mode.

Chapter **22**

emacs – more than just an editor

The choice of an editor can become a religious question. There are mainly two opposite groups: the stubborn **vi** users and the increasing amount of **emacs** adherents. Since **vi** was created when there were no graphical interfaces at all it is mostly used by users that keep sticking to this age of computing. Well, some vi addicts nowadays confess that **emacs** is really easy o handle (which for itself is nothing special whatsoever) and much more powerful.

22.1 Basics

That **emacs** is more than a simple editor is proven by people that execute any task available on their machine with this program. You can compile, read and maintain mails, use ftp very easily, read news with **emacs** and many more. There are extensive commands at your disposition (search and replace using regular expressions). Moreover **emacs** can be programmed using its own language (Elisp: a Lisp dialect). Quite an amount of **emacs** itself is programmed in Lisp. Thus you can easily adapt **emacs** to suit your personal needs. Due to this versatility there are plenty of **emacs** modes for any language which makes handling of them very easy.

After all this praise we do not want to hide the disadvantages of this famous tool. The system resources it eats are quite big (the **emacs** binary itself needs 1.3 MB). Since a great amount has been programmed in Lisp it is not as fast as other (smaller) editors. The first item gas been removed in the meantime since there are fast and small **emacs** clones (e. g.**jove** and **jed**). For those who just want to make a small change to a configuration file, but do not want to miss **emacs** handling, they can use one of the smaller ones instead without having to load **GNU**'s monster **emacs**.

22.1.1 Invoking and handling

emacs is invoked (under X) using:

```
user@earth:/home/user > emacs
```

If you supply **emacs** with one or more names at start-up time these files will all be loaded into different buffers. Most of the handling in **emacs** is done via keyboard. The most important commands can alternatively be started via mouse. We just cover keyboard handling since mouse handling should be self-explanatory and because most users do not use the mouse. It is rather tedious to change from mouse to keyboard all the time at writing. After launching **emacs** the copyright messages and some helpful commands are presented. The second most line is the status line. Here the name of the buffer (`*scratch*`, since the buffer does not have a name yet) and the *mode* where the buffer is in are presented (in this case this is *Lisp Interaction*). The line below the status line is the *mini-buffer* where commands are entered.

22.1.2 Entering commands

Most **emacs** commands are launched via the Ctrl or Esc key. In Emacs they are spelled as **`C-h`**; this means simultaneously pressing Ctrl and h . **`M-h`** means press and release the Esc key followed by h . After having pressed and released Esc the message `"ESC"` appears in the mini-buffer. If X-Windows has been configured to assign the left Alt key to **`ESC`** commands as **`M-h`** can be entered as Alt + h too.

22.2 Basic commands

22.2.1 On-line help

The most important command for the **emacs** novice is **`C-h`**. This is for starting the on-line help system. Entering **`?`** gives you a list of available options that can be scrolled using space. For getting accustomed to **emacs** it is recommended to call **`help-with-tutorial`** (by entering **`t`**). This is an interactive tutorial showing you how to handle everything using examples. The tutorial file is copied and can be changed in any way by the novice user to learn handling. After having worked through the tutorial you should know all the basic commands of **emacs**.

22.2.2 Cancel and quit

The most important command besides starting is ending: **`C-x C-c`**. For cancelling commands (e. g.if an entry is requested in the mini-buffer and the cursor automatically jumps there) you use **`C-g`**. Repeatedly pressing **`C-g`** and **`C-x C-c`** should enable you to leave **emacs**.

22.2.3 Undo

Command **`C-x u`** (undo) lets you undo the last command. Pressing **`C-x u`** repeatedly lets you undo step by step. The variable <undo-limit> sets how many bytes can be undone (default is 20000).

22.2.4 Cursor control, kill and yank

Those of you who read the manual thoroughly might have recognized that any cursor movement can be performed via keyboard commands. This of course is still useful on terminals without cursor keys. Anyway mainly the cursor is moved around by the mouse or the keyboard. The keys ⟵ (Backspace), Del (Delete) oder **`C-d`** also can be used for deleting single characters. The expert might press **`C-d`** faster than Del . This also is useful if you are handling emacs via a network connections where key-mapping is different (or not fully configured). `C-v` and `M-v` or Page ↑ and Page ↓ can be used for page-up and page-down .

`C-k` deletes all characters to the right of the cursor and saves them in the *kill-ring*. They can be reentered anywhere by `C-y` (yank). Frequently pressing `C-k` saves all strings in the kill-ring. By pressing `C-y` one can insert the text back again.

`M-w` saves all text between the mark and the cursor-position into the kill-ring without deleting the text.

22.3 Loading and saving texts

`C-x C-f` (find) loads a file into a buffer of its own. The cursor jumps to the mini-buffer expecting a file name. If a non-existent file is entered an empty buffer is opened. If you save the buffer this file is created. If you want to load an existing file use the ⟼ key for globbing.

`C-x i` (insert) lets you insert a file into the current buffer at the current cursor position. Files are saved using `C-x C-s` (save). If you want to save the file under another name type `C-x C-w` (write).

22.3.1 Backup-files

emacs automatically creates a backup-file. If the file is deleted by accident or another disaster happens you can rely to this file. The backup-file has got exactly the same name but with an `'~'` attached at the end. Thus the backup name of `filename` is `filename~`. So do not be astonished if every file exists twice while working with **emacs**.

22.4 Buffer and windows

Any text in **emacs** is in a buffer of its own. You can work with as many texts as you like at the same time. `C-x b` lets you jump between buffers. `C-x C-b` shows a list of all existing buffers.

In **emacs** you can use “text”-windows. Each of these windows contains a buffer. `C-x 2` splits the window horizontally. So you can edit or view different (or the same) buffers at the same time. Quite helpful is `C-x o` (other) which lets you jump to the other window. `C-x 3` splits the window horizontally. `C-x 1` goes back to one window and `C-x 0` removes the current window without closing the buffer.

22.5 Search and replace

`C-s` (search) lets you search a string. After having entered the command itself all other keystrokes are regarded as text to be searched. As you enter the letter the cursor jumps to the next occurrence of the given string. Pressing `C-s` (search) again jumps to the next suitable string. For deleting reversely you enter `C-r` (reverse). This search is an incremental search. It even is possible to delete characters from the search string. The search can be cancelled by either `C-g` or a cursor key.

Start of the search is marked for you. So if you want to jump back to where you were before just enter **C-x C-x**.

Search and replace is done via the commands **M-x replace-string**, **M-x query-replace**, **M-%** respectively. The latter differs in so far as it lets you decide whether you want to replace or not (so if you are unsure you should use this command). Both commands are invoked via **M-x**. As you have seen some commands have key combinations as well. After having entered **M-x** the cursor jumps in the mini-buffer and expects the name of an executable command. Most interactive commands can be launched that way either. Quite often it is useful to use **M-x** for finding out whether a certain command exists. Globbing is done, same as with files, by ⟼.

22.6 Keyboard macros

By means of keyboard macros key-combinations can be saved and executed any time. Making (intelligent) use of macros can save you quite an amount of (programming) time. To start recording a macro type **C-x (** Everything entered after this command is saved into the macro. **C-x)** lets you stop recording. and **C-g** cancels it.

22.6.1 Executing and saving macros

Macros that have been defined can be executed by **C-x e** (execute). This command always launches the macro that was defined last. If you want to use more than one macro you have to give them names. **M-x name-last-kbd-macro** gives a name to the last macro; **M-x** <**macroname**> lets you execute this macro.

For further use it is necessary to save these macros. For saving macros you use **insert-kbd-macro**. This inserts the macro definition at the cursor position. It is recommended to create a file of its own for macros; all macro definitions are written in Elisp. This Elisp file can be loaded by **M-x load-file**. Immediately after having load this file you are able to use your macros. The other idea is to save it into ~/.emacs[1]. Thus these programs are available immediately.

A quite useful combination in connection with macros is **C-u** <**number**> for repeating a command <number>-times. For example **C-u 5 l** writes the letter 'l' 5 times. For repeating it 20 times it must be **C-u 20 l**.

> On S.u.S.E. Linux there is a ~/.gnu-emacs and a ~/.xemacs-options besides the usual ~/.emacs. This unfortunately is necessary as the Lisp syntax of GNU-**emacs** differs from X-**emacs** *and* the precompiled lisp files of the two editors have an incompatible binary format. So if there is no ~/.gnu-emacs in your **HOME** after an update you should save your ~/.emacs and copy /etc/skel/.emacs and /etc/skel/.gnu-emacs to your **HOME**.

[1] `$\sim$/.emacs` and files loaded by it contains emacs commands that are launched when emacs is invoked.

22.7 Info

Info is an hypertext documentation system integrated into **emacs**. It is invoked by entering **`M-x info`** or **`C-h i`**. **`h`** now gives you help on info. For selecting an item in **`info`** you have to move the cursor to the corresponding line and press [↵]. **`m`** is for directly selecting an item. Alternatively it can be selected by **`m`** and entry of the menu item. **`u`** moves you up one node upwards, **`p`** shows the previous node, **`l`** the last (which not necessarily has to be the previous). Selecting an reference node and pressing [↵] moves you that point. **`d`** always brings you back to the toplevel menu and last but not least **`q`** quits **`info`**. Many **GNU**-programs (including **emacs** and **info**) are documented in info.

22.7.1 More emacs

In *info* there are lots of functions and settings described for **emacs** which can not be discussed in this book. For those who are interested just make use of the real good documentation in **`info`**. We recommend having a look at the following topics: *Dired*, *VM*, *GNUS* and *Ange-FTP*.

Furthermore in `/usr/share/emacs/19.34/etc/refcard.tex.ps` there is a **emacs** reference-card that might be of use to you.

22.8 Literature

Some books on **emacs**:

Cameron, D. / Rosenblatt, B. *Learning GNU Emacs*, Sebastopol, Cal., O'Reilly & Associates 1991

Schoonhover, M. / Bowie, J. *GNU Emacs*, Reading, Mass., Addison-Wesley 1992

Grieger, W. *Wer hat Angst vorm Emacs?*, Bonn, Addison-Wesley 1994

22.9 Command overview (selection)

Stop, Undo, Redo	
`C-g`	Cancels a command
`C-x u`	Undoes the last command
`C-u` <**`number`**> <**`command`**>	Repeats <command> <number>-times
Help	
`C-h ?`	Help menu
`M-x info`	Hypertext documentation system
`C-h i`	Hypertext documentation system
`M-x apropos`	Online search a keyword

Table 22.1: to be continued...

`C-h k <key>`	Explains the function of <key>
Cursor movement	
`C-v`	Page down
`M-v`	Page up
`C-M-v`	Other window page up
`M-f`	Word forward
`M-b`	Wort backward
`M-<`	Cursor at beginning of text
`M->`	Cursor at end of text
`C-a`	Cursor at beginning of line
`C-e`	Cursor at end of line
`M-a`	Cursor at beginning of paragraph
`M-e`	Cursor at end of paragraph
`M-m`	Cursor onto first character that is no white space
`C-x C-x`	Change cursor and mark
`C-l`	Centre cursor position in window
Deleting and moving	
`C-k`	Deletes to line end
`C-y`	Inserts deleted text
`C-` space	sets a mark
`C-x h`	Marks the whole buffer
`C-w`	Deletes from cursor to mark
`M-w`	Deletes from cursor to mark into kill-ring
`M-` space	Removes blanks + Tabs except for one
`C-x C-o`	Deletes blank lines except for one
`M-d`	Deletes to end of word
`M-` Entf	Deletes to beginning of word
`M-k`	Deletes to end of phrase
`M-z <character>`	Deletes to (including) <character>
File commands	
`C-x C-f`	Load file
`C-x C-r`	Load file *read only*
`C-x C-s`	Save file
`C-x C-w`	Save file under another name
`C-x i`	Insert file into text
Search and replace	
`C-s`	Search forwards
`C-r`	Search backwards

Table 22.1: to be continued...

`C-x C-x`	Go back where search started
`M-x replace-string`	Replace string
`M-x query-replace`	Query replacement of string
`M-%`	Query replacement of string
Buffer and windows	
`C-x b`	Changes to another buffer
`C-x C-b`	Shows a list of all buffers
`C-x k`	Removes current buffer
`C-x C-q`	*read only* switch for current buffer
`C-x 2`	Splits current window horizontally
`C-x 3`	Splits current window vertically
`C-x 1`	Removes all window except for the current one
`C-x 0`	Removes the current window
`C-x o`	Changes to another window
`C-x >`	Scrolls horizontal to the right
`C-x <`	Scrolls horizontal to the left
`C-x ^`	Enlarges current window vertically
`C-x +`	Adjusts all windows
`C-x -`	Reduces window according to its contents
`C-x {`	Reduces window horizontally
`C-x }`	Enlarges window horizontally
Transform	
`M-l`	The following word to lower-case
`M-u`	The following word to upper-case
`C-x C-l`	Marked region to lower-case
`C-x C-u`	Marked region to upper-case
`M-c`	First character upper-case, rest lower-case
Miscellaneous	
Shift ⇑ - F5	Shows line number
`C-t`	Change characters at cursor
`C-x C-t`	Change lines at cursor position
`C-q` <`character`>	Insert character, can be octal
`C-x =`	Info on character at cursor position
`C-x f`	Column for carriage-return at cursor position
`C-x` TAB	Indent the marked region
`M-i`	Fills with blanks to next tab stop
`M-!`	Executes shell commands

Table 22.1: to be continued...

Esc |	Executes shell command, uses selected region as input
`M-=`	Cunts lines and charachters in marked region
Keyboard macros	
`C-x (`	Starts recording a *Aufnahme* macro
`C-x )`	Stops recording a *Aufnahme* macro
`C-x e`	Executes the last defined macro
`M-x name-last-kbd-macro`	Gives a name to the last macro
`M-x insert-kbd-macro`	Inserts a macro in current buffer

Table 22.1: Selected emacs commands

Chapter 23

Why LaTeX?

TeX and LaTeX have become a de-Facto standard in science as well as in publishing houses. The reason lies in its enormous capabilities of manipulating and formatting texts.

There are many fonts available at highest PostScript quality (mathematic, Cyrillic, arabica etc.). Various graphic formats can easily be integrated. And, although at the beginning LaTeX might look strange or cryptical, really, it is not. The longer you work with it the more you benefit from it and use it as an easy means of designing your texts.

LaTeX itself provides some macro packages for text formatting (such as articles, books, letters etc.); this reduces layout problems and ensures that all documents use the same format. This e. g.is useful if many authors cooperate.
.

Normally you just enter plain ASCII text without taking care of formats.

You can create your own macros if you encounter formats that you want to use more often. This can reduce a great amount of work. Macros are just typed into the document as if they were normal text.

23.1 Maintenance of TeX

The TeX-package that comes with your S.u.S.E. Linux is a widely spread package in the Unix™world: teTeXby **Thomas Esser**. Along with teTeX there comes a couple of documentation under `/usr/lib/teTeX/texmf/doc/` on TeX, LaTeX, and some layout styles. Besides the extensive TeX-system including BibTeX, **makeindex**, **dvips**, **xdvi**, **MetaFont** and various others there is a well-designed maintenance tool. **texconfig** enables you to maintain and configure the whole teTeX-system. For example, if you invoke **texconfig** and enter `'MODE'` you can select the default font that suits your printer and much more. It really is recommended to read the manual page of **texconfig**. Just launch **texconfig** and let yourself be guided through the menus. There is another tool, **texhash**. When invoked it creates a hash-table (**ls-R(5)**) which contains the whole directory under `/usr/lib/teTeX/texmf/`. Most of teTeXś tools consider this hash-table to find files under `/usr/lib/teTeX/texmf/`. After installation of new styles, classes, fonts, or graphics you should *always* start **texhash** afterwards.

23.2 Creating a document

With this little example we can only cover a small part of LaTeXś capabilities. There already are plenty of good books on this subject [Goossens et al., 1993] and [Lamport, 1994]; we will not add one but give you an every-day's use of TeX. We concentrate on the latest LaTeX release, LaTeX. This version is the predecessor of LaTeX 2.09 and the predecessor of the developer's version LaTeX3.

Before we go into the gory details of the *letter layout* some words on TeX, and LaTeX. TeX is an interpreter for word processing. It has been developed by **Donald E. Knuth** and compiles a text using control and macro commands. TeX, or better LaTeX is a complex TeX macro package which avoids the cryptical language of "plain" TeX. One word on compatibility to the experienced LaTeX 2.09 user: LaTeX contains an LaTeX 2.0.9 mode which smoothly works together with LaTeX 2.0.9 macro calls. Problems might be caused by LaTeX 2.08 styles as they access internal macros of LaTeX 2.0.9 and those differ from the styles used by LaTeX.

A rather important part is writing letters. Included in our teTeX-packages there is the LaTeX documentation class **letter**. We take this as an example to show you how LaTeX works.

file contents 23.2.1, page 379 shows a letter using **letter** document class.

A detailed description of the **letter**'s capabilities can be found in the teTeX-package or under [Lamport, 1994].

Besides this **letter** there are many other classes; among these are **book**, **report** oder **article**. There are certain font-sizes that belong to these classes which can be selected via options. The classes **book**, **report** and **article** have three different fonts: 10 points (Option `10pt`), 11 points (Option `11pt`) and 12 points (Option `12pt`).

\usepackage is used to load other layout styles or special macro packages.

All TeX commands start with a '\'.

Environments are structured using
\begin{<**environment**>} and
\end{<**environment**>}

\begin{**document**} starts the document and
\end{**document**} ends it.
\begin{**textbf**}... **\end**{**texbf**} turns all to bold whereas **\textbf**{ ... } is considered for small, bold paragraphs. **\bfseries** switches to bold-mode.

Besides this there are font families:

\textrm{ ... } or **\rmfamily**, **\textsf**{ ... } or **\sffamily**,
\texttt{ ... } or **\ttfamily**

and different styles

\textup{ ... } or **\upshape**, **\textit**{ ... } or **\itshape**,
\textsl{ ... } or **\slshape**.

```
%%
%% Created with  Emacs and the AucTeX package from  ‘e-site’
%% package.
%% Uses: LaTeX2e from ‘tetex’ package.
%
% Document class ‘letter.cls’
% Option ‘norm’ for size
%
\documentclass[12pt]{letter}
% A document :-)
%
\begin{document}
%
% By whom?
%
\address{Gladstone Ganter\\
          10 Luckyavenue.\ 100\par
          08015 Duckcity}
%
\subject{\textbf{Your fort-knox}}
%
\encl{A Dollar bill}
%
\backaddress{Heinz Hugo, Poor-road\ 100, 08015 Duckcity}
%
\signature{Heinz Hugo}
%
\date{Duckcity /06/12/1995}
%
% To whom?
%
\begin{letter}{Scrooge McDuck \\
               1 Dollaravenue\ 1\par
               04711 Duckcity}
%
% Opening
%
\opening{Dear Mr Duck,}

Your fort-knox. ...

%
% Closing
%
\closing{yours sincerley}
%
\end{letter}
\end{document}
```

File contents 23.2.1: Example letter of the **letter** document class

23.3 Compiling the document

Using the command

```
latex <new_filename>.tex
```

the letter is compiled. This should run without any problems (if you did not change the structure of the file). If an error occurs the compiler stops and requests you to correct the error. Since this normally is not so easy you can cancel compiling by pressing Strg + d. Last output of TeX gives you a hint where you can find the error.

23.4 Viewing the document

Now the document can be viewed (under X):

```
xdvi <new_filename>.dvi
```

This can take a while as some fonts have to be generated automatically. Positioning of the new window should be somewhere in the upper left corner (it is quite big; you can fix it using the left mouse button).

There is a zoom in `xdvi`. Pressing a number lower or higher than the standard `3` followed by s the screen is zoomed out (or in).

The graphic can be scrolled sidewards using the cursor keys; moving page by page is done with Page↑ and Page↓.

If your document contains more pages you can switch to a certain page by entering the page number and g. Of course you can use the *mousebuttons* on the right hand side as well. More information can be obtained via manual pages.

23.5 Printing the document

Printing of the `.dvi` file is done by:

```
dvips texbrief[.dvi] [Optionen].
```

You do not have to give the `.dvi` extension. **dvips** is a rather powerful tool that supports various options for printing PostScript; you have to read the manual pages. Here just some options:

-r	Pages are printed in reverse.
-A	Only odd numbers are printed.
-B	Only even numbers are printed. In connection with the previous option this serves printing both sides of a page.
-c <n>	Each page is printed <n> times.
-C <n>	The whole document is printed <n> times.
-o	Output directly goes into another file. If no new filename is selected the file keeps its name but with an `.ps` extension. Without this option a name given in `config.ps` is used.
-pp	Printing certain defined pages. Single pages are separated by commas; A range of pages can be given by a dash (e. g.A-Z).

Table 23.1: Options of **dvips**

Example:

```
dvips Linux_manual -r -A -pp 2,5,24-32 -C 2
```

Pages 5, 25, 27, 29, 31 would be printed in reverse and twice. Page 2 would not be printed at all.

Appendix A
Important keys

A short summary of helpful and important keys.

Enter	Executing a command on commandline, otherwise linefeed.
Alt + F1 - F6	Change to another console (in text mode).
Ctrl + Alt + F1 - F6	Under X11 and Dosemu: Change to one of the text consoles.
Alt + F7	Switching back to X11.
Ctrl + Alt + Backspace	Exit X11.
Shift ⇑ + Page ↑	Shows previous screens on console Works until consoles are switched.
Shift ⇑ + Page ↓	Opposite of Shift ⇑ + Page ↑ .
↑	Shows the last command in command line.
↓	Just the opposite of ↑ (in shell).
Shift ⇑ + Alt + Cursor	X11: Change to the next virtual desktop (direction with the cursor) Amount of virtual desktops is set in `.fvwm[2]rc` with the parameter "DeskTopSize".

Table A.1:

Ctrl + Alt + ±(grey)	Changes resolution according to entries in *Xconfig*.
Ctrl + D	Log out; corresponds to Befehl **`exit`**. Entry in `/etc/profile`: **`ignoreeof=x`**. **`x`** how often to repeat until the command is active.

Appendix **B**

Glossary

We have limited our glossary to **UNIX** or Linux specific terms respectively, since a complete introduction to electronic data processing is beyond the scope of this book. The reader of this glossary should be familiar with terms as *bits* and *bytes*.

ATAPI

ATAPI is a type of CD-ROM drive that is connected to an (E)IDE controller. Apart from ATAPI drives there are SCSI- CD-ROM drives, that are handled by a SCSI controller and proprietary CD-ROM drives that use their own controller or are connected to a sound card.

Account

Combination of *login* and *password*. In general the account is created by the ☞*system administrator*. He also assigns one or more groups to the user, as well as the resulting permissions. Creating an user account normally includes assigning a ☞*home directory* and the delivery of ☞*email*.

Acronym

Quite often abbreviations are called acronyms. ☞*Linux*, ☞*FTP* and ☞*GNU* are well-known acronyms.

Alias

This expression often is used in connection with ☞*shells*. An alias lets you abbreviate often used commands. Please have a look in chapter on shells.

Background process background process

If your ☞*Shell* (apparently) is only occupied with one process one speaks of a *foreground* process. Moreover most shells offer the ability of running processes in the background.

If you want to run a process in the background you have to attach a `'&'` at the end. The only thing that is required for such a behaviour is a ☞*multitasking* operating system.

Backup

Backups should be done regularly, especially the important files! Even some configuration files that have been created with difficulty are worth being back-upped. The Linux specific backup command is **`tar`**. It back-ups the files to a ☞*device*. Quite often **`tar`** is used in combination with **`gzip`**. ☞*GNU*

Base-Linux

When you install Linux for the first time the base-Linux has to be brought up first. It works without the hard disk which is not accessible at this moment. Its kernel is on the boot-disk or on CD-ROM, the root-image is loaded into a RAM-disk (also on CD-ROM or on boot-disk). The other programs (e. g.YaST) are also loaded into RAM-disk.

After the first login one starts YaST and prepares for installation of the "real" Linux.

BIOS Basic Input Output System

Any PC contains a little memory area that contains the "BIOS". This is a program for executing basic operations as memory check and recognizing hard disks. Under Linux the BIOS is not active since it runs in real mode and is switched off by the kernel at boot time. Anyway the Linux ☞*Kernel* provides much better capabilities than the BIOS.

Booting

Booting is the sequence from the very start until the system is ready for use. Under Linux this is booting the kernel which can be seen by the message `"uncompressing linux..."` and which ends with the `'login:'` prompt

Buffer

A buffer is a sort of intermediate memory which enables you to speed up access to data stored in itself. Under Linux there are many buffers.

Command line

Working with Unix is command line oriented. This in particular means that any process you enter in a ☞*shell* can have its own command line (e. g.the command **`ls`** can take a lot of options to change its behaviour.

Console

In former times this was a synonym for terminal. Nowadays under Linux you have *virtual consoles*. This enables you to use one screen for many independent running sessions. In standard ☞*runlevel* 2 you have 6 virtual consoles which can be reached by pressing Alt + F1 to Alt + F6. From within a running X-Window System[1] you reach the text consoles by pressing Strg + Alt + F1 to Strg + Alt + F6.

CPU Central Processing Unit

The processor. Intel x86 processors can be run in many different modes; here we want to distinguish between two of them:

[1] which uses either console 7 (runlevel 2) or console 3 (runlevel 3)

- real Mode: The “original” mode using a segmented memory protocol. This is slow and antiquated (“16 bit software”).
- protected Mode: (avaliable from 386 onwards). Mode with linear memory model. Only this mode uses the full power of the CPU. Linux only runs in protected mode.

By time of writing of this book Linux exists for the following processor architectures: Intel x86, DEC alpha, Motorola m68k, Sparc, PowerPC, MIPS, ARM.

Cursor

The cursor normally is a little block-character which marks the input. Under Linux this term occurs in:

- Shell/Editor: A spot marked with a rectangle of a (flashing) line where the next entry will occur. In the shell the cursor is right to the ☞*Prompt*.
- Mouse cursor under X: The mouse cursor under X. Depending on the background it changes its shape. For example an arrow in a xterm, X-shaped on the root window or I-shaped in an ☞*Editor* (buffer).
- GPM cursor (Console): A one character block which can be positioned on the console by means of the mouse. It can be used for Cut & Paste (☞*Selection*).

Directories

Build a ☞*file-systems* structure. In a directory files or directories are listed. You say a ☞*file* `'x'` lies in directory `'y'` if its name is listed in this directory. Because there are branches of subdirectories within other directories this is often referred to as ☞*directory tree*. If you want to see another directory you can change into it. Files are considered the leaves of the directory tree which cannot (quite logically) contain any more subdirectories (branches). Directories follow the same restrictions as files. The special directories `'.'` and `'..'` refer to the directory itself and to its parent directory in file-system hierarchy.

Editor

Editors are programs for changing text (e. g.by entering text). Well-known editors under Linux are **GNU Emacs** (`emacs`) or the UNIX editor **vi**.

Email electronic mail

Means of transporting electronic mail between registered users via a network, Same as with “normal” mail (often referred to as “snail mail”) the address has to be entered as “email address”. electronic mail not only lets you send text, but sound-documents or pictures. electronic mail has got plenty of advantages: first its quite cheap and second the mails reach the address normally within minutes.

Environment

A ☞*shell* normally provides some kind of environment where the user temporarily can set options such as paths of programs, the user name, the current path, prompt, shape and more. This data is stored in a ☞ *environment variable*. These variables can be assigned e. g.by the shells configuration files.

Environment variable

A storage place in the ☞*environment* of the ☞*shell*. Every variable consists of a name (which usually is written in uppercase) and a value (e. g.pathname). If you use the bash this is done like:

```
root@earth:/ > export EDITOR=emacs
```

Invoking **env** you can get a list of the variables set. If a variable is needed (e. g.in a shell-script) it is dereferenced by attaching a $ at the beginning. Important environment variables are **$HOME** (path to user directory), **$SHELL** (shell path), **$USER** (user name), **$PATH** (path for executables), **$MANPATH** (path for man pages).

EXT2 second extended File-system

is the file-system used by Linux. It offers a high throughput, long file names, permissions, and error tolerance as well.

File

Under Linux a file is the central concept for handling data. As with other systems files serve for writing data onto a mass storage media. The file name has to be unique in its directory, By means of a ☞*file-system* these files can be hierarchically structured. Please have a look at other topics on files in this glossary.

Additionally there are special files under Linux as well. See ☞*link*, ☞ *device* and ☞*proc*.

Filesystem

A file-system is a system for structuring files. There are many file-systems available which differ (sometimes extremely) in their performance and power. Some file-systems are strictly tied to certain media. One cannot easily say "Linux uses file-system 'X'". A list of supported file-systems can be found in section 19.11.

Focus

A control item, e. g.the command line of a ☞*terminal*, has the focus if the actual input is directed to it. Mostly the focus is connected with the ☞ *cursor*. The way a window manager manages the focus is called focus policy. You can differentiate between a focus that follows the mouse or a focus that has to be set by clicking the mouse.

FTP file transfer protocol

FTP is a means under UNIX for transferring files from one machine to another. There is the FTP server (the machine which sends the files) on the one side, and the FTP client (which receives the packages) on the other.

GNU

GNU stands for *GNU is Not Unix* and is a product of the **Free Software Foundation (FSF)** whose aim it is to provide a completely free (source code available by no costs) Unix compatible operating system. In this progress any Unix tools are completely rewritten and new functionality is added. Linux profits from these tools but should not be confused with it.

Graphical User Interface

A GUI is the graphical form of a normal desktop. Whereas you lay different papers onto a normal desktop, here these "papers" are called ☞ *windows*. You can put as many of these windows on your desktop as you like. All your different processes run in a separate window. A GUI normally is controlled by ☞*control elements* via a mouse, trackball or something similar. Well-known GUIs are e. g.☞*X Window System*, **Apple MacIntosh** System 7, **Digital Research GEM**, **MS-Windows**.

Home directory

The home directory is the starting point for most user activities on the machine. He can put his private files into his home directory. Besides the ☞*system administrator* he is the only one who has access to this directory. The location of the home directory in the file-system under **UNIX** is given by the ☞ *environment variable* **$HOME**.

Interface

Generally speaking, an interface is a device where different systems exchange information in any way. Such an interface e. g.is a keyboard where men and machine meet. This is rather abstract, some other interfaces are much more exact:

- *Hardware interface*: for connecting peripherical devices as a parallel port, SCSI or serial interfaces.
- *Software interface*: Assigns how programs should communicate. See ☞*protocol*.
- *User interface*: Here persons and machine exchange data. e. g.mouse, monitor and keyboard.

Internet

The internet is a worldwide heterogenuos (i. e. it consists of lots of different machines) ☞*network*. Machines are accessed via their ☞*IP addresses* (IP = internet protocol) which are unique. Those IP addresses are structured in an hierarchical manner. There are national, domains, sub-domains and the machines address. Besides the numerical IP address (like: `192.168.0.1`) there are aliases (like: `helios.cosmos.com`) which simplify the names for the user. It is not only the ☞*hardware* layer that keeps the internet up but also a couple of ☞*protocols* (e. g.ftp, http, tcp) on certain logical layers. Well-known services of the internet are ☞ *email* or the WorldWideWeb (WWW or W3). A real important keyword in connection with the internet is the “Netiquette” which makes sure that everything runs in order. This can be regarded as a voluntar self-control.

Inode

The ☞*EXT2* file-system uses inodes for organizing information on files. Moreover inodes contain information such as owner of its file, permissions and more . . .

LAN local area network

A LAN is a local ☞*network* which means that is is rather small and generally supervised by one ☞*system administrator*. LANs are normally connected to other LANs via a gateway thus forming a ☞*WAN*.

Main memory (RAM)

This is often referred to as ☞*RAM*. RAM access is very fast in comparison to hard disk access. Under Linux this ☞*memory* is often referred to as physical memory.

Mass storage media

A collection of different media for store data.

Typical mass storage media are disks, hard disks, tapes, CD-ROMs, magneto-optical disks, holographic media and many more.

Memory

The memory is the brain of your machine. Under Linux one often refers to two different types of memory:

- *physical memory*: This is the typical memory (RAM) chip. Typical size is from 8 MB up to 128 MB.
- *virtual memory*: By means of virtual memory the system can consider certain ranges of the disk/floppy etc. as memory (☞*swap*).

Menu

On ☞*graphical user interfaces* functions and actions of an application can be reached via a menu. Menus offer all available commands and options (just in a restaurant), so the user can choose either of them. Normally there is a menu bar with sub-menus. Moreover there are popup-menus which can be invoked e. g.with a button or a mouse-click.

Mountpoint

A mountpoint is the directory where a partition or another device is attached to the Linux file-system.

Multiprocessing

If your machine works with more than one ☞*processor* this is called a multiprocessor or multiprocessing system. Under Linux you will encounter the term SMP which stands for *symmetric multi processing* and is a special form of multiprocessing.

Multitasking

Operating systems that can invoke more than one ☞*program* simultaneously are called multitasking systems. There are two forms of multitasking:

- preemptive multitasking: The operating system is responsible for sharing cpu-time between the processes.
- cooperative multitasking: The processes themselves give back cpu-time.

As you can see the first variant is the better one, since no process can block the whole CPU. Linux offers you real preemptive multitasking.

Multiuser

☞*Multitasking* is required if more than one person wants to work on the same machine. An operating system that offers this capability is called a multiuser system.

Network

A network is a connection of different machines. There are different types of networks, that depends on how the machines are connected: ring, star,bus, tree and more. Well-known hardware standards for networks are e. g.ethernet and Token-Ring or ISDN. Important terms on software protocols are (on different layers) e. g.TCP, UDP, IPX and more.

Operating system

The operating system is a process running permanently and in background which enables basic operation with the machine. Tasks of an operating system are the management of all machine specific resources. Under Linux this is done by the ☞*kernel* and perhaps some modules. Well known operating systems are **Linux**, **AmigaOS**, **MacOS**, **OS/2**, **UNIX**, **Windows NT** and many more . . .

Path

Via the path a ☞*file* unambiguous is assigned to a ☞*file-system*. In Unix the different layers are separated by a `'/'`, a “slash”. There are two sorts of paths:

- *relative path*: here the position of a file or a directory is relative to the actual (current) path.
- *absolute path*: here the position is described relative to the ☞*root directory*.

Pipe

A pipe stands for connecting the standard-out of a *process* (☞*program*) with the *standard-in* (☞*standard in/out*) of a successor process. Thus avoiding having to write temporary files for further processing. In shell mode the processes to be “piped” are entered one after another separated by a pipe `'|'` (ASCII 124).

Process

A process is a program or an executables files (☞*Shell*) “living” variant. Often it is referred to as task.

Prompt

Within a text ☞*Shell* the place where users can type commands to the ☞ *operating system* is referred to as prompt. The prompt might contain the users name, the machines name, current time and certain others. Most of the time the ☞*cursor* stands directly behind the prompt. If the prompt reappears this shows that the system is ready (for yourinput).

Protocol

Protocols organize either on hardware or on software side the communication between the different machines in ☞*networks*. They assign how the format of the transferred data has to be, which machines controls a machine etc. Well-known protocols are e. g.FTP, UDP, TCP, HTTP, etc.

ROM Read-only Memory

A CD is a good example for a ROM.

Reset

If the machine hangs and does not react anymore to keystrokes it might be trapped in an endless loop. The only solution to such a problem is to bring the machine back to a defined state. This is called a reset. After a reset the machine is in exactly the same state as when it is started. A reset should always be preferred as this is better to the machinery.

IMPORTANT!: As you might guess after a reset all data that resided in ☞ *RAM* of the machine is irreversibly lost!

Rlogin

Using a *remote login* one can log into a remote machine via the ☞*internet* as if you were sitting at its ☞*console*. If there is a ☞*X-Server* running on both machines you can even display the output of an X-application on the local display provided the **$DISPLAY** variable is set correctly.

Root

See ☞*System administrator*

Root directory

The toplevel directory of a ☞*file system*. On the contrary to all other directories there is no parent directory of the root directory. `'..'` points to itself. Under UNIX the root directory is known as `'/'`.

Runlevel

A runlevel describes a certain state of your system. In other words: The system behaves differently on different runlevels. There is a runlevel for system administration (`S`), as well as a runlevel for the GUI of the “X-Display Manager” (xdm; `3`).

Selection

Selection is a mechanism of the X-Server. You can mark text characters with the mouse (by moving over them holding down the left mouse button) and "move" them to e. g. another application (by moving the cursor in the appropriate window and pressing the middle mouse button). This is called "Cut & Paste".

Server

A server generally is a quite powerful machine which serves other machines connected via a ☞*network* and provides services or data. Besides the machines there are programs that are called servers as well. These programs offer (serve) services too. An example for a software server is e. g.the ☞*X-Server*.

Shell

The shell is the fundamentary interface to the ☞*operating system* (Kernel). By means of the shell commands can be entered. The shell provides a command line. To make processes run automatically most shells provide an scripting language of their own. These programs (called shell-scripts) can be considered intelligent batch files. Examples for shells are: **bash**, **sh**, **tcsh**

Standard- in/out

Every process inherits three channels where it can read or output data. These are standard-in (stdin), standard-out (stdout) and the standard-error (stderr). These channels are set to certain devices as default. Standard-in is set to your keyboard, standard-out and standard-err to the screen. By means of the shell these channels can be redirected. If the characters are not read from keyboard but from a file instead this is called a redirection of standard-in. In shell-mode redirection is invoked by entering `'<'` (std-in), `'>'` (std-out) and `'2>'` (std-err) . See ☞*Pipe*

Switch

Switches can change the (default) behaviour of programs. The so called ☞*command line* consists of the programs name and some (optional) switches.

System administrator

The person who is responsible for maintaining and supervising a complex system or network. Generally the system administrator is the only one who has access to any corner of the system (root permissions).

Task

See ☞*Process*

Telnet

Telnet creates a connection to a (remote) host and gives you a login on this machine, provided you have an account.

Terminal

A combination of a screen and a keyboard without computing capabilities. Also used on workstations that emulate a real terminal.

UNIX

is an operating system that is widely spread on workstations. UNIX supports vital concept such as running different machines in a network. UNIX consists of a kernel, a ☞*shell* and applications. Since mid-nineties there is a ☞*pc* freeware version available: Linux.

WAN wide area network

As opposed to a ☞*LAN* this ☞*network* is widely spread.

Wildcards

The characters `'*'` and `'?'` are generic signs and are used as joker or wildcards (as in a card game). The `'?'` stands for exactly on character, `'*'` replaces any number of characters even no characters. Wildcards are often used in regular expressions. For example: the command **`ls -l bild*`** lists all files in the actual directory that start with a "`bild`" and end with any character(s) (even only `bild`).

Window

Windows are rectangle shaped screen segments which normally are decorated by a frame. This frame normally contains decorations for changing the size, for moving the window and more. For being able to work with windows a ☞*X-server* and a ☞*Window manager* have to run.

Window manager

A window manager is responsible (among other things) for decorating your windows, provide you with certain functionality as resizing, moving and destroying windows. Furthermore it is responsible for the "loo&feel" of your system.

X-Server

Machines that run a X-Server can use ☞*graphical user interfaces* services provided by the ☞*X Window System*. An important task of the X-Server is to manage the displays. Normally each terminal has exact one display. Name of this display is set by the ☞*environment variable* **$DISPLAY**. The format of this variable is: <machine-name>:<displaynumber> ZB **`earth:0`**. You need to know your displays name for ☞*rlogin*.

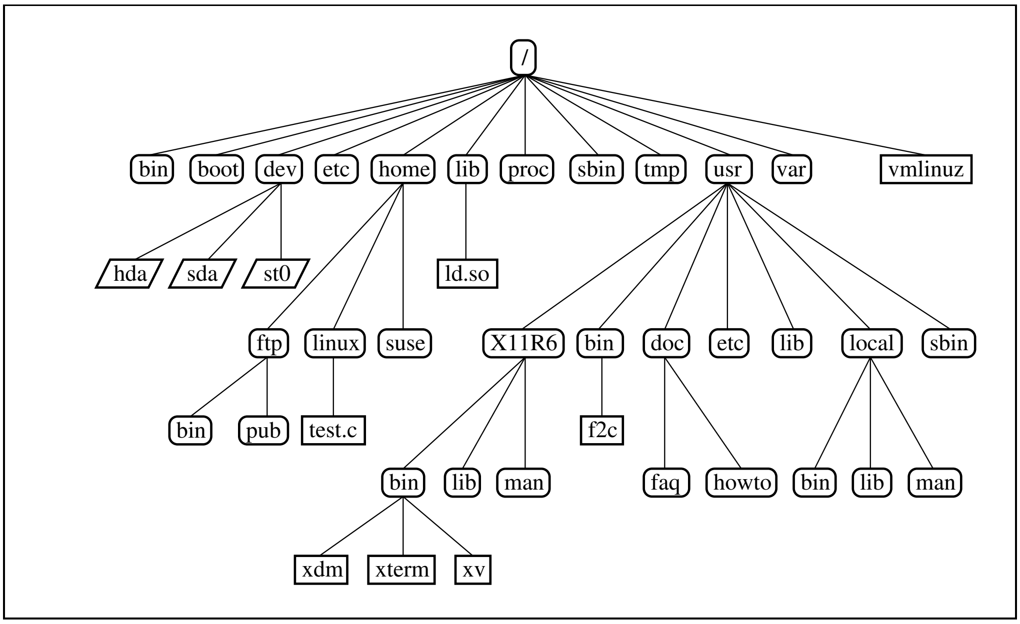
X Window System

A collection of programs, protocols and routines for organizing and maintaining a ☞*graphical user interface*. The X-Window System (short: X) has been developed as project Athena at MIT (Massachusetts Institute of Technology). Some of the big advantages of X-Window versus other systems (e. g.MS-Windows or GEM) are its networking capabilities and its flexibility. It is, e. g. possible to run programs on other machines and redirect their output to your display while free adapting the look&feel of your system.

Appendix C
The directory tree

C.1 Overview

The following figure shows a small portion of the Linux directory tree:



C.2 Important directories

The directory tree of your Linux system is well organized. Some important directories are:

`/`	"root" directory, start of the directory tree
`/home`	the (private) directories of the users
`/dev`	files, that represent the system hardware (device files)
`/etc`	important system configuration files
`/usr/bin`	publicly accessible commands
`/bin`	commands needed to start the system
`/usr/sbin`	commands reserved for the superuser
`/sbin`	commands reserved for the superuser and needed for system start
`/sbin/init.d`	scripts for system start
`/usr/include`	header files for the C compiler
`/usr/include/g++`	header files for the C++ compiler
`/usr/doc`	documentation files
`/usr/man`	manual pages
`/usr/src`	source code for the system software
`/usr/src/linux`	the kernel sources
`/tmp`	temporary files
`/var/tmp`	big temporary files
`/usr`	user commands and applications, configuration files, can be mounted *readonly*
`/var`	configuration files (linked from `/usr`).
`/lib`	shared libraries (for dynamically linked programs)
`/proc`	the process file system
`/usr/local`	local extensions, independent from the distribution
`/opt`	optional software, large systems (e. g. KDE)

Table C.1: Overview of important directories.

Appendix D

Important files

The most important file in your system is the kernel itself. It can be found in the root directory in the file `/vmlinuz`.

D.1 Device files in the /dev directory

Disks and hard disks: [1]

`/dev/fd0`	first floppy drive
`/dev/fd1`	second floppy drive
`/dev/hda`	first AT bus HD
`/dev/hda1` - `/dev/hda15`	partitions of first AT bus HD
`/dev/sda`	first SCSI HD
`/dev/sda1` - `/dev/sda15`	partitions of first SCSI HD
`/dev/sdb`	second SCSI HD
`/dev/sdc`	third SCSI HD

Table D.1: Overview of device file of mass storage media

D.1.1 CDROM drives

`/dev/cdrom`	Link to the used CD-ROM drive; one of the following files. Is assigned by YaST)
`/dev/aztcd`	Aztech CDA268-01 CD-ROM
`/dev/cdu535`	Sony CDU-535 CD-ROM
`/dev/cm206cd`	Philips CM206
`/dev/gscd0`	Goldstar R420 CD-ROM

Table D.2: to be continued...

[1] Besides the listed device files, you may create additional ones. Information: Manual Pages for "*mknod*".

`/dev/hda` to `/dev/hdd`	ATAPI IDE CD-ROM
`/dev/lmscd`	Philips CM 205/250/206/260 CD-ROM
`/dev/mcd`	Mitsumi CD-ROM
`/dev/sbpcd0` to `/dev/sbpcd3`	CD-ROM on SoundBlaster
`/dev/scd0` to `/dev/scd1`	SCSI CD-ROM drives
`/dev/sonycd`	Sony CDU 31a CD-ROM
`/dev/sjcd`	Sanyo CD-ROM
`/dev/optcd`	Optics Storage CD-ROM

Table D.2: Overview of device files of CD-ROM drives

D.1.2 Tapes

`/dev/rmt0`	1. SCSI-Streamer "‘rewinding’" (rewinds automatically)
`/dev/nrmt0`	1. SCSI-Streamer "‘non rewinding’"
`/dev/ftape`	Floppy-Streamer "‘rewinding’" (rewinds automatically)
`/dev/nftape`	Floppy-Streamer "‘non rewinding’"

Table D.3: Overview of device files of streamers

D.1.3 Mice (bus and PS/2)

`/dev/mouse`	Link to the interface used by the mouse (an pseudo file for bus mice) a serial interface for others. Is assigned by YaST).
`/dev/atibm`	ATI graphics card bus mouse.
`/dev/logibm`	Logitech bus mouse.
`/dev/inportbm`	PS/2 bus mouse.

Table D.4: Overview of device file of mice

D.1.4 Modem

/dev/modem	Link to the COM port where the modem is connected to. I assigned by YaST).

Table D.5: Overview of device files of modems

D.1.5 Serial interfaces

/dev/ttyS0 to /dev/ttyS3	Serial interfaces 0 to 3
/dev/cua0 to /dev/cua3	Serial interfaces 0 to 3 (for outgoing modem connections).

Table D.6: Overview of device files of serial interfaces

D.1.6 Parallel ports

Table D.7: Overview of device files of parallel ports

D.1.7 Special devices

/dev/null	"swallows" any data (data bin)
/dev/tty1 to /dev/tty8	Virtual consoles
/dev/zero	Outputs any number of null bytes.

Table D.8: Overview of device files of special (virtual) devices

D.2 Configuration files in /etc

`/etc/rc.config`	The systems central configuration file; is created by YaST and read by the bootscripts and **SuSEconfig**.
`/etc/inittab`	Configuration file for the init process.
`/etc/lilo.conf`	Configuration of LILO
`/etc/conf.modules`	Configuration of kernel modules
`/etc/DIR\_COLORS`	Assigning colours for **ls**
`/etc/XF86Config`	Configuration of the X-Window System
`/etc/fstab`	Table of file systems that are automatically mounted at boot time
`/etc/profile`	The shells login script
`/etc/passwd`	User database: user names, `home` directory, login-shell, user number
`/etc/shadow`	Passwords
`/etc/group`	User groups
`/etc/printcap`	Description of installed printers. Is used by the lpd printer daemon. page 143
`/etc/hosts`	Assigning hostnames to IP addresses; necessary if no nameserver is installed.
`/etc/inetd.conf`	Definition of configured IP services (**telnet**, **finger**, **ftp** and many more).
`/etc/syslogd.conf`	Configuration file for the syslog daemon; it servers for reporting certain system messages.

Table D.9: Configuration files in /etc

D.3 Hidden configuration files in `home`

In the home directory of a user there are several configuration files which for practical reasons are “hidden”. The are touched rather rarely. Each file becomes hidden if you attach a dot at the beginning of the file. One can see those files by typing **`ls -a`**. Some examples are to be found in table D.10.

`.profile`	the users private login script (for bash)
`.bashrc`	Bash configuration
`.exrc`	Vi configuration
`.xinitrc`	X-Window system startup script
`.fvwmrc`	Configuration of fvwm windowmanager
`.ctwmrc`	Configuration of ctwm windowmanager
`.openwin-menu`	Configuration of olvwm and olwm windowmanager

Table D.10: Invisible files in the user’s home directory

These files are copied from `/etc/skel` when creating a new user.

Appendix **E**

Manual page of e2fsck

```
E2FSCK(8)                                              E2FSCK(8)


NAME
       e2fsck - check a Linux second extended file system

SYNOPSIS
       e2fsck  [  -pacnyrdfvstFSV ] [ -b superblock ] [ -B block-
       size ] [ -l|-L bad_blocks_file ] device

DESCRIPTION
       e2fsck is used to check a Linux second extended file
       system.

       device is  the  special  file  corresponding to the device
              (e.g /dev/hdXX).

OPTIONS
       -a     This option does the same thing as the  -p  option.
              It is provided for backwards compatibility only; it
              is suggested that people use -p option whenever
              possible.

       -b superblock
              Instead  of  using  the  normal superblock, use the
              alternative superblock specified by superblock.

       -B blocksize
              Normally, e2fsck will search for the superblock  at
              various different block sizes in an attempt to find
              the appropriate block size.   This  search  can  be
              fooled in some cases.  This option forces e2fsck to
              only try locating the superblock  at  a  particular
              blocksize.  If the superblock is not found, e2fsck
              will terminate with a fatal error.

       -c     This option causes e2fsck to run  the  badblocks(8)
              program  to  find  any  blocks which are bad on the
              filesystem, and then marks them as  bad  by  adding
```

— *to be continued*

```
        them to the bad block inode.

-d      Print debugging output (useless unless you are
        debugging e2fsck ).

-f      Force checking even if the file system seems clean.

-F      Flush the filesystem device's buffer caches before
        beginning. Only really useful for doing e2fsck
        time trials.

-l filename
        Add the blocks listed in the file specified by
        filename to the list of bad blocks.

-L filename
        Set the bad blocks list to be the list of blocks
        specified by filename. (This option is the same as
        the -l option, except the bad blocks list is
        cleared before the blocks listed in the file are
        added to the bad blocks list.)

-n      Open the filesystem read-only, and assume an answer
        of ``no'' to all questions. Allows e2fsck to be
        used non-interactively. (Note: if the -c, -l, or
        -L options are specified in addition to the -n
        option, then the filesystem will be opened read-
        write, to permit the bad-blocks list to be updated.
        However, no other changes will be made to the
        filesystem.)

-p      Automatically repair ("preen") the file system
        without any questions.

-r      This option does nothing at all; it is provided
        only for backwards compatibility.

        -s This option will byte-swap the filesystem so
        that it is using the normalized, standard byte-
        order (which is i386 or little endian). If the
        filesystem is already in the standard byte-order,
        e2fsck will take no action.

-S      This option will byte-swap the filesystem, regard-
        less of its current byte-order.

-t      Print timing statistics for e2fsck. If this option
        is used twice, additional timing statistics are
        printed on a pass by pass basis.

-v      Verbose mode.

-V      Print version information and exit.

-y      Assume an answer of ``yes'' to all questions;
```

— *to be continued*

```
              allows e2fsck to be used non-interactively.

EXIT CODE
       The exit code returned by e2fsck is the sum of the
       following conditions:
            0    - No errors
            1    - File system errors corrected
            2    - File system errors corrected, system should
                   be rebooted if file system was mounted
            4    - File system errors left uncorrected
            8    - Operational error
            16   - Usage or syntax error
            128  - Shared library error

REPORTING BUGS
       Almost any piece of software  will have bugs.  If  you
       manage  to find a filesystem which causes e2fsck to crash,
       or which  e2fsck  is unable to repair, please report it to
       the author.

       Please include as much information as possible in your bug
       report.  Ideally, include a  complete  transcript  of  the
       e2fsck  run,  so I can see exactly what error messages are
       displayed.  If you have a writeable filesystem  where  the
       transcript can be stored, the script(1) program is a handy
       way to save the output of to a file.

       It is also useful to send the output of dumpe2fs(8).  If a
       specific  inode  or  inodes  seems to be giving e2fsck
       trouble, try running the debugfs(8) command and send the
       output  of the stat command run on the relevant inode(s).
       If the inode is a directory, the debugfs  dump  command
       will allow you to extract the contents of the directory
       inode, which can sent to me after being first run  through
       uuencode(1).

       Always  include  the full version string which  e2fsck
       displays when it is run, so I know which version you are
       running.

AUTHOR
       This  version  of  e2fsck  is  written  by  Theodore  Ts'o
       <tytso@mit.edu>.

SEE ALSO
       mke2fs(8), tune2fs(8), dumpe2fs(8), debugfs(8)


E2fsprogs version 1.06     October 1996
```

Appendix **F**

The GNU General Public License

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software–to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR

ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

Appendix: How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

< one line to give the program's name and a brief idea of what it does.>
Copyright (C) 19yy < name of author>

This program is free software; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this program; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

Gnomovision version 69, Copyright (C) 19yy name of author Gnomovision comes with ABSOLUTELY NO WARRANTY; for details type 'show w'. This is free software, and you are welcome to redistribute it under certain conditions; type 'show c' for details.

The hypothetical commands 'show w' and 'show c' should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than 'show w' and 'show c'; they could even be mouse-clicks or menu items–whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

Yoyodyne, Inc., hereby disclaims all copyright interest in the program 'Gnomovision' (which makes passes at compilers) written by James Hacker.

< signature of Ty Coon>, 1 April 1989 Ty Coon, President of Vice

This General Public License does not permit incorporating your program into proprietary programs. If your program is a subroutine library, you may consider it more useful to permit linking proprietary applications with the library. If this is what you want to do, use the GNU Library General Public License instead of this License.

Bibliography

[Almesberger, 1994]Almesberger, W. (1994). *LILO User's guide*. (siehe Datei `/usr/doc/lilo/user.dvi`).

[Bailey, 1997]Bailey, E. C. (1997). *Maximum RPM*. Red Hat. (ISBN 1-888172-78-9).

[Cameron and Rosenblatt, 1991]Cameron, D. and Rosenblatt, B. (1991). *Learning GNU Emacs*. O'Reilly & Associates, Inc. (ISBN 0 937175-84-6).

[Costales et al., 1993]Costales, B., Allman, E., and Rickert, N. (1993). *sendmail*. O'Reilly & Associates, Inc. (ISBN 1-56592-056-2).

[Dawson, 1995]Dawson, T. (v2.8, 07 Jan 1995). *Linux NET-2/NET-3 HOWTO*. (siehe Datei `/usr/doc/howto/NET-2-HOWTO`).

[Fang et al., 1993]Fang, C., Crosson, B., and Raymond, E. S. (1993). *The Hitchhiker's Guide to X386/XFree86 Video Timing (or, Tweaking your Monitor for Fun and Profit)*. (siehe Datei `/usr/X11/lib/X11/doc/VideoModes.doc`).

[Frisch, 1993]Frisch, Æ. (1993). *Essential System Administration*. O'Reilly & Associates, Inc. (ISBN 0-937175-80-3).

[Garfinkel and Spafford, 1993]Garfinkel, S. and Spafford, G. (1993). *Pratical UNIX Security*. O'Reilly & Associates, Inc. (ISBN 0-937175-72-2).

[Gilly, 1992]Gilly, D. (1992). *UNIX in a nutshell: System V Edition*. O'Reilly & Associates, Inc. (ISBN 1-56592-001-5).

[Goossens et al., 1993]Goossens, M., Mittelbach, F., and Samarin, A. (1993). *The LaTeX Companion*. Addison Wesley GmbH. (ISBN 3-54199-8).

[Grieger, 1994]Grieger, W. (1994). *Wer hat Angst vorm Emacs?* Addison Wesley GmbH. (ISBN 3-89319-620-X).

[Herold, 1992]Herold, H. (1992). *UNIX Grundlagen*. Addison Wesley GmbH. (ISBN 3-89319-542-8).

[Hetze et al., 1995]Hetze, S., Hohndel, D., Müller, M., and Kirch, O. (1995). *Linux Anwenderhandbuch*. LunetIX Softfair, 5.s edition. (ISBN 3-929764-04-0).

[Hunt, 1995]Hunt, C. (1995). *TCP/IP Netzwerk Administration*. O'Reilly & Associates, Inc. (ISBN 3-930673-02-9).

[Kirch, 1995]Kirch, O. (1995). *LINUX Network Administrator's Guide*. O'Reilly & Associates, Inc. (ISBN 1-56592-087-2).

[Kofler, 1995]Kofler, M. (1995). *Linux*. Addison Wesley GmbH. (ISBN 3-89319-796-6).

[Kopka, 1994]Kopka, H. (1994). *LaTeX-Einführung*. Addison Wesley GmbH. (ISBN 3-89319-664-1).

[Lamb, 1990]Lamb, L. (1990). *Learning the vi Editor*. O'Reilly & Associates, Inc. (ISBN 0-937175-67-6).

[Lamport, 1994]Lamport, L. (1994). *LaTeX User's Guide and Reference Manual*. Addison Wesley GmbH. (ISBN 0-201-52983-1).

[Leffler, 1996a]Leffler, S. (1996a). *HylaFAX Home Page*. `http://www.vix.com/hylafax/`.

[Leffler, 1996b]Leffler, S. (1996b). *TIFF Software*. `http://www-mipl.jpl.nasa.gov/~ndr/tiff/`.

[O'Reilly and Todino, 1992]O'Reilly, T. and Todino, G. (1992). *Manging UUCP and Usenet*. O'Reilly & Associates, Inc. (ISBN 0-937175-93-5).

[Perlman, 1994]Perlman, G. (1994). *Unix For Software Developers*. Prentice-Hall. (ISBN 13-932997-8).

[Pugh, 1994]Pugh, K. (1994). *UNIX For The MS-DOS User*. Prentice-Hall. (ISBN 13-146077-3).

[Schoonover and Bowie, 1992]Schoonover, M. and Bowie, J. (1992). *GNU Emacs*. Addison Wesley GmbH. (ISBN 0-201-56345-2).

[The XFree86™-Team, 1996]The XFree86™-Team (1996). *XF86Config(4/5) - Configuration File for Xfree86™*. Manual-Page zu XFree86™.

[Todino et al., 1993]Todino, G., Strang, J., and Peek, J. (1993). *Learning the UNIX operating system*. O'Reilly & Associates, Inc. (ISBN 1-56592-060-0).

[Welsh, 1994]Welsh, M. (1994). *Linux Installation and Getting Started*. S.u.S.E. GmbH, 2. edition. (ISBN 3-930419-03-3).

[Welsh and Kaufman, 1995]Welsh, M. and Kaufman, L. (1995). *Running Linux*. O'Reilly & Associates, Inc. (ISBN 1-56592-100-3).

Index

Y

Z