Manuál k $\mathcal{C}\mathcal{S}$TEXu

Petr Olšák

Hněvanice, Morávka, Praha – léto 2002

Autor TEXu je profesor Donald Knuth.

TEX je ochranná známka American Mathematical Society.

Ostatní v manuálu použité názvy programových produktů, firem apod. mohou být ochrannými známkami nebo registrovanými ochrannými známkami příslušných vlastníků.

Tento text je volně šířen společně s balíkem CSTEX. Jedná se o referenční manuál k tomuto balíku. Text je k dispozici ve formátech `tex` (CSplain), PostScript a PDF. Výchozí adresa textu je `ftp://math.feld.cvut.cz/pub/cstex/doc`.

Text můžete tisknout a jinak používat pro soukromé účely. V nezměněném stavu v elektronické podobě jej můžete také distribuovat bez dalšího omezení. Není dovoleno tento text zveřejňovat v pozměněném stavu a publikovat jej v papírové podobě bez předchozí dohody s autorem.

This document is free distributed as a part of the CSTEX package. It is a reference manual of the CSTEX. It is available in formats `tex` (CSplain), PostScript and PDF. The base URL of it is `ftp://math.feld.cvut.cz/pub/cstex/doc`.

You can print this document or use it in another way only as an individual end-user. If you don't do any change in this document then you can distribute it in electronical form without any limitation. It is not permitted to distribute this document in changed versions or to publish it in hardcopy form without previous agreement with the author.

This original is written in Czech language. The translation into other languages are welcome.

Verze textu 15. 5. 2003

Obsah

1. Úvod

V tomto manuálu se pokusím vyčerpávajícím způsobem popsat vlastnosti $\mathcal{CS}$TEXu. Z tohoto důvodu to asi nebude jednoduché čtení pro TEXové začátečníky. Těm doporučuji nejprve přečíst [4].

$\mathcal{CS}$TEX je sada TEXových maker, fontů, vzorů dělení slov a doplňujícího software pro podporu české a slovenské sazby v TEXu. Je vytvořen tak, aby mohl být použit na libovolné TEXové distribuci na libovolném operačním systému.

Dříve byla slovem $\mathcal{CS}$TEX označována také kompletní emTEXová distribuce doplněná o zmíněná makra, fonty a vzory dělení. To bylo v roce 1993, kdy sdružení $\mathcal{CS}$TUG rozesílalo pod tímto názvem emTEXovou distribuci na disketách svým členům. V té době většina členů používala DOS, takže emTEX pro DOS byl pro ně vyhovující. V dnešní době uživatelé pracují s nejrůznějšími operačními systémy a s různými distribucemi TEXu pro tyto systémy. Z toho důvodu se v tomto manuálu přidržíme jen užšího významu slova $\mathcal{CS}$TEX, definovaného v předchozím odstavci.

$\mathcal{CS}$TEX sestává ze tří základních pilířů: $\mathcal{CS}$fonty, $\mathcal{CS}$plain a $\mathcal{CS}$LATEX. $\mathcal{CS}$fonty jsou konzervativním rozšířením Knuthových Computer Modern fontů, $\mathcal{CS}$plain je konzervativním rozšířením Knuthova formátu plain a konečně $\mathcal{CS}$LATEX je jistou modifikací běžně používaného formátu LATEX. Tyto tři pilíře jsou podrobně dokumentovány v následujících kapitolách. K těmto pilířům $\mathcal{CS}$TEXu je ještě připojena podpora použití základních 35 PostScriptových fontů v češtině a slovenštině prostřednictvím virtuálních fontů.

Základní balíčky $\mathcal{CS}$TEXu přístupné na Internetu mají následující názvy:

```
csfonts.tar.gz   ... CSfonty -- Metafontové soubory a metriky.
csplain.tar.gz   ... Makra pro formát CSplain a vzory dělení slov.
cslatex.tar.gz   ... Makra pro formát CSLaTeX.
cspsfonts.tar.gz ... Virtuální fonty základních 35 PostScriptových
                     fontů v kódování podle CSfontů.
```

Pokud někdo řekne, že má na své distribuci TEXu instalován $\mathcal{CS}$TEX, pak to znamená, že má určitě instalovány tyto čtyři balíky. Kromě toho k $\mathcal{CS}$TEXu řadíme i následující doplňující software, který už nemusejí mít všichni uživatelé $\mathcal{CS}$TEXu instalován:

```
csfonts-t1.tar.gz .. Varianta CSfontů ve formátu PostScript Type1.
enctex.tar.gz    ... Makra a návod na modifikaci tex.ch souboru na
                     zahrnutí podpory přímého přístupu k vektorům
                     xord/xchr prostřednictvím primitivů.
cstrip.tar.gz    ... Test funkcionality formátu CSplain.
csindex-19980713.tar.gz ... Zdrojové soubory v jazyce C programu
                     csindex, který je modifikací programu
                     makeindex se zahrnutou podporou českého a
                     slovenského řazení.
vlna.tar.gz      ... Zdrojové soubory v jazyce C a v jazyce cweb
                     programu vlna na automatické doplňování vlnek
                     za neslabičné předložky.
```

Na Internetu naleznete všechny uvedené balíky na

`ftp://math.feld.cvut.cz/pub/cstex/base`

Základní balíčky a `csfonts-t1.tar.gz` obsahují adresářovou strukturu podle TDS (TEX directory standard), takže pokud vaše distribuce TEXu používá tento standard, pak rozbalení balíčků nad adresářem `texmf` povede automaticky k zařazení všech souborů na správná místa v `texmf` stromu.

Starší verze uvedených balíků najdete na

`ftp://math.feld.cvut.cz/pub/cstex/base/old`

Dokumentaci k CSTEXu (včetně tohoto manuálu) najdete na

`ftp://math.feld.cvut.cz/pub/cstex/doc`

Adresář `ftp://math.feld.cvut.cz/pub/cstex/` obsahuje kromě zmíněných věcí kompletní distribuci emTEXu s podporou CSTEXu (v podadresáři `emtex`), minimalizovanou distribuci emTEXu+CSTEXu pro pouhé tři instalační diskety (v podadresáři `minitex`), návody a software na instalaci web2c TEXu (v podadresáři `web2c`) a RPM balíky teTEXu podporující CSTEX pro různé distribuce Linuxu (v podadresáři `tetex-rpm`). Tyto podadresáře vznikaly postupně v poslední dekádě minulého tisíciletí podle potřeby a zájmu o uvedené distribuce TEXu. Podle toho, jak rostla a klesala popularita jednotlivých distribucí, jsou tyto podadresáře aktualizovány (nebo spíše neaktualizovány). Na druhé straně mohou vzniknout další adresáře s balíčky dalších distribucí TEXu obsahujících CSTEX – stačí, když je někdo udělá a pošle o tom informaci na mou elektronickou adresu. Mohu třeba založit podadresář `miktex-zip` analogicky k adresáři `tetex-rpm`, pokud se to někomu bude hodit a pokud ty archivy samozřejmě udělá.

2. CSfonty

CSfonty jsou konzervativním rozšířením CM fontů Donalda Knutha. Tím je míněno, že každý CSfont má svůj protějšek v nějakém CM fontu, přičemž se naprosto shoduje v kódování, tvarech a šířkách znaků na prvních 128 pozicích (kódy 0 až 127), tj. na všech pozicích, které jsou tímto CM fontem použity. CSfont tedy pouze doplňuje další znaky do pozic s kódem větším než 127. Tam jsou umístěny znaky české a slovenské abecedy podle kódování ISO 8859-2. Některé pozice stále zůstávají neobsazeny. Níže uvádím tabulku CSfontu. Druhý znak na pozicích 0B až 0F se vyskytuje namísto prvního znaku ve fontech bez ligatur typu fi (`csr5`, `cstt*` a `cssltt*`), druhý znak na pozicích 20, 3C, 3E, 5C, 5F a 7B až 7D najdeme ve strojopisných fontech (`cstt*` a `cslstt*`) a konečně libra místo dolaru na pozici 24 se vyskytuje v kurzívách (`csti*`). Tyto nejednoznačnosti v kódování mají samozřejmě i původní CM fonty.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	ϒ	Φ	Ψ	Ω	ff ↑	fi ↓	fl '	ffi ¡	ffl ¿
1x	ı	ȷ	`	´	ˇ	˘	¯	˚	¸	ß	æ	œ	ø	Æ	Œ	Ø
2x	ˉ␣	!	”	#	$ £	%	&	’	(	)	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	¡ <	=	¿ >	?
4x	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5x	P	Q	R	S	T	U	V	W	X	Y	Z	[	“ \	]	ˆ	˙ _
6x	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7x	p	q	r	s	t	u	v	w	x	y	z	– {	— |	˝ }	˜	¨
8x														‰		
9x									À				-	˛	«	»
Ax						Ľ				Š		Ť			Ž	
Bx						ľ			à	š		ť			ž	
Cx	Ŕ	Á			Ä	Ĺ			Č	É			Ě	Í		Ď
Dx			Ň	Ó	Ô		Ö		Ř	Ů	Ú		Ü	Ý		
Ex	ŕ	á			ä	ĺ			č	é			ě	í		ď
Fx			ň	ó	ô		ö		ř	ů	ú		ü	ý	„	“

Následuje seznam všech CM fontů. Pokud není vpravo uveden alternativní název CSfontu, jedná se o matematický font, který nemá v CSfontech alternativu. Také ji nepotřebuje.

```
CM font                           CSfont
------------------------------------------------------------------
cmr17, 12, 10, 9, 8, 7, 6, 5      csr17, 12, 10, 9, 8, 7, 6, 5
cmbx12, 10, 9, 8, 7, 6, 5         csbx12, 10, 9, 8, 7, 6, 5
cmsl12, 10, 9, 8                  cssl12, 10, 9, 8
cmtt12, 10, 9, 8                  cstt12, 10, 9, 8
cmsltt10, cmvtt10                 cssltt10, csvtt10
cmss17, 12, 10, 9, 8              csss17, 12, 10, 9, 8
cmssi17, 12, 10, 9, 8             csssi17, 12, 10, 9, 8
cmssdc10, cmssbx10                csssdc10, csssbx10
cmssqi8, cmssq8                   csssqi8, csssq8
cmdunh10, cmbxsl10, cmb10         csdunh10, csbxsl10, csb10
cmff10, cmfib10                   csff10, csfib10
------------------------------------------------------------------
cmti12, 10, 9, 8, 7               csti12, 10, 9, 8, 7
cmbxti10, cmitt10                 csbxti10, csitt10
cmu10, cmfi10                     csu10, csfi10
------------------------------------------------------------------
cmcsc10, cmtcsc10                 cscsc10, cstcsc10
------------------------------------------------------------------
cminch10                          csinch10
```

```
------------------------------------------------------------------
cmmi12, 10, 9, 8, 7, 6, 5
cmmib10
------------------------------------------------------------------
cmtex10, 9, 8
------------------------------------------------------------------
cmsy10, 9, 8, 7, 6, 5
cmbsy10
------------------------------------------------------------------
cmex10
------------------------------------------------------------------
```

Mezi soubory metrik CSfontů navíc najdeme metriky vytvořené Sauterovou extrapolací, které nemají přímou obdobu mezi CM fonty:

```
                                  CSfont
------------------------------------------------------------------
                                  csb17, 12, 9, 8, 7, 6, 5
                                  csbxsl12, 5, 6, 7, 8, 9
                                  csbxti17, 12
                                  cscsc17, 12
                                  csdunh17, 12, 5, 6, 7, 8, 9
                                  csfib12, 10, 9
                                  csitt12, 17, 8, 9
                                  cssl17, 5, 6, 7
                                  cssltt12, 8, 9
                                  csssbx12, 17, 9
                                  cstcs12, 17
                                  csti17
                                  csu12, 17, 7, 8, 9
                                  csvtt12, 8, 9
------------------------------------------------------------------
```

METAFONTová makra pro Sauterovu extrapolaci jsou součástí balíčku s CSfonty `csfonts.tar.gz`. Princip této extrapolace je například popsán v [2].

2.1 Odlišnosti od CM fontů

Nelze tvrdit, že text používající jen znaky z pozic 0–127 bude 100% shodně zpracován při použití CM fontů i CSfontů. Odlišnosti existují, ale jsou tak nepatrné, že je velmi malá prvaděpodobnost, že by při běžném užívání byla pozorovatelná rozdílnost. Nicméně přesto zde všechny odlišnosti uvádím včetně komentářů. Uvedené hodnoty jsou příkladem při srovnání fontu `csr10` s `cmr10`.

1. Kerningové páry

Mezi tečkami (`..`) je v `csr10` implicitní kern, aby bylo možno sázet elipsu. Kern $0,011111$pt. V cmr10 není.

Dvojice `ka` – `csr10`: $-0,0027777$pt, `cmr10`: $-0,0055555$pt.

Dvojice `P.` a `P,` – `csr10`: $-0,0027777$pt, `cmr10` není.
Dvojice `F.`, `F,`, `V.`, `V,`, `W.` a `W,` – `csr10`: $-0,0055555$pt, `cmr10` není.
Dvojice `Av` a `Aw` – `csr10`: $-0,011111$pt, `cmr10` není.

2. Ligatury
Dvojice `<<` vede v csr10 na francouzské uvozovky, kód 158, v cmr10 není.
Dvojice `>>` vede v csr10 na francouzské uvozovky, kód 159, v cmr10 není.

3. Výšky znaků
Formát `tfm` je omezen na maximálně 16 různých výšek znaků v jednom fontu. V `cmr10` je obsazeno všech 16 různých výšek. Přitom v `csr10` přicházejí další výšky znaků dané akcentovanými znaky. Proto METAFONT provedl v `csr10` jistá zaokrouhlení, která způsobí odlišnost výšek od výšek v `cmr10` maximálně o 0,007779pt. Jedná se o tyto znaky:

- Γ až Ω, Æ, Œ a všechny kapitálky: v `csr10` jsou menší o 0,00773pt.
- Nadržítko (kód 22), nadpuntík (kód 95) a přehláska (kód 127): v `csr10` větší o 0,007779pt.
- Písmena i, j jsou v `csr10` větší o 0,007779pt.
- Znak plus (+) je v `csr10` menší o 0,007778pt.

Rozdílnost výšek není kritická, protože při sazbě se většinou berou v úvahu jen šířky znaků. Pouze výjimečně promluví do sazby i výška (většinou když objekt v řádku je větší než `\baselineskip`).

2.2 Struktura METAFONTových zdrojových souborů CSfontů

METAFONT čte nejprve soubor, který má stejné jméno, jako je jméno generovaného fontu, a má příponu `mf`. Například `csr10.mf`. Takovému souboru říkáme hlavní soubor generování fontu. V CSfontech je takových hlavních souborů celkem 121. Všechny mají stejný dvouřádkový obsah:

```
input cscode
use_driver;
```

V souboru `cscode.mf` se načtou jména a kódy akcentovaných znaků (jsou uspořádány podle ISO 8859-2) a definuje se `use_driver` v závislosti na jménu hlavního souboru tak, že se načte odpovídající hlavní soubor CM fontů (tj. odpojí se předpona `cs` a připojí předpona `cm`). V případě fontu `csr10` se tedy načte v tuto chvíli soubor `cmr10.mf`. Navíc je v souboru `cscode.mf` předefinováno `generate` (v CM fontech má význam `input`) tak, že poslední řádek `cmr10.mf` s textem

```
generate roman
```

neprovede `input roman.mf`, ale provede `input kmroman.mf`. Další průběh výpočtu je tedy zase v režii CSfontů. Kromě souboru `kmroman.mf` jako alternativy k souboru `roman.mf` z CM fontů najdeme analogické alternativy s názvy `kmtextit.mf`, `kmcsc.mf`, `kmtexset.mf` a `kmtitle.mf`.

Z uvedených souborů se postupně načítají soubory generující tvary jednotlivých znaků. Jedná o všechny odpovídající soubory z CM fontů a navíc soubory uvedené v následující tabulce.

- `csaccent.mf` definuje makra pro akcenty,
- `csacutl.mf` generuje á, é, í, ó, ŕ, ú, ý,
- `csacutu.mf` generuje Á, É, Í, Ĺ, Ó, Ŕ, Ú, Ý,
- `cshachel.mf` generuje č, ě, ň, ř, š, ž,
- `cshacheu.mf` generuje Č, Ď, Ě, Ň, Ř, Š, Ť, Ž,
- `csotherl.mf` generuje ô, ů, à, ĺ, ť, ď, ľ, ä, ö, ü,
- `csotheru.mf` generuje Ô, Ů, À, Ľ, Ä, Ö, Ü,
- `csadded.mf` generuje „, “, », «, ‰, ˛,
- `cshyph.mf` generuje alternativní hyphenchar.

V souboru `kmroman.mf` a jemu podobných souborech jsou ještě zapsány kernové páry a údaje pro tabulku ligatur. V tomto souboru činnost METAFONTu končí příkazem `bye`.

2.3 Alternativní hyphenchar

Na pozici 156 je v CSfontech spojovník s úplně stejnou kresbou a metrikou, jako na pozici 45. Nastavíme-li `\hyphenchar\beznyfont=156`, budeme mít zaručeno, že ve slovech „je-li“ nebude TeX dělit slovo. Bez tohoto nastavení by TeX rozdělil „je-/li“, což není v souladu s požadavky na českou sazbu. Mnoho dalších způsobů řešení tohoto problému najdeme v [3].

Další aplikací znaku 156 je modifikace jeho metriky tak, aby kresba přesahovala přes šířku znaku výrazně doprava. Pak nastavením `\hyphenchar\beznyfont=156` dosáhneme tzv. visící interpunkce, viz například [1] nebo [3]. V poslední době se visící interpunkce stává módní záležitostí, protože to umí i nejnovější verze programu Adobe InDesign. Metriku CSfontu můžete pro vlastní potřeby modifikovat například programy tftopl a pltotf. Takto modifikovanou metriku ovšem nesmíte distribuovat pod stejným názvem, jaký má původní metrika CSfontu.

2.4 Anabáze UNIXových hard-linků

Jak jsme uvedli v předchozí sekci, hlavní METAFONTové soubory CSfontů mají jednu zvláštnost: ačkoli zde existuje 121 různě nazvaných souborů, všechny obsahují stejný dvouřádkový text.

Rozhodl jsem se těch 121 stejných různě nazvaných souborů implementovat pro úsporu inodů v UNIXu jako hard linky. Takto jsem fonty zabalil do balíčku `csfonts.tar.gz` a dal k dispozici internetové veřejnosti.

Pokud je systém, na který se balíček `csfonts.tar.gz` instaluje, rovněž UNIXového typu, pak program `tar` vytvoří v cílovém adresáři 121 hard linků a je vše v pořádku. Systém ušetřil 120 inodů, které nemusel alokovat.

Jestliže ale cílový systém nepracuje s hard linky, mohou nastat potíže. Osobně se ale domnívám, že pokud tento systém provozuje nějakou implementaci programu tar, měla by se tato implementace s problémem hard linků vyrovnat například tak, že místo hard linků vytvoří při extrahování archivu stejné soubory.

Proto považuji stížnosti uživatelů MS Windows na podivnost archivu cs-fonts.tar.gz za neopodstatněné. Tito uživatelé si stěžují, že se jim rozbalí z těch 121 souborů jen jeden, přičemž používají jakýsi Wintar. Odpovídám: nechť si tito uživatelé stěžují u svého dodavatele implementace programu tar. Odmítám kvůli Windowsovým uživatelům balit CSfonty jinak a opustit tak možnost šetření inodů na UNIXových systémech. Navíc se mi doneslo, že Windowsoví uživatelé mají možnost použít jiné implementace programu tar, které popsanou chybu neobsahují, a skutečně na Windowsovém filesystému založí 121 různých souborů se stejným obsahem.

2.5 Virtuální fonty přidané do balíčku CSfontů

V balíčku csfonts.tar.gz jsou přítomny dvě skupiny virtuálních fontů. Jedny mapují CSfonty na CM fonty a druhé mapují CM fonty na CSfonty. V tomto odstavci vysvětlím důvody existence obou skupin a způsob jejich použití.

Až donedávna bylo potřeba při prohlížení dvi souborů programem xdvi (a jemu podobnými programy) počkat, až se pomocí METAFONTu a programu gftopk vygenerují potřebné bitmapy fontů, a teprve pak jsme je viděli v prohlížeči. Teprve od roku 2002 má program xdvi schopnost přímo zobrazovat Type1 verze fontů, ale popisované virtuální fonty vznikly v době, kdy tomu tak nebylo a kdy CSfonty ještě neměly svou Type1 variantu. Pokud jsme tehdy nemuseli čekat na generování nových fontů METAFONTem, pak jen proto, že už byly fonty vygenerovány dříve a zůstaly uloženy na disku v podobě pk souborů.

Za této situace bylo výhodné udržovat na disku co nejméně pk souborů. Protože uživatel CSTeXu obvykle používá CSfonty, dá se předpokládat, že jejich pk soubory má na disku v hojném množství. Když pak jednou za čas potřebuje takový uživatel prohlédnout dvi soubor odkazující na CM fonty, pak je přeci zbytečné pro ně generovat nové bitmapy, když CSfonty jsou nadmnožinou CM fontů a bitmapy CSfontů pravděpodobně na disku už vygenerované jsou. Stačí pro zobrazení znaků CM fontů čerpat z bitmap CSfontů.

Právě k tomu účelu slouží první skupina virtuálních fontů označovaná jako cm2cs. Po rozbalení csfonts.tar.gz je tato skupina virtuálních fontů uložena do adresáře fonts/vf/public a je tedy aktivní. Každý dvi-ovladač, který potřebuje vykreslit znak z CM fontu, pak použije tyto virtuální fonty, které jej nasměrují na CSfonty. Takže dvi-ovladač bude tyto znaky nakonec čerpat z CSfontů. K žádnému zkreslení informace přitom nedochází, protože všechny znaky CM fontů jsou v odpovídajících CSfontech na stejných pozicích a vypadají úplně stejně.

Pokud CSfonty vůbec nepoužíváte (nebo jen občas), bude pro vás asi výhodné virtuální fonty cm2cs z adresáře fonts/vf/public odstranit. Stejně tak se nehodí mít tyto virtuální fonty aktivní v mezinárodních TeXových distribucích, kde většina uživatelů CSfonty nikdy nepoužije. Tito uživatelé by asi byli velmi překvapeni, že při prohlížení dvi souboru se standardními CM fonty jim distribuce generuje na disk bitmapy jakýchsi CSfontů.

Druhá skupina virtuálních fontů s označením cs2cm není po rozbalení balíčku s $\mathcal{CS}$fonty aktivní, protože není v adresáři `fonts/vf`, ale v adresáři `fonts/vf-cnv`, který dvi-ovladače s obvyklou konfigurací neprocházejí.

Tuto skupinu virtuálních fontů použijeme v případě, že jsme vytvořili v $\mathcal{CS}$TEXu dokument odkazující na $\mathcal{CS}$fonty, a nyní jej chceme ve formátovaném tvaru (jako soubor `dvi`) poslat například do zahraničí. Přitom předpokládáme, že příjemce nebude mít ve své TEXové distribuci $\mathcal{CS}$fonty.

Virtuální fonty cs2cm mapují všechny znaky $\mathcal{CS}$fontů na odpovídající znaky z CM fontů. Pokud znak v CM fontech neexistuje, virtuální font jej nahradí kompozitem (například písmeno+háček). Pokud tedy aplikujeme například program `dvicopy` s těmito virtuálními fonty, dostáváme nový `dvi` soubor, který už odkazuje jen na CM fonty, a akcentovaná písmena jsou v tomto `dvi` souboru nahrazena kompozity. Takový `dvi` soubor můžeme poslat do zahraničí a máme jistotu, že bude zpracovatelný na libovolné TEXové distribuci, protože každá distribuce obsahuje CM fonty. Kvalita sazby se ovšem zhorší, protože akcenty (coby samostatné znaky) jsou v CM fontech takové poněkud nedomrlé, zatímco v $\mathcal{CS}$fontech jsou kresleny společně s písmenem s velikou péčí. Informace v dokumentu ovšem není uvedenou modifikací `dvi` souboru vůbec změněna. Výjimkou je případ výskytu jednoho z těchto čtyř znaků: promile, ogonek (ocásek pod polské a), levá a pravá francouzská uvozovka. Tyto znaky nelze nijak pomocí CM fontů nahradit. Pokud dokument některý z těchto znaků obsahuje, pak program `dvicopy` ohlásí „`---missing character packet`“ a ponechá místo těchto znaků prázdná místa.

Ve web2c distribuci TEXu v UNIXu můžete pro uvedenou konverzi `dvi` souboru z $\mathcal{CS}$fontů na CM fonty použít příkaz:

```
$ VFFONTS=\$TEXMF/fonts/vf-cnv// dvicopy vstup.dvi vystup.dvi
```

Poznamenejme, že v žádném případě nesmí `dvi` ovladač najít současně virtuální fonty cm2cs i cs2cm. V takovém případě dojde k havárii, neboť odkazy ve virtuálních fontech se dostanou do nekonečného cyklu.

2.6 Historie a budoucnost $\mathcal{CS}$fontů

Tvary akcentů $\mathcal{CS}$fontů byly vytvořeny a implementovány v jazyce METAFONTu Petrem Novákem ve spolupráci s českými typografy někdy na začátku 90. let. Autor přenechal $\mathcal{CS}$fonty $\mathcal{CS}$TUGu, který s nimi může libovolně nakládat.

METAFONTový kód byl pak v letech 1992–1993 dále upraven Karlem Horákem. Karel se inspiroval z METAFONTových zdrojů pro fonty vytvořené v Polsku. Zapracoval tam možnost nastavení kódování fontu a vytvořil makra umožňující mít všechny hlavní `mf` soubory se stejným dvouřádkovým obsahem.

Na schůzce tvůrců $\mathcal{CS}$TEXu na FEL v roce 1993 bylo rozhodnuto, že $\mathcal{CS}$fonty budou mít kódování podle ISO 8859-2. Později, při implementaci $\mathcal{CS}$TEXu do UNIXových distribucí nepodporujících změny xord/xchr vektorů, se ukázalo, že to bylo velmi prozíravé rozhodnutí.

V roce 1993 jsem převzal údržbu $\mathcal{CS}$fontů do svých rukou. Udělal jsem jen velmi drobné změny. Poslední 28. 9. 1996:

Písmeno Č a další akcentované kapitálky měly před tímto datem větší výšku než kresba o 1,2pt. Opravil jsem. Také jsem tehdy odstranil nevhodné záporné kerny: Tě, Tř, Tö, Tü, Tä, Tà (analogicky pro Ť, Y, Ý). Vě, Vř, Vö, Vü (analogicky pro F, W) a redukoval jsem přílišné záporné kerny: Té, Tó, Tů, Tŕ, Tá, Tú (analogicky pro Ť, Y, Ý).

Pak jsem vývoj CSfontů zmrazil podobným způsobem, jako Knuth přestal měnit CM fonty. Prioritním požadavkem je, aby dokument opírající se o CSfonty byl od roku 1996 formátován naprosto stejně dnes i kdykoli v budoucnu. Aby byl tento požadavek splněn, není tedy možné zasáhnout do rozměrů znaků, kernů a ligaturních tabulek.

V roce 1996 jsem do CSfontů přidal virtuální fonty podporující náhradu Computer Modern fonty a naopak.

V roce 1998 se podařilo dohodnout s autorem teTEXu Thomassem Esserem, aby zařadil do své distribuce CSfonty a celý CSTEX. Od této chvíle jsou distribuce odvozené z teTEXu implicitně vybaveny CSfonty.

V roce 1998 jsem také pro potřeby výstupu do formátu PDF vytvořil variantu CSfontů, tentokrát ve formátu PostScript Type1. Vyšel jsem z BaKoMa Type1 implementace CM fontů a vytvořil jsem si program `t1accent`, který k písmenkům přidával akcenty podle vzorových PostScriptových tahů generovaných z původních CSfontů METAPOSTem. Na mnoha místech jsem byl nucen přistoupit k mikrotypografickým kompromisům – v drobnostech se kresby některých znaků CSfontů z Type1 liší od svých originálních METAFONTových protějšků. Proto jsem distribuci Type1 CSfontů označil jako „alpha" a do komentáře jsem dal důrazné varování, že tyto fonty je možné používat na vlastní riziko. Z toho důvodu jsem také ponechal implicitní konfiguraci programu `dvips` tak, aby program používal léty osvědčený výstup z METAFONTu, tedy bitmapy formátu `pk`. Type1 CSfonty byly původně konfigurovány jen pro pdfTEX.

Rozhodnutí ponechat `dvips` pracovat implicitně s bitmapami naráželo na problémy. Neustále dokola se uživaté ptali, jak je možné, že výstup z pdfTEXu je dobrý, ale při cestě dvips – pstopdf dostávají roztřesená písmenka. Protože jsem byl uondán velmi častým odpovídáním na tuto otázku, ani jsem se nakonec nezlobil, když v roce 2001 autor teTEXu rozhodl, že bude CSfonty pro `dvips` implicitně konfigurovat ve verzi Type1. Asi ty mikrotypografické kompromisy ani tak moc nevadí, zatímco roztřesená písmenka v PDF způsobovala oheň na střeše.

V současné době existují volně dostupné nástroje, jako například `textrace` opírající se o `autotrace`. Tyto nástroje umožní převést METAFONTový font do Type1 „obtahováním bitmap" skoro automaticky. Vyzkoušel jsem to na CSfontech a s výsledkem jsem nebyl vůbec spokojen: výsledné `pfb` soubory byly asi pětkrát větší než ty moje „ručně" vyrobené. Proto jsem zatím alpha verzi Type1 formátu CSfontů z roku 1998 neopustil.

Do budoucna bych velmi rád do CSfontů přidal znak euro a paragraf. Taková změna by byla zpětně kompatibilní, takže bych se jí nebránil. METAFONTové zdroje pro paragraf ověřené na všech CSfontech už několik let mám, ale nezveřejnil jsem je. METAFONTové zdroje znaku euro by se snadno daly převzít z jiného METAFONTového fontu. Největší potíž je ovšem v tom, že s uvedením nové verze CSfontů dnes nestačí zveřejnit jen METAFONTové zdroje a metriky, ale je třeba mít okamžitě s tím konzistentní Type1 varianty fontů. Do manuální práce na nové verzi Type1 varianty $\mathcal{CS}$fontů se mi ale moc nechce. Je to nevděčná a rozsáhlá práce: `pfb` souborů je v balíčku 57 a každý je třeba disassemblovat, v editoru přidat nové znaky a znovu převést na `pfb`. Přitom s automatickými nástroji, jak jsem uvedl před chvílí, nejsem spokojen.

3. `cspsfonts.tar.gz` – 35 základních PostScriptových fontů

Každý PostScriptový RIP musí být vybaven aspoň 35 základními fonty ze sady „base 35“. Sada mimo jiné obsahuje sedm rodin textových fontů, každá rodina obsahuje čtyři fonty (základní, kurzíva, tučný a tučná kurzíva). Strojopis se obvykle kombinuje z rodiny Courier. Pro tyto fonty je vytvořena podpora ve formě metrik v kódování $\mathcal{CS}$fontů a virtuálních fontů, které tyto metriky mapují na skutečné fonty, kódované samozřejmě úplně jinak. Tato podpora je v balíčku `cspsfonts.tar.gz`, který je povinnou součástí $\mathcal{CS}$TEXu.

Kódování těchto metrik je inspirováno kódováním $\mathcal{CS}$fontů, ale chybí matematické znaky ze začátku tabulky a znaky, které v běžných PostScriptových fontech nenajdeme (např. pozice 20: škrtátko polského l). Na druhé straně jsou přidány znaky s kódy 80–86 (hexadecimálně) a některé další (např. kompletní polské Ł a ł). Viz následující tabulku fontu `ptmr8z`, implementující písmo Times-Roman pro $\mathcal{CS}$TEX:

<table>
<tr><th></th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>A</th><th>B</th><th>C</th><th>D</th><th>E</th><th>F</th></tr>
<tr><td>0x</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>ﬁ</td><td>ﬂ</td><td></td><td></td></tr>
<tr><td>1x</td><td>ı</td><td></td><td>`</td><td>´</td><td>ˇ</td><td>˘</td><td>¯</td><td>˚</td><td>¸</td><td>ß</td><td>æ</td><td>œ</td><td>ø</td><td>Æ</td><td>Œ</td><td>Ø</td></tr>
<tr><td>2x</td><td></td><td>!</td><td>”</td><td>#</td><td>$</td><td>%</td><td>&</td><td>’</td><td>(</td><td>)</td><td>*</td><td>+</td><td>,</td><td>-</td><td>.</td><td>/</td></tr>
<tr><td>3x</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>:</td><td>;</td><td>¡ <</td><td>=</td><td>¿ ></td><td>?</td></tr>
<tr><td>4x</td><td>@</td><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td></tr>
<tr><td>5x</td><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td><td>U</td><td>V</td><td>W</td><td>X</td><td>Y</td><td>Z</td><td>[</td><td>“ \</td><td>]</td><td>ˆ</td><td>˙ _</td></tr>
<tr><td>6x</td><td>‘</td><td>a</td><td>b</td><td>c</td><td>d</td><td>e</td><td>f</td><td>g</td><td>h</td><td>i</td><td>j</td><td>k</td><td>l</td><td>m</td><td>n</td><td>o</td></tr>
<tr><td>7x</td><td>p</td><td>q</td><td>r</td><td>s</td><td>t</td><td>u</td><td>v</td><td>w</td><td>x</td><td>y</td><td>z</td><td>– {</td><td>— |</td><td>˝ }</td><td>˜</td><td>¨</td></tr>
<tr><td>8x</td><td>…</td><td>†</td><td>‡</td><td>•</td><td>£</td><td>¶</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>‰</td><td></td><td></td></tr>
<tr><td>9x</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>À</td><td></td><td></td><td></td><td>-</td><td>˛</td><td>«</td><td>»</td></tr>
<tr><td>Ax</td><td></td><td></td><td>˘</td><td>Ł</td><td>¤</td><td>Ľ</td><td></td><td>§</td><td></td><td>Š</td><td></td><td>Ť</td><td></td><td></td><td>Ž</td><td></td></tr>
<tr><td>Bx</td><td>°</td><td></td><td></td><td>ł</td><td></td><td>ľ</td><td></td><td></td><td>à</td><td>š</td><td></td><td>ť</td><td></td><td></td><td>ž</td><td></td></tr>
<tr><td>Cx</td><td>Ŕ</td><td>Á</td><td>Â</td><td></td><td>Ä</td><td>Ĺ</td><td></td><td>Ç</td><td>Č</td><td>É</td><td></td><td>Ë</td><td>Ě</td><td>Í</td><td>Î</td><td>Ď</td></tr>
<tr><td>Dx</td><td></td><td></td><td>Ň</td><td>Ó</td><td>Ô</td><td></td><td>Ö</td><td></td><td>Ř</td><td>Ů</td><td>Ú</td><td></td><td>Ü</td><td>Ý</td><td></td><td></td></tr>
<tr><td>Ex</td><td>ŕ</td><td>á</td><td>â</td><td></td><td>ä</td><td>ĺ</td><td></td><td>ç</td><td>č</td><td>é</td><td></td><td>ë</td><td>ě</td><td>í</td><td>î</td><td>ď</td></tr>
<tr><td>Fx</td><td></td><td></td><td>ň</td><td>ó</td><td>ô</td><td></td><td>ö</td><td></td><td>ř</td><td>ů</td><td>ú</td><td></td><td>ü</td><td>ý</td><td>„</td><td>“</td></tr>
</table>

Fonty rodiny Courier mají kódování inspirováno $\mathcal{CS}$fontem `cstt10`, takže se na pozicích 3C, 3E, 5C, 5F a 7B až 7D znaky poněkud liší. Toto kódování je označeno `8u` na rozdíl od kódování ostatních PostScriptových fontů z balíčku cspsfonts.tar.gz, které je nazváno 8z. Tuto nejednotnost kódování známe už u $\mathcal{CS}$fontů a prapůvodně existuje u Computer Modern fontů. Znak vlevo na zmíněných pozicích odpovídá kódování 8z a znak vpravo kódování 8u. Na ostatních pozicích je kódování 8z a 8u shodné.

Upozornění: Ve výše uvedené tabulce jsou prázdná místa, na kterých mohou být umístěny nedokumentované znaky. Když si vytisknete například skutečnou tabulku fontu ptmr8z, zjistíte, že tomu tak skutečně je. Pokud chcete mít dokument zpracovatelný i v budoucích verzích $\mathcal{CS}$TEXu, nemůžete se o tyto nedokumentované znaky opírat. Tyto znaky byly obsazeny v rámci soukromé iniciativy Zdeňka Wagnera, který potřeboval použít font pro více jazyků. Bohužel umístil některé znaky na pozice, které nejsou pro budoucí rozšíření fontů vhodné. Každý si může do volných pozic doplnit co potřebuje, ovšem s tím rizikem, že jeho dokumenty využívající tyto pozice nemusejí být kompatibilní s $\mathcal{CS}$TEXem.

Sada metrik balíčku cspsfonts.tar.gz zahrnuje následující fonty:

```
Metrika    odkazuje na   Odpovídající PostScriptový font
------------------------------------------------------------
pagk8z       rpagk       AvantGarde-Book
pagko8z      rpagko      AvantGarde-BookOblique
pagd8z       rpagd       AvantGarde-Demi
pagdo8z      rpagdo      AvantGarde-DemiOblique
pagkc8z  *   rpagk       AvantGarde-Book Caps & Small Caps
pagdc8z  *   rpagd       AvantGarde-Demi Caps & Small Caps
pbkl8z       rpbkl       Bookman-Light
pbkli8z      rpbkli      Bookman-LightItalic
pbkd8z       rpbkd       Bookman-Demi
pbkdi8z      rpbkdi      Bookman-DemiItalic
pbklc8z  *   rpbkl       Bookman-Light Caps & Small Caps
pbkdc8z  *   rpbkd       Bookman-Demi Caps & Small Caps
pcrr8u       rpcrr       Courier
pcrro8u      rpcrro      Courier-Oblique
pcrb8u       rpcrb       Courier-Bold
pcrbo8u      rpcrbo      Courier-BoldOblique
pcrrc8u  *   rpcrr       Courier Caps & Small Caps
pcrbc8u  *   rpcrb       Courier-Bold Caps & Small Caps
phvr8z       rphvr       Helvetica
phvro8z      rphvro      Helvetica-Oblique
phvb8z       rphvb       Helvetica-Bold
phvbo8z      rphvbo      Helvetica-BoldOblique
phvrc8z  *   rphvr       Helvetica Caps & Small Caps
phvbc8z  *   rphvb       Helvetica-Bold Caps & Small Caps
phvrn8z  *   rphvrrn     Helvetica Narrow
phvron8z *   rphvron     Helvetica-Oblique Narrow
phvbn8z  *   rphvbrn     Helvetica-Bold Narrow
phvbon8z *   rphvbon     Helvetica-BoldOblique Narrow
phvbnc8z *   rphvbrn     Helvetica-Bold Narrow Caps & Small Caps
```

phvrnc8z	*	rphvrrn	Helvetica Narrow Caps & Small Caps
pncr8z		rpncr	NewCenturySchlbk-Roman
pncri8z		rpncri	NewCenturySchlbk-Italic
pncb8z		rpncb	NewCenturySchlbk-Bold
pncbi8z		rpncbi	NewCenturySchlbk-BoldItalic
pncrc8z	*	rpncr	NewCenturySchlbk-Roman Caps & Small Caps
pncbc8z	*	rpncb	NewCenturySchlbk-Bold Caps & Small Caps
pplr8z		rpplr	NewCenturySchlbk-Roman
pplri8z		rpplri	NewCenturySchlbk-Italic
pplb8z		rpplb	NewCenturySchlbk-Bold
pplbi8z		rpplbi	NewCenturySchlbk-BoldItalic
pplrc8z	*	rpplr	NewCenturySchlbk-Roman Caps & Small Caps
pplbc8z	*	rpplb	NewCenturySchlbk-Bold Caps & Small Caps
ptmr8z		rptmr	Times-Roman
ptmri8z		rptmri	Times-Italic
ptmb8z		rptmb	Times-Bold
ptmbi8z		rptmbi	Times-BoldItalic
ptmrc8z	*	rptmr	Times-Roman Caps & Small Caps
ptmbc8z	*	rptmb	Times-Bold Caps & Small Caps
pzcmi8z		rpzcmi	ZapfChancery-MediumItalic

V tabulce jsou hvězdičkou označeny ty metriky, které neodkazují na speciální PostScriptový font, ale jsou implementovány pomocí virtuálního fontu (Caps & Small Caps). Nebo se jedná o Narrow varianty fontu Helvetica, které jsou konfigurovány v `psfonts.map` pomocí PostScriptové transformace.

České a slovenské znaky jsou ve fontech `8z` a `8u` implementovány jako kompozity (tj. virtuální font je poskládá ze samostatného akcentu a základního písmene). Použité segmenty jsou přítomny v každém PostScriptovém fontu. Proto virtuální fonty po sestavení kompozitů odkazují na metriky fontů (s písmenem `r` na začátku), které jsou již kódovány podle StandardEncoding. Fonty v PostScriptovém RIPu tedy nepotřebujeme mít počeštěny ani poslovenštěny a také nepotřebujeme měnit Encoding vektor těchto fontů na úrovni PostScriptu.

Program `dvips` je obvykle konfigurován tak, že do výstupního PostScriptového kódu nezavádí žádný z fontů „base 35“ a ponechá tam jenom odkaz. Starost o font přenechá PostScriptovému RIPu. PostScriptové fonty z „base35“ tedy principiálně není nutné pro provoz v TEXu instalovat na disk a nejsou tedy ani obsaženy v balíčku `cspsfonts.tar.gz`.

Pokud si chcete prohlížet `dvi` soubor odkazující na fonty z „base 35“ například starší verzí programu `xdvi`, pak tento program potřebuje (alespoň na přechodnou dobu) bitové mapy těchto fontů. Pokud bitové mapy v instalaci ještě neexistují, program zavolá skript `mktexpk`, který pro vytvoření rastru volá dále Ghostscript. To je implementace PostScriptového RIPu, která musí obsahovat fonty z „base 35“.*

* Z licenčních důvodů Ghostscript neobsahuje originální fonty od Adobe, ale jen jejich hodně dobré náhražky od firmy URW. Rozdíl většinou pouhým okem nerozeznáte.

Odtud se fonty konvertují do bitových map `pk` a odtud je nakonec použije program `xdvi`.

Poznamenejme ještě, že novější verze programu `xdvi` obsahují tzv. modul `t1lib`, díky němuž `xdvi` umí přímo číst PostScriptové fonty a vykreslovat je na obrazovku. Není tedy potřeba vůbec vyrábět `pk` bitmapy fontů, ani na přechodnou dobu. Náhražky fontů od firmy URW z „base 35“ jsou proto v novějších distribucích TEXu přímo instalovány a nespoléhá se v tomto případě na Ghostscript. Nemusíte pak ani spoléhat na přítomnost fontů v PostScriptovém RIPu v tiskárně. Novější verze web2c TEXu nabízejí pro takový případ možnost použít u programu `dvips` přepínač `-Pdownload`.

3.1 Historie a budoucnost balíčku `cspsfonts.tar.gz`

Balíček `cspsfonts.tar.gz` má méně bohatou historii než CSfonty. Tento balíček začal vznikat v září roku 1994. Tehdy jsem zjistil, že popisy kompozitů v `AFM` metrice pomocí řádků `CC` jsou správně převáděny programem `afm2tfm` na odpovídající kompozity ve vytvářeném virtuálním fontu. Problém byl jen v tom, že originální `AFM` metriky od Adobe neobsahovaly popisy všech kompozitů potřebných pro český a slovenský jazyk. Z toho důvodu jsem si vytvořil program `a2ac` [7], který na základě přehledné tabulky kompozity do `AFM` metrik doplnil a současně doplnil kerningové informace pro nově vytvářené znaky. Za použití tohoto programu pak vznikla sada metrik a virtuálních fontů s písmenem `c` na začátku (např. `cptmr`).

V roce 1996 pak uveřejnil pan Wagner nové metriky generované stejným způsobem, ovšem opravil několik estetických nedostatků a navíc metriky nazval podle doporučení Karla Berryho (`8z` a `8t` na konci). Od té doby jsou v balíčku `cspsfonts.tar.gz` obsaženy metriky pana Wagnera.

Konečně v roce 1999 jsem musel po konzultaci s Karlem Berrym metriky pro rodinu Courier přejmenovat z původního `*8t` na nynější `*8u`, protože názvy s `8t` na konci nám kolidovaly s názvy stejných fontů v kódování podle Corku. To je zatím poslední změna v tomto balíčku.

Chystám doplnění balíčku `cspsfonts.tar.gz` o nové metriky kódované jako `8z`, které mapují další běžně používané PostScriptové fonty. Jako první na řadě se nabízí přidání metrik `8z` pro volně šířenou rodinu fontů Charter písmolijny BitStream.

4. Formát CSplain

Popíšu jenom odlišnosti formátu CSplain (povel `csplain`) od Knuthova formátu plain (povel `tex`), popsaného v TEXbooku [1].

4.1 Překódování v input procesoru TEXu

Vstupní text pro formát CSplain se předpokládá 8bitový. Kódování češtiny nebo slovenštiny vstupního textu je takové, jaké běžně používáte v systému, kde je váš TEX instalován. O tomto kódování budeme nadále mluvit jako o *vstupním kódování*.

Ze vstupního kódování je třeba text překódovat do *vnitřního kódování*, se kterým interně pracuje TEX. Toto vnitřní kódování je v CSplainu implicitně nastaveno na ISO-8859-2 nezávisle na operačním systému. V tomto kódování musí být připraveny fonty použité v dokumentu. $\mathcal{CS}$fonty a fonty zaváděné pomocí `\input ctimes`, `\input cbookman` atd. tuto vlastnost mají.

Překódování textu do vnitřního kódování můžete udělat ručně před spuštěním TEXu (to moc nedoporučuji) nebo je tato konverze implementována do input procesoru samotného TEXu. Jak je to uděláno závisí na použité distribuci TEXu. Nejběžnější metody nastavení input procesoru budou zmíněny níže.

Problém vztahu vstupního a vnitřního kódování ilustruji na příkladě. Nechť je ve vstupním souboru napsáno `\char174=Ž`. Po zpracování $\mathcal{CS}$plainem musím na výstupu dostat: Ž=Ž. Tímto příkladem jsem naznačil dvě věci. Za prvé: některá makra mohou být závislá na zvoleném vnitřním kódování TEXu (zde makro `\char174`, které má vytisknout písmeno Ž). Pokud použiju v systému povel `csplain` bez doplňujících maker, pak předpokládám, že kód 174 znamená písmeno Ž.

Za druhé: pokud vpravo od rovnítka vidím v editoru, kterým zpracovávám vstupní dokument, písmeno Ž, pak se tento znak musí dostat do vnitřních částí TEXu pod kódem 174, ačkoli v tom editoru je třeba kódován úplně jinak. Konverze by měla být implementována (závisle na použitém vstupním kódování) v input procesoru TEXu. Důležité je, že to ve svém editoru *vidím jako Ž* a očekávám na základě této vizuální informace jednotné chování $\mathcal{CS}$plainu na všech systémech, kde existují editory, kterými se mohu do zdrojového textu podívat a vidět Ž (ačkoli různé operační systémy mohou toto Ž kódovat různě).

Konverzi do jednotného vnitřního kódování je možné realizovat v různých distribucích TEXu různě. Pročtěte si dokumentaci k použité distribuci. V emTEXu se pro tyto účely používá tzv. TCP tabulka zaváděná při inicializaci formátu. Ve velmi starých web2c distribucích TEXu (léta 1992–1997) můžete najít tzv. Škarvadovu záplatu, která nastavovala kódování podle proměnné prostředí systému. Později se pro překódování používal encTEX(rok 1997), který nastavoval xord/xchr vektor input procesoru TEXu pomocí přidaných primitivů (viz [6] a [5]). Dnes (po roce 1998) se v distribucích web2c, teTEX, TEXlive a odvozených používají tzv. TCX tabulky, které se vyvolají z příkazového řádku pomocí přepínače `-translate-file` nebo `-default-translate-file`.

Se správným nastavením překódovací tabulky úzce souvisí schopnost TEXu zapisovat 8bitový text do logů a do `\write` souborů (tam musí být text zpětně konvertován do vstupního kódování). TEX implicitně pro výstup do těchto souborů používá pro znaky s kódem větším než 127 hexadecimální přepis uvozený dvěma znaky `^`. To je pro $\mathcal{CS}$plain nepřípustné, a proto je nutné v TEXu rozšířit tzv. tabulku znaků povolených pro tisk do logů a pracovních souborů. Ta je ve web2c distribuci rozšířena automaticky zavedením TCX tabulky: znaky, které jsou v této tabulce zmíněny, se stávají povolenými pro tisk. To vysvětluje nutnost používání tabulky `il2-cs.tcx` v UNIXových distribucích, přestože tabulka pouze deklaruje překódování „jedna ku jedné".

Věnujte prosím při implementaci $\mathcal{CS}$plainu do jiných TEXových distribucí zvýšenou pozornost právě problémům překódování v input procesoru a problému množiny povolených znaků pro tisk do logů a pracovních souborů. Pokud se chcete přesvědčit o správnosti implementace $\mathcal{CS}$plainu, vyzkoušejte test cstrip (viz [8]).

4.2 Inicializace formátu

Pokud se vám podařilo $\mathcal{C}\mathcal{S}$plainem zpracovat tento dokument, pak máte formát $\mathcal{C}\mathcal{S}$plain již inicializovaný a můžete směle tuto sekci přeskočit. Pokud používáte nejnovější verzi TEXlive, pak stačí napsat na příkazový řádek `csplain dokument` a pokud tak činíte poprvé, formát se automaticky vytvoří. Také můžete inicializovat formát $\mathcal{C}\mathcal{S}$plain pomocí nástroje `texconfig` například tak, že odstraníte komentářové znaky u slova `csplain` v souboru `texmf/web2c/fmtutil.cnf` a spustíte `texconfig init`.

Následující text popisuje možnost vytvoření formátu „ručně". Příkazem `initex` označuji v tomto textu spuštění TEXu v inicializačním módu, kdy TEX dokáže načítat tabulky vzorů dělení pro různé jazyky a ukládat nabyté vědomosti příkazem `\dump` do binárních souborů, tzv. formátů. Dříve na to existoval zvláštní program iniTEX, dnes se často používá nějaký přepínač: `tex -ini`, `tex /i` atd.

Formát $\mathcal{C}\mathcal{S}$plain (soubor `csplain.fmt`) vygenerujete jednoduše:

```
initex csplain.ini
```

Používat jej pak můžete pomocí příkazu

```
tex &csplain dokument
```

V UNIXu musíte použít `tex \&csplain dokument`, aby se znak `&` neinterpretoval na úrovni shellu. Nezapomeňte ale nastavit správnou překódovací tabulku způsobem obvyklým v použité TEXové distribuci. Níže uvádím několik příkladů.

- **teTEX, TEXLive, vstupní kódování ISO-8859-2**:
  inicializace: `tex -ini csplain.ini`
  obsah skriptu `csplain`:
  `tex -fmt=csplain -default-translate-file=il2-cs $@`
  spuštění $\mathcal{C}\mathcal{S}$plainu: `csplain dokument`
- **teTEX, TEXLive, vstupní kódování podle Kamenických** (hypotetický příklad):
  inicializace: `tex -ini csplain.ini`
  obsah skriptu `csplain`:
  `tex -fmt=csplain -default-translate-file=kam-cs $@`
  spuštění $\mathcal{C}\mathcal{S}$plainu: `csplain dokument`
- **web2cTEX s encTEXem, vstupní kódování ISO-8859-2**:
  inicializace: `tex -ini csplain.ini`
  instalace povelu `csplain`: `ln -s tex csplain`
  spuštění $\mathcal{C}\mathcal{S}$plainu: `csplain dokument`
- **web2TEX s encTEXem, vstupní kódování podle Kamenických**:
  inicializace: `tex -ini '\let\enc=k \input csplain.ini'`
  instalace povelu `csplain`: `ln -s tex csplain`
  spuštění $\mathcal{C}\mathcal{S}$plainu: `csplain dokument`
- **emTEX, vstupní kódování podle Kamenických**:
  inicializace: `htex386 /i /8 /mt26000 /Ckamenic.tcp csplain.ini`

obsah dávky `csplain.bat`:

```
htex386 /mt26000 &csplain %1 %2 %3 %4 %5 %6 %7
```

Zastavím se na chvíli u emTEXu. Parametr `/mt` zde zvětšuje prostor pro ukládání vzorů dělení slov tak, aby se do formátu vešlo celkem pět vzorů dělení: anglické vzory dělení a dále české i slovenské vzory dělení obě v kódování ISO-8859-2 a Cork*. Paměť pro vzory dělení v TEXu se skládá ze dvou částí. První je nastavována interní TEXovou proměnnou trie_size (ta koresponduje s přepínačem `/mt`) a druhá proměnnou trie_op_size (viz [3], str. 301). Bohužel, hodnota trie_op_size není v emTEXu měnitelná a je ve verzích `tex.exe` a `tex386.exe` nastavena na malou hodnotu: pět vzorů dělení slov se tam nevejde. Proto jsem v příkladu použil `htex386.exe`, ve kterém je trie_op_size nastavena na dostatečně velkou hodnotu. Pokud chcete v emTEXu použít `tex.exe` nebo `tex386.exe`, pak musíte generovat formát CSplain jen se třemi vzory dělení – vyloučit alternativní kódování Cork. Toho lze dosáhnout jednak tím, že použijete verzi CSplainu starší než `<Feb. 2000>`, nebo tak, že potlačíte opakované volání vzorů dělení pro Cork pomocí `\let\Cork=\relax` takto:

```
tex /i/8/mt17000/Ckamenic.tcp \let\Cork=\relax \input csplain.ini
```

Poznamenejme, že potlačit načítání vzorů dělení v alternativním kódování (Cork) lze pomocí `\let\Cork=\relax` ve všech distribucích TEXu. Nejedná se tedy jen o emTEXovou záležitost. Nicméně v ostatních mě dostupných distribucích nejsou problémy s výchozí hodnotou paměti TEXu pro vzory dělení a je možno načíst bez nutnosti cokoli nastavovat všech pět vzorů dělení.

4.3 Výchozí nastavení v CSplainu

Aby byla zachována co největší kompatibilita s mezinárodním formátem plain a aby se necítili Češi nebo Slováci vzájemně utlačováni, jsou při startu CSplainu inicializovány americké vzory dělení slov a zapnuto větší mezerování za tečkami (tzv. `\nonfrenchspacing`).

Pro přepnutí na české vzory dělení slov použijte povel `\chyph` a do slovenštiny přepnete pomocí `\shyph`. Oba povely navíc zapnou stejnoměrné mezerování i za tečkou (tzv. `\frenchspacing`). Takové mezerování je v české a slovenské sazbě obvyklejší. Zpět na americké vzory dělení a větší mezerování za tečkou přejdete pomocí povelu `\ehyph`.

Příklad: Chceme-li v CSplainu zpracovat český dokument, stačí na začátku dokumentu uvést `\chyph`:

```
\chyph
Tady už bude vše fungovat česky.
\bye
```

Upozornění: nedoporučuji v dokumentech pro CSplain používat styl `czech.sty` resp. `slovak.sty`. Tím se totiž dokument zcela zbytečně stává závislým na externím

* Možnost zapnout kódování Cork jako alternativní vnitřní kódování CSplainu je implementována od verze `<Feb 2000>`. Podrobněji se o tom zmiňuji v sekci 4.5.

balíku maker, který není příliš stabilní. Pokud k tomu nemáte vážné důvody, styl *nepoužívejte*. Styl je vyvíjen spíše pro uživatele LaTeXu, zatímco uživatel CSplainu si vystačí s povely `\chyph` a `\shyph`. V budoucnu přestěhuji tyto styly z archivního souboru `csplain.tar.gz` (kam nepatří a jsou tam jen z historických důvodů) do archivu `cslatex.tar.gz`.

Rozměr tiskového zrcadla (výška a šířka textu) je implicitně nastaven tak, aby všechny čtyři okraje měly velikost 1 palec na papíru formátu A4. To je významný rozdíl oproti nastavení ve formátu plain, kde se sice také předpokládají okraje 1 in, ale na papíru US-letter. Níže uvádím tabulku nastavení rozměrů zrcadla v CSplainu a pro srovnání tytéž parametry v originálním plainu:

```
   CSplain                      plain
-----------------------------------------------------
\hsize   = 159.2mm          \hsize = 6.5in (165.1mm)
\vsize   = 239.2mm          \vsize = 8.9in (226.06mm)
\hoffset = 0mm              \hoffset = 0in
\voffset = 0mm              \voffset = 0in
```

Výchozí fonty zavedené do formátu CSplain jsou CSfonty. Pro matematiku jsou zavedeny původní Computer Modern fonty. Na rozdíl od plainu formát CSplain nezavádí do paměti při inicializaci desítky dalších fontů označených jako `\preloaded`, protože další fonty se dají zavést povelem `\font` až v době potřeby. Na starodávných strojích bylo možná užitečné zavést fonty „do rezervy“ už při inicializaci formátu, aby pak příkaz `\font` použitý v dokumentu moc nezdržoval. Při dnešních rychlostech počítačů je toto opatření zcela zbytečné, a proto v CSplainu nepoužité.

Matematická sazba funguje v CSplainu zcela stejně, jako v originálním plainu. Je to díky tomu, že implicitně zavedené CSfonty jsou konzervativním rozšířením Computer Modern fontů. Při zavádění jiných fontů do dokumentu je potřeba počítat při matematické sazbě s některými obtížemi (viz následující dvě sekce).

4.4 Použití PostScriptových fontů v CSplainu

Součástí CSTeXu v balíčku `cspsfonts.tar.gz` jsou kromě metrik také soubory, které předefinují textové fonty z výchozích CSfontů na fonty některé rodiny z „base 35“. Jak jsme již uvedli v předchozí kapitole, fonty z balíčku `cspsfonts.tar.gz` jsou kódovány v ISO-8859-2, tedy v souladu s vnitřním kódováním použitým v CSplainu. Níže je tabulka souborů, které zavádějí PostScriptové fonty do TeXu primitivem `\font`:

```
soubor           Rodina fontů
--------------------------------
cavantga.tex     Avantgarde Book
cbookman.tex     Bookman
chelvet.tex      Helvetica
cncent.tex       New Century
cpalatin.tex     Palatino
ctimes.tex       Times Roman
```

Chcete-li například přepnout do písma Bookman, stačí napsat na začátek dokumentu:

```
\input cbookman
```

Pokud sami pracujete s primitivem `\font` například pro zavedení větších velikostí fontů u nadpisů, doporučuji použít následující konstrukci:

```
\font\titulfont=\fontname\tenbf\space scaled \magstep2
```

Tato konstrukce není závislá na konkrétním fontu, takže když později změníte před takovou konstrukcí `\input bookman` například na `\input cpalatin`, změní se automaticky i font pro nadpisy.

Po zavedení rodiny PostScriptových fontů je potřeba dát pozor na matematickou sazbu. Jedná se o veškerou sazbu realizovanou ve zdrojovém textu mezi dolary. Ta i po změně textových fontů pracuje s původními $\mathcal{CS}$fonty a Computer Modern fonty. Jedině tak je dosaženo, že realizace všech matematických symbolů z plainu zůstává zachována.

Napíšete-li po zavedení nového textového fontu v textu například číslovku 2 bez použití dolarů, dostanete dvojku z nového fontu. Použijete-li ale `$2$`, na výstupu bude dvojka z $\mathcal{CS}$fontů. Toto míchání fontů není estetické. Chcete-li se mu aspoň částečně vyhnout, použijte po zavedení rodiny fontů povel `\setsimplemath`, například:

```
\input cbookman \setsimplemath
```

Příkaz zavede i pro fonty, které TEX používá při sazbě mezi dolary (TEXová rodina 0 a 1), odkazy na nově použité PostScriptové fonty. Tento příkaz je ale potřeba použít s velkou opatrností, protože bude fungovat zřejmě jen jednoduchá matematika. Například řecké symboly, které použité rodiny fontů neobsahují, nenávratně ztratíte. Navíc matematické symboly a natahovací závorky (TEXová rodina 2 a 3) zůstávají i nadále v Computer Modern, takže se míchání typů fontů zcela nevyhnete.

Pokud chcete používat složitější matematickou sazbu kombinovanou s PostScriptovými fonty, pak máte dvě možnosti:

1. Nastavit `\mathcode` všech matematických symbolů tak, aby byly použity například znaky z PostScriptového fontu Symbol. Matematické znaky z tohoto fontu jsou navrženy jako doplněk k fontu Times Roman, ale hodí se i k jiným PostScriptovým fontům. Bohužel, natahovací závorky a velké operátory (jedna ze specialit TEXu) v tomto fontu nejsou, takže tyto objekty je nutné ponechat v Computer Modern. Abyste nemuseli `\mathcode` pracně nastavovat, doporučuji použít makro OFS [9].

2. Zakoupit nějaký komerční matematický font včetně natahovacích závorek. Vhodný je například MathTimes firmy Y&Y. Pro šťastné majitele této sady fontů je v $\mathcal{CS}$plainu distribuován soubor `cmt.tex`, takže pokud na začátku dokumentu uvedete

```
\input ctimes
\input cmt
```

pak budete mít i matematickou sazbu v „Times-Roman stylu“ se vším všudy včetně řeckých symbolů a natahovacích závorek. Jen některé drobnosti (například skákavé číslice v plainu dosažené pomocí `\fam=1`) musíte udělat jinak. Přečtěte si manuál k zakoupené sadě fontů.

Matematické fonty v „Times-Roman stylu“ se dají kombinovat i s mnoha textovými PostScriptovými fonty dynamické antikvy s dostačujícím estetickým výsledkem. Podstatně horší je kombinovat dynamickou antikvu textového PostScriptového fontu se statickou antikvou Computer Modern. To je důvod, proč se nákup sady fontů MathTime vyplatí.

4.5 Použití fontů kódovaných podle Corku (T1 kódování) v CSplainu

Od verze CSplainu `<Feb. 2000>` načítá tento formát při inicializaci tabulky vzorů dělení nejen v ISO-8859-2, ale též v kódování podle Corku (tzv. T1 kódování). Tyto tabulky „čekají“ na případ, kdy uživatel bude chtít použít ve svém dokumentu fonty, jejichž metriky jsou kódované v T1. Může se totiž stát, že uživatel nebude chtít připravovat metriky a virtuální fonty pro nově zakoupené PostScriptové fonty ručně (pomocí programů `a2ac` a `afm2tfm` nebo pomocí `fontinst`), ale bude chtít využít již hotové metriky nabízené v TEXových archivech. Tyto hotové metriky jsou ale většinou v T1 kódování.

Upozorňuji na určitá omezení použití CSplainu s T1 kódovanými fonty:

- Všechny fonty v celém dokumentu musejí být kódovány jednotně v ISO-8859-2 nebo v T1. V dokumentu nelze fonty obou kódování jednoduše míchat.
- CSplain *implicitně* nastavuje vnitřní kódování do ISO-8859-2. Změníte-li tuto vlastnost, pak použitý povel v systému se už nesmí nazývat `csplain`.
- Změnit vnitřní kódování TEXu příkazem `%&` na prvním řádku dokumentu je možné ve web2c distribucích TEXu a odvozených. V jiných instalacích nemusí být tato vlastnost možná. Proto je třeba počítat s tím, že dokument opírající se o tento příkaz nemusí být zcela přenositelný na jiné implementace TEXu.

Přepnutí na vnitřní kódování podle Corku uděláte příkazem `\input t1code`. Kromě toho se musíte postarat o to, aby se vstupní kódování správně konvertovalo na vnitřní kódování TEXu. Například ve web2c distribuci použijete volbu `-translate-file`. Příklad použití T1 kódovaného fontu:

```
%&csplain -translate-file=il2-t1
\input t1code   %% Některé definice závislé na kódování
\chyph          %% Příkaz nyní zapne tabulku 15 místo tabulky 5
\font\f=ptmr8t  %% Zavedení T1 kódovaného fontu Times-Roman
\f
Tady je český text zpracovaný uvnitř \TeX{}u v kódování T1
včetně použití správné tabulky pro dělení slov.
\end
```

Takto připravený soubor TEXujte na web2c instalaci nikoli povelem `csplain`, ale raději povelem `tex`, tedy:

```
tex dokument
```

Program `tex` z web2c implementace se podívá do prvního řádku dokumentu a pokud tam najde dvojici `%&`, zavede formát a TCX tabulku (která pozměňuje xord/xchr vektor) podle toho, co je za touto dvojicí napsáno. Pokud byste použili skript `csplain` obsahující přepínač `-translate-file=il2-cs`, nebude to fungovat, protože přepínač na příkazovém řádku má vyšší prioritu než v dokumentu. V novějších implemetacích příkazu `csplain` je použit skript, který obsahuje přepínač `-default-translate-file=il2-cs`. Ten má prioritu nižší, takže pak lze při zpracování dokumentu s konstrukcí `%&` použít i příkaz `csplain`.

Do distribuce CSplainu jsou zařazeny soubory `dcfonts.tex`, `ecfonts.tex` a pro inspiraci též `ttimes.tex`. Pokud na začátku dokumentu například napíšete:

```
%&csplain -translate-file=il2-t1
\input ecfonts
\chyph
```

převádíte celou sazbu do EC fontů kódovaných podle Corku. V zaváděném souboru `ecfonts.tex` je proveden `\input t1code`, takže se o to přímo v dokumentu už není potřeba starat.

Při zavádění PostScriptových fontů (například použitím `ttimes.tex`) nastávají stejné problémy s matematickou sazbou, jako při použití PostScriptových fontů kódovaných v ISO-8859-2. Tyto problémy jsou popsány v předchozí sekci.

4.6 Nové řídicí sekvence CSplainu oproti Knuthovu plainu

`\austrian`
: Registr rezervovaný pro číslo tabulky vzorů dělení slov pro rakouskou němčinu (hodnota 2). Vzory dělení pro tento jazyk implicitně nejsou zavedeny.

`\cmaccents`
: Nastaví kontrolní sekvence `\^`, `` \` ``, `\'`, `\v` a `\"` tak, aby fungovaly stejně, jako v originálním plainu (použití primitivu `\accent`). Sekvence `\r` je nedefinována. Toto je defaultní nastavení. Pokud slovo obsahuje akcentovaný znak zapsaný některou z těchto sekvencí, pak pro ně nefunguje automatické dělení.

`\cmaccentsmesage`
: Způsobí výpis informace do logu a na terminál o použití příkazu `\cmaccents`. Pokud vás tato zpráva obtěžuje, nastavte `\let\cmaccentsmesage=\relax` před vyvoláním příkazu `\cmaccents`.

`\csaccents`
: Nastaví kontrolní sekvence `\^`, `` \` ``, `\'`, `\v`, `\"` a `\r` tak, aby expandovaly na odpovídající 8bitové reprezentace znaků podle ISO-8859-2 (po `\input t1code` pak expandují na kódování podle Corku). Sekvence `\r u` expanduje na ů. Pokud slovo obsahuje akcentovaný znak zapsaný některou z těchto sekvencí, pak pro ně funguje i automatické dělení slov.

`\csaccentsmesage`
: Způsobí výpis informace do logu a na terminál o použití příkazu `\csaccents`. Pokud vás tato zpráva obtěžuje, nastavte `\let\csaccentsmesage=\relax` před vyvoláním příkazu `\csaccents`.

\clqq

Vytiskne levé české uvozovky („).

\crqq

Vytiskne pravé české uvozovky (“).

\chyph

Aktivuje tabulku dělení slov podle registru \czech a nastaví stejnoměrné mezerování za tečkami (tzv. \frenchspacing).

\czech

Registr rezervovaný pro číslo tabulky vzorů dělení slov pro češtinu (hodnota \iltwoczech nebo \toneczech podle toho, zda nebylo nebo bylo použito \input t1code).

\ehyph

Aktivuje tabulku dělení slov podle registru \USenglish a nastaví větší mezerování za tečkami (tzv. \nonfrenchspacing). Toto nastavení je výchozí.

\english

Registr rezervovaný pro číslo tabulky vzorů dělení slov pro britskou angličtinu (hodnota 4). Vzory dělení pro tento jazyk implicitně nejsou zavedeny.

\extrahyphenchar

Alternativní znak spojovníku (-). Kód 156. Po \input t1code není toto makro definováno.

\extrahyphens

Nastaví rozdělovací znak pro běžné textové fonty na \extrahyphenchar. Defaultně je v plainu jako rozdělovací znak nastaven znak minus (ASCI 45), což může činit v českém textu komplikace. TEX totiž v případě rozdělovacího znaku přímo ve vstupním textu může dělit tak, že znak neopakuje na dalším řádku. Po \extrahyphens bude mít dělicí znak kód 156, což je kód, který se na vstupu nevyskytuje. Máme tím zaručeno, že ve slovech „je-li“ nebude TEX dělit vůbec. Jinou aplikací je použití speciální metriky nebo dokonce speciálního tvaru pro \extrahyphenchar.

\flqq

Levé francouzské uvozovky («), které se v češtině a němčině používají vpravo.

\frqq

Pravé francouzské uvozovky (»), které se v češtině a němčině používají vlevo, tedy »takto«.

\french

Registr rezervovaný pro číslo tabulky vzorů dělení slov pro francouzštinu (hodnota 3). Vzory dělení pro tento jazyk implicitně nejsou zavedeny.

\german

Registr rezervovaný pro číslo tabulky vzorů dělení slov pro němčinu (hodnota 1). Vzory dělení pro tento jazyk implicitně nejsou zavedeny.

\iltwoczech

Konstanta 5 rezervovaná pro číslo tabulky vzorů dělení slov pro češtinu při kódování fontů podle ISO-8859-2.

\iltwoslovak

Konstanta 6 rezervovaná pro číslo tabulky vzorů dělení slov pro slovenštinu při kódování fontů podle ISO-8859-2.

`\ogonek`
: Vytvoří polský akcent pod písmenem: ą.

`\promile`
: Vytiskne znak ‰.

`\shyph`
: Aktivuje tabulku dělení slov podle registru `\slovak` a nastaví stejnoměrné mezerování za tečkami (tzv. `\frenchspacing`).

`\slovak`
: Registr rezervovaný pro číslo tabulky vzorů dělení slov pro slovenštinu (hodnota `\iltwoslovak` nebo `\toneslovak` podle toho, zda nebylo nebo bylo použito `\input t1code`).

`\toneczech`
: Konstanta 15 rezervovaná pro číslo tabulky vzorů dělení slov pro češtinu při kódování fontů podle Corku.

`\toneslovak`
: Konstanta 16 rezervovaná pro číslo tabulky vzorů dělení slov pro slovenštinu při kódování fontů podle Corku.

`\USenglish`
: Registr rezervovaný pro číslo tabulky vzorů dělení slov pro americkou angličtinu (hodnota 0).

`\uv`
: Makro pro uvozovky. Používá se `\uv{takto}`. Makro je definováno tak, aby se uvnitř jeho argumentu dala použít verbatim konstrukce. To ale způsobí, že je potlačen kerning před pravou uvozovkou. Pokud nepoužíváte verbatim konstrukce v argumentu uvozovek, doporučuji makro předefinovat jednoduše na:

```
\long\def\uv#1{\clqq#1\crqq}.
```

Poznámka: CSplain nedefinuje makro pro ‚jednoduché' uvozovky, protože CSfonty neobsahují pro tyto uvozovky speciální znaky. Můžete ale místo nich použít čárku a apostrof třeba takto:

```
\long\def\singleuv#1{,#1'}
```

4.7 Přidání vzorů dělení slov dalších jazyků do CSplainu

CSplain implicitně načítá jen anglické vzory dělení (US) a české a slovenské. Od verze `<Feb. 2000>` navíc načítá české a slovenské vzory dělení nejen v kódování ISO-8859-2, ale i v kódování podle Corku. Anglické vzory dělení jsou implicitně aktivní a přepínáme na ně pomocí `\ehyph`. České vzory dělení se zapínají pomocí `\chyph` a slovenské pomocí `\shyph`.

Pokud potřebujete přidat další jazyk, pak před generováním formátu CSplain editujte soubor `hyphen.lan`. Tam najdete příklady doplnění vzorů dělení němčiny nebo francouzštiny (stačí odkomentovat odpovídající řádky). Podle těchto příkladů můžete analogicky zařadit potřebný další jazyk. Nezapomeňte v souboru `hyphen.lan` také definovat přepínač na nové vzory dělení. Pro němčinu definujeme například přepínač `ghyph` takto:

```
\def\ghyph{\language=\german
           \lccode`\'=`\'
           \frenchspacing
           \lefthyphenmin=2
           \righthyphenmin=2 }
```

Nakonec přegenerujte formát způsobem popsaným v sekci …

Vzory dělení západoevropských jazyků jsou většinou kódovány podle ISO-8859-1, přičemž kódování Cork je nadmnožinou tohoto kódování. Je tedy možné přepnout pomocí `\input t1code` na začátku dokumentu do Corku, použít fonty kódované podle Corku a pak přepínat mezi češtinou a třeba němčinou střídavě pomocí přepínačů `\chyph` a `\ghyph`.

Vzory dělení exotických jazyků jsou kódovány ve svém specifickém kódování, ke kterému musíme sehnat font stejně kódovaný. Po přepnutí do nového jazyka musíme přepnout i na tento font. Dále je nutno řešit otázku možné kolize vstupního kódování češtiny/slovenštiny se vstupním kódováním pro nový jazyk. Je proto bezpečnější si pro speciální znaky jazyka udělat makra, která expandují na znak podle kódování fontu, a psát vstupní text pomocí těchto maker.

4.8 CSplain a AMSTEX

AMSTEX můžete používat v plainu i CSplainu bez nutnosti generovat kvůli tomu formátový soubor. Máte-li dokument v AMSTEXu, napište jednoduše na jeho začátek:

```
\input amstex
```

a dále můžete dokument zpracovat příkazem CSplain.

4.9 pdfTEX + CSplain = pdfCSplain

Pokud vygenerujete formát CSplain pdfTEXem, je potřeba jej nazvat jinak, aby byl odlišen od formátu CSplain pro originální TEX. Proto je v CSTEXu zvoleno jméno `pdfcsplain.fmt`. Dále je potřeba připravit příkaz `pdfcsplain`, který spustí pdfTEX s formátem `pdfcsplain.fmt` a zajistí případné překódování na úrovni input procesoru pdfTEXu stejně, jako do dělá příkaz `csplain`. Pokud uživatel nepoužil ve svých makrech konstrukce měnící sazbu při použití pdfTEXu, měl by dostat příkazem `pdfcsplain` naprosto stejný výstup jako při použití příkazu `csplain`. Pouze s tím rozdílem, že výstup nebude v `dvi` souboru, ale v `pdf`.

Ve web2c TEXu a odvozených distribucích lze generovat formát příkazem

```
pdftex -ini -fmt pdfcsplain csplain.ini
```

a příkaz `pdfcsplain` v UNIXu implementovat jako skript s obsahem

```
pdftex -fmt pdfcsplain -default-translate-file=il2-cs $@
```

Toto je pouze příklad implementace. Na jiných operačních systémech a jiných TEXových distribucích se situace může trochu lišit.

Poznamenejme, že v teTEXu a odvozených distribucích většinou stačí napsat `pdfcsplain dokument`, a pokud jste `pdfcsplain` ještě nepoužili, vygeneruje se automaticky.

4.10 Historie a budoucnost CSplainu

CSplain vznikl v roce 1992 jako jednoduché a minimální rozšíření Knuthova plainu používající CSfonty a akceptující 8bitový vstup. Jeho vytvoření bylo motivováno zařazením do emTEXové distribuce, která se připravovala k rozesílání členům CSTUGu.

CSplain se opíral a stále opírá o starší makra `hyphen.lan` a `plaina4.tex`, která už měl Olin Ulrych vytvořena dříve. CSplain v době svého vzniku načítal české vzory dělení, které vytvořil Láďa Lhotka heuristicky bez použití slovníků a programu `patgen`. Já jsem pro CSplain vytvořil `csplain.ini` a makro `csfonts.tex`, umožňující při generování formátu číst přímo Knuthovo makro `plain.tex`, a přitom natáhnout místo CMfontů CSfonty.

V roce 1994 byly vzory dělení slov Ládi Lhotky vyměněny za nové české vzory dělení od Pavla Ševečka, který na to použil slovník a `patgen`. V rámci své firmy tyto vzory dělení prodává komerčním firmám pro potřeby DTP programů, jako byly Ventura, PageMaker nebo Quark. Dnes jsou tyto vzory dělení také například ve Wordu. Aby Pavel Ševeček odlišil volně šířené vzory dělení pro CSTUG od komerčně šířených, volně šířené vzory dělení mírně modifikoval. Tvrdí se, že běžný uživatel nepozná rozdíl v kvalitě vzorů dělení komerčních a volně šířených. Ševečkovy vzory dělení slov jsou výrazně kvalitnější než původní Lhotkovy, a proto jsme u těchto vzorů dělení zůstali.

Opuštěním Lhotkových vzorů dělení došlo k poslední změně v CSplainu, která může způsobit zpětnou nekompatibilitu: tj. dokumenty vytvořené v CSplainu před rokem 1994 mohly dopadnout jinak než dnes, protože některá slova mohla být rozdělena jinak. Od této doby je CSplain fixován a stabilní podobně, jako Knuthův plain.

Protože jsem autorem názvu CSplain, souborů `csplain.ini`, `csfonts.tex` a množství dokumentace k CSplainu a protože jej od jeho vzniku udržuji, rozhodl jsem se přísně dbát na zpětnou kompatibilitu. Změny do CSplainu dělám jen takové, které jsou opravdu nezbytné. To se stává jednou za několik let (viz historické poznámky v `csplain.ini`). Změny dělám tak, že pouze přidám další nejnutnější makra, ale stávající makra a jejich význam nechávám nezměněna. Uživatelům CSplainu ručím, že jejich dokumenty napsané v CSplainu a opírající se o neměnné fonty (např. CSfonty nebo base 35 PostScriptové fonty, metriky z CSTEXu), budou i v budoucnu CSplainem formátovány naprosto stejně, jako dnes.

Abych mohl takovou záruku uživatelům poskytnout, není CSplain zveřejněn pod GNU GPL, ale jedná se o licenci velmi podobnou Knuthově. Přesné znění licence je uvedeno na konci souboru `csplain.ini`. Zhruba řečeno, jedná se o „patent na název“. CSplain můžete svobodně distribuovat, používat a měnit, ale pokud jej změníte, nesmíte jej dále distribuovat pod názvem CSplain. Změny v CSplainu může dělat jen tzv. „současný administrátor CSTEXu“, což jsem zatím stále já. Pokud bych v budoucnu toto břímě někomu předal, pak určitě jen takovému člověku, který má na budoucnost CSplainu stejný názor jako já.

CSplain i nadále považuji za minimální rozšíření Knuthova plainu a nikdy do něj nepřidám žádné složitější makro vylepšující uživatelský komfort (jako například eplain). Zastávám názor, že uživatel plainu a CSplainu chce mít všechna makra pod svou vlastní kontrolou a raději si je udělá sám, než aby spoléhal na hotová, ale méně stabilní, řešení. Uživatel plainu/CSplainu si vytváří vlastní stále znovu používaná makra, která mu musí fungovat ve všech i budoucích verzích CSplainu. Proto se snažím CSplain pokud možno neměnit.

Lákavá je například změna definice uvozovek v CSplainu jednoduše na

```
\long\def\uv#1{\clqq#1\crqq}
```

aby fungoval automatický kerning s oběma stranami uvozovek. Tuto změnu ale nikdy v CSplainu neudělám, protože není zpětně kompatibilní se stávajícím řešením. Raději budu psát do omrzení do dokumentace, že taková jednoduchá definice je asi lepší, než ta z CSplainu, a že si ji každý může zařadit do svých maker.

Změnu v CSplainu z `<Feb. 2000>` považuji za asi největší, kterou jsem byl ochoten udělat. K zařazení alternativního kódování Cork (které samozřejmě není a nikdy nebude v CSplainu implicitní), mě motivovala skutečnost, že se kolem mě pohybovalo mnoho uživatelů plainu, kteří rádi používají fonty v tomto alternativním kódování. Svým krokem jsem jim umožnil používat CSplain, takže si nemusejí vytvářet své vlastní formáty.

5. Formát CSLaTeX

5.1 Různé LaTeXy

Do roku 1992 byl LaTeX udržován Leslie Lamportem. Naposledy měl jeho LaTeX verzi 2.09. Typické pro tento „starý“ LaTeX bylo použití příkazu `\documentstyle` namísto `\documentclass` v záhlaví dokumentu. Tato větev LaTeXu není dále udržována a podporována. Dnes převzali iniciativu nad LaTeXem Němci Frank Mittelbach a Rainer Schöpf a nazvali jej LaTeX 2_ε. Každý půlrok vytvářejí novou verzi (většinou obsahující jen opravy předchozí verze) a názvem dávají najevo, že se jedná o předchůdce LaTeXu 3, na němž už deset let pracují. Dokument napsaný pro LaTeX 2_ε poznáme vesměs podle toho, že začíná příkazem `\documentclass` místo `\documentstyle`. LaTeX 2_ε ovšem také akceptuje příkaz `\documentstyle`, přechází přitom do pokusu o emulaci starého LaTeXu 2.09, a navíc uživatele upozorní na to, že použil zastaralé záhlaví dokumentu.

V LaTeXu 2_ε je zabudovaná „časovaná bomba“ s rozbuškou na rok a půl. Ta se projeví při generování formátu, nikoli při běžném provozu. Pokud generujete formát ze zdrojů LaTeXu, které jsou starší než rok a půl, generování končí chybou a na terminálu se objeví varování, že máte instalován příliš starý LaTeX a že je vhodné si obstarat nový. Pokud na toto hlášení odpovíte klávesou Enter, formát se přesto vygeneruje.

V tomto manuálu se budeme dále zabývat jen dnes asi nejpoužívanějším LaTeXem 2_ε. Pojmem LaTeX tedy budeme rozumět LaTeX 2_ε. I za těchto okolností budeme nuceni rozlišovat mezi třemi „druhy“ LaTeXů:

- vanilla LaTeX
- babelizovaný LaTeX
- CSLaTeX

Vanilla LaTeX (nedotčený LaTeX) vzniká vygenerováním formátu LaTeX pouze za přítomnosti souborů z adresáře `latex/base`, tj. při generování nejsou čteny žádné soubory ovlivňující generování formátu. V distribuci tedy zcela chybí tyto soubory: `fonttext.cfg`, `fontmath.cfg`, `preload.cfg` a `hyphen.cfg`. Tento formát obsahuje pouze anglické dělení slov. Pro český nebo slovenský jazyk není tedy příliš použitelný. Není často používaný.

Babelizovaný LaTeX vzniká generováním formátu LaTeX za přítomnosti souborů `fonttext.cfg`, `fontmath.cfg` a `preload.cfg` z balíčku generic a `hyphen.cfg` z balíčku Babel. Tento formát obsahuje vzory dělení těch jazyků, které jsou v době generování formátu uvedeny v konfiguračním souboru Babelu s názvem `language.dat`. Je běžně používaný.

Formát CSLaTeX vzniká generováním formátu LaTeX za přítomnosti souborů `fonttext.cfg` a `hyphen.cfg` z balíčku CSLaTeX. Ostatní soubory ovlivňující generování formátu mohou chybět nebo být z balíčku generic. Obvykle je formát CSLaTeXu nazván `cslatex.fmt` a nikoli `latex.fmt`, aby bylo možno jej odlišit od běžně používaného babelizovaného LaTeXu. Z toho důvodu je v balíčku CSLaTeXu k dispozici soubor `cslatex.ini`, který provede `\input latex.ltx`. CSLaTeX obsahuje vzory dělení anglického, českého a slovenského jazyka. Navíc příkaz `cslatex` musí zajistit konverzi ze vstupního kódování (podle použitého systému) na vnitřní kódování ISO 8859-2 na úrovni vstupního preprocesoru TeXu. Tento princip je tedy shodný s CSplainem.

Je patrno, že absolutní jistotu o tom, jaký LaTeX vlastně používáme, získáme jen pečlivým pročtením logu po vygenerování formátu. Koncept, kdy vlastnosti formátu jsou závislé na obsahu nějakých dodatečných souborů v distribuci v době generování formátů, se mi nelíbí, ale nelze proti tomu nic dělat. Takto to navrhli tvůrci LaTeXu a je to tedy jejich věc. Důsledek tohoto rozhodnutí je, že pokud řekneme: „používám LaTeX“, nikdy není zcela přesně řečeno, co se tím vlastně myslí. Tím se tento koncept diametrálně liší od Knuthova plainu nebo též CSplainu.

Co se stane, pokud jsou například v TeXové distribuci přítomny hned dva soubory `hyphen.cfg` – jeden z balíčku Babel a druhý z balíčku CSLaTeX? Není-li taková situace v distribuci řešena speciální konfigurací, není definováno, který z těchto souborů se vlastně načte, a tudíž není dopředu známo, jaký vygenerujeme LaTeX.

V novějších web2c distribucích TeXu lze konfigurovat způsob prohledávání `texmf` stromu podle názvu použitého programu nebo podle názvu použitého (resp. generovaného) formátu. Nejčastěji jsou tyto rozdílnosti konfigurovány v souboru `web2c/texmf.cnf`. Tam je (mimo jiné) řečeno:

```
% cstex, from Petr Olsak
TEXINPUTS.cslatex= .;$TEXMF/tex/{cslatex,csplain,latex,generic,}//
TEXINPUTS.csplain= .;$TEXMF/tex/{csplain,plain,generic,}//
TEXINPUTS.pdfcslatex= \
          .;$TEXMF/{pdftex,tex}/{cslatex,csplain,latex,generic,}//
TEXINPUTS.pdfcsplain= \
          .;$TEXMF/{pdftex,cstex,tex}/{csplain,plain,generic,}//
```

což znamená, že pokud použijeme nebo generujeme formát CSLaTeX, pak se vstupní soubory přednostně čtou z adresáře `tex/cslatex` a `tex/csplain` a teprve potom v ostatních adresářích. Takže `hyphen.cfg` a `fonttext.cfg` si iniTeX vezme přednostně z adresáře `tex/cslatex`. Při provozu CSLaTeXu pak bude stylový soubor `czech.sty` nebo `slovak.sty` při použití formátu CSLaTeX přečten z adresáře `tex/csplain` a nikoli z adresáře `tex/generic/babel`, kde se nalézá soubor stejného jména, ovšem pro balík Babel.

Aby se hledací algoritmy přizpůsobily názvu generovaného formátu, je nutné tento formát při generování CSLaTeXu explicitně specifikovat na příkazový řádek. Nestačí tedy napsat:

```
$ tex -ini cslatex.ini
```

ale musíme psát

```
$ tex -ini -fmt cslatex cslatex.ini
```

Při provozu formátu pak stačí psát

```
$ tex -fmt cslatex dokument
```

Ve web2c distribuci jsou k babelizovanému LaTeXu v konfiguračním souboru `web2c/texmf.cnf` uvedeny následující řádky:

```
TEXINPUTS.latex = .;$TEXMF/tex/{latex,generic,}//
TEXINPUTS.pdflatex = .;$TEXMF/{pdftex,tex}/{latex,generic,}//
TEXINPUTS = .;$TEXMF/tex/{generic,}//
```

což znamená, že formát přednostně hledá vstupní soubory v adresáři `tex/generic` a tam je podadresář `babel`. Dokonce je babel upřednostněn i v případě, kdy neuvedeme na příkazové řádce žádný název formátu.

S uvedenou konfigurací je možné bezproblémové soužití babelizovaného LaTeXu s CSLaTeXem v jediné distribuci. CSLaTeX pak obvykle voláme skriptem `cslatex`, který provede `tex -fmt cslatex`, zatímco babelizovaný LaTeX voláme skriptem `latex`, což spustí `tex -fmt latex`.

Pokud nelze ve vaší TeXové distribuci konfigurovat prohledávací algoritmy podle názvu formátu, pak je možné soužití babelizovaného LaTeXu s CSLaTeXem za předpokladu, že budete dodržovat následující zásady:

- Babelizovaný LaTeX vygenerujte v době, kdy není v distribuci přítomen CSLaTeX. Tak máte jistotu, že iniTeX nenajde soubor `hyphen.cfg` z CSLaTeXu, ale z balíku Babel.

- $\mathcal{CS}$LaTeX pak vygenerujte nad adresářem `cslatex`. Aktuální adresář má totiž ve všech distribucích TeXu přednost před ostatními adresáři v `texmf` stromu. Proto si iniTeX přečte soubory `fonttext.cfg` a `hyphen.cfg` z balíčku $\mathcal{CS}$LaTeX.
- Po instalaci $\mathcal{CS}$LaTeXu do `texmf` stromu je nutno ještě vymazat soubory

  ```
  tex/generic/babel/czech.sty
  tex/generic/babel/slovak.sty
  ```

  protože soubory s těmito názvy jsou v `texmf` stromu dvojmo. Přitom v balíčku Babel je nikdy nevyužijete, pokud budete psát korektní záhlaví dokumentu pro babelizovaný LaTeX (o tom pojednám podrobněji v následující sekci). Na druhé straně soubory `czech.sty` a `slovak.sty` z $\mathcal{CS}$LaTeXu budete potřebovat velmi často, takže tyto soubory nemažte.

5.2 Záhlaví LaTeXového dokumentu

Pokud používáte $\mathcal{CS}$LaTeX, pak záhlaví dokumentu může mít tvar

```
\documentclass{article}
\usepackage{czech} % nebo \usepackage{slovak}
... další příkazy záhlaví dokumentu
\begin{document}
...
\end{document}
```

Jestliže zapomenete na `\usepackage{czech}` nebo `\usepackage{slovak}`, pak v dokumentu nebudou fungovat akcentovaná písmena, nebude zapnuto české ani slovenské dělení slov a automaticky generované názvy (kapitoly, sekce, obrázky, ...) nebudou přeloženy do národního jazyka. V tomto případě tedy $\mathcal{CS}$LaTeX bude pracovat stejně jako vanilla LaTeX.

Pokud naopak používáte babelizovaný LaTeX, pak záhlaví dokumentu může mít tvar

```
\documentclass{article}
\usepackage[czech]{babel} % nebo \usepackage[slovak]{babel}
\usepackage[T1]{fontenc}
\usepackage[kódování-vstupu]{inputenc}
... další příkazy záhlaví dokumentu
\begin{document}
...
\end{document}
```

Všimněte si jiného způsobu volání stylového souboru `czech` nebo `slovak`. V tomto případě se slovo `czech` či `slovak` zapíše pouze jako volba balíčku `babel`. Prakticky je to realizováno načtením souboru `czech.lfd` nebo `slovak.lfd`. Pro fungování Babelu tedy není vůbec nutná existence souboru `czech.sty` nebo `slovak.sty`. Tyto soubory v balíčku Babel sice existují, ale jejich funkce spočívá pouze v tom, že „vynadají“ uživateli za nesprávné použití příkazu `\usepackage` a dále načítají odpovídající `lfd` soubor.

Volba `[czech]` nebo `[slovak]` v balíčku Babel ještě nemusí zajistit přepnutí na odpovídající dělení slov. K tomu je navíc potřeba, aby byl babelizovaný LaTeX s těmito vzory dělení už vygenerován.

Na rozdíl od CSLaTeXu není v babelizovaném LaTeXu řešena synchronizace vstupního kódování dokumentu s vnitřním kódováním LaTeXu. LaTeX implicitně pracuje s vnitřním kódováním podle CM fontů (tzv. `OT1`), což pro většinu jazyků není užitečné. Je tedy třeba při použití babelizovaného LaTeXu přepnout do vnitřního kódování podle Corku (tzv. `T1`), protože v tomto kódování jsou načteny vzory dělení. Použijeme tedy balíček fontenc s volbou `[T1]`. Pak se automaticky použijí též fonty v tomto kódování. Babelizovaný LaTeX nepodporuje (na rozdíl od CSLaTeXu) žádné jiné vnitřní kódování vhodné pro češtinu nebo slovenštinu.

Dále je potřeba v babelizovaném LaTeXu na vnitřní kódování `[T1]` navázat vstupní kódování dokumentu. Pokud například máte dokument napsán v kódování ISO 8859-2, pak je potřeba použít balíček inputenc s volbou `[latin2]`. Pokud máte dokument v kódování MS Windows CP 1250, napište volbu `[cp1250]`.

5.3 Vlastnosti CSLaTeXu

Uvedeme podrobně rozdíly mezi vanilla LaTeXem a CSLaTeXem.

Oba formáty deklarují vnitřní kódování CM fontů jako implicitní (tzv. `OT1`) a oba načítají ještě deklaraci kódování podle Corku (tzv. `T1`). CSLaTeX navíc načítá deklaraci vnitřního kódování podle CSfontů (tzv. `IL2`).

CSLaTeX má ve svém souboru `hyphen.cfg` u příkazu `\DeclareLanguage` pro jazyky czech a slovak možnost použít kromě volby `IL2` ještě volbu `T1`. V novějších distribucích CSLaTeXu jsou již obě volby implicitně napsány. V takovém případě se při generování formátu načtou vzory dělení češtiny a slovenštiny nejen v kódování `IL2`, ale také v kódování `T1`. CSLaTeX dále definuje mechanismy přepínání mezi těmito vzory dělení poté, co uživatel přepne vnitřní kódování z `T1` na `IL2` nebo naopak standardními LaTeXovými prostředky. Kvůli tomu je v CSLaTeXu předefinováno interní makro LaTeXu `\DeclareFontEncoding`, protože vanilla LaTeX nepředpokládá, že je po přepnutí kódování nutné přepnout i vzory dělení slov.

V CSLaTeXu je navíc definováno makro `\splithyphens` a `\standardhyphens`. Po použití makra `\splithyphens` se nastaví znak `-` jako aktivní a funguje podobně jako `\discretionary{-}{-}{-}`. Znamená to, že slova jako „je-li“ se rozdělí správně česky: „je-/-li“. Spojovník se opakuje na následujícím řádku. Makro navíc složitě ošetřuje výskyt `--` a `---`, který tiskne „normálně“ jako za sebou následující znaky s kódem 45, které se promění v ligaturu pomlčky nebo dlouhé pomlčky. Makro `\standardhyphens` dává vše do původního stavu, tj. po jeho použití znak `-` není aktivní. (Uživatelé CSplainu mohou pro tyto potřeby použít makro podle [3] na straně 217.)

Příkaz `cslatex` musí také zajistit konverzi z kódování češtiny/slovenštiny obvyklé v použitém systému do vnitřního kódování ISO 8859-2 alias `IL2`. Ve web2c distribuci TEXu je proto ve skriptu `cslatex` resp. dávce `cslatex.bat` použit přepínač `-default-translate-file`. Pro systémy MS Windows je vedle tohoto přepínače použita hodnota `cp1250cs`, zatímco pro UNIXové systémy se používá hodnota `il2-cs`, což nastavuje konverzi „jedna ku jedné". V jiných distribucích se musí implementovat konverze na úrovni vstupního procesoru TEXu způsobem závislým na použité distribuci. Například v emTEXu je možné použít TCP tabulky.

5.4 Vlastnosti stylových souborů `czech.sty` a `slovak.sty`

V této sekci budu z důvodu stručnosti mluvit o češtině a souboru `czech.sty`, ale to samé samozřejmě platí i pro slovenštinu a stylový soubor `slovak.sty`.

Načtení souboru `czech.sty` z CSLATEXu pomocí `\usepackage{czech}` způsobí následující změny:

- Nastaví se vnitřní kódování LATEXu na `IL2`, takže se implicitně použijí CS fonty.
- Inicializuje se české dělení slov v kódování `IL2`.
- Mezerování se nastaví na `\frenchspacing`, tj. rovnoměrné mezerování mezi slovy i za tečkami.
- Nastaví se české názvy pro jména „Kapitola", „Obsah" atd. Viz následující tabulka.
- Makro `\today` expanduje na český datum.
- Definují se makra `\clqq` a `\crqq` pro dvojité české uvozovky, a to i jejich nouzová varianta pro případ, kdy není použito kódování `IL2`. Definují se též makra `\clq` a `\crq` pro jednoduché české uvozovky.
- Definuje se makro `\uv` jako `\def\uv#1{\clqq#1\crqq}`.
- Definují se přepínače `\csprimeson` a `\csprimesoff` (popis viz níže).

Tabulka automaticky generovaných slov LATEXu, která jsou změněna po načtení `czech.sty` na české názvy:

```
Původní název          czech.sty            slovak.sty
-------------------------------------------------------------
Preface                Předmluva            Predhovor
References             Reference            Literatúra
Abstract               Abstrakt             Abstrakt
Bibliography           Literatura           Literatúra
Chapter                Kapitola             Kapitola
Appendix               Příloha              Dodatok
Contents               Obsah                Obsah
List of Figures        Seznam obrázků       Zoznam obrázkov
List of Tables         Seznam tabulek       Zoznam tabuliek
Index                  Rejstřík             Register
Figure                 Obrázek              Obr.
Table                  Tabulka              Tabuľka
Part                   Část                 Časť
encl                   Příloha              Príloha
cc                     Na vědomí            cc.
```

`To`	`Komu`	`Pre`
`Page`	`Strana`	`Str.`
`see`	`viz`	`viď`
`see also`	`viz také`	`viď tiež`

Kromě toho styl `czech.sty` akceptuje následující volby:

`T1`
: místo implicitního vnitřního kódování LaTeXu `IL2` se použije kódování `T1`.

`IL2`
: nemusíte psát, je to implicitní volba.

`OT1`
: použije se vnitřní kódování podle CM fontů. Pak samozřejmě nefunguje dělení slov češtiny a slovenštiny.

`split`
: zapne `\splithyphens`.

`nocaptions`
: výstup makra `\today` a automaticky generovaná slova zůstanou v angličtině.

`olduv`
: použije se stará definice makra `\uv`, která umožňuje použít v argumentu verbatim konstrukce.

Například `\usepackage[split,olduv]{czech}` zapne navíc zdvojování spojovníku při rozdělení a definuje makro `\uv` tak, že jsou možné verbatim konstrukce uvnitř argumentu.

Nyní vysvětlím vlastnosti přepínačů `\csprimeson` a `\csprimesoff`. Po použití příkazu `\csprimeson` jsou přiděleny znakům „`‘`“ a „`’`“ aktivní kategorie, aby `‘‘takto po anglicku’’` zapsané uvozovky se převedly na „takové“ uvozovky. Rovněž `‘jednoduché’` anglické uvozovky jsou pak vytištěny ‚jednoduše‘. Makro `\csprimesoff` vrací vše do původního stavu, kdy jsou uvedené znaky neaktivní. Po načtení `czech.sty` je implicitně nastaveno `\csprimesoff`.

Uvedený popis chování stylového souboru `czech.sty` se týká jen případu, kdy je tento stylový soubor načten z CSLaTeXu. Pokud je načten z babelizovaného LaTeXu, pak se provede `\input czech.lfd`, takže veškeré definice „češtiny“ jsou v režii balíku Babel. Tam například vůbec není definováno makro `\uv`. Pokud je `czech.sty` načten z CSplainu, pak se provede `\chyph`, definuje se `\csprimeson`, `\csprimesof` a makro `\today` expanduje na datum po česku. Je-li načten `czech.sty` z originálního plainu, pak se provede totéž jen s tím rozdílem, že místo `\chyph` se objeví na terminálu varování o nemožnosti přepnout na české vzory dělení.

5.5 PostScriptové fonty v CSLaTeXu

Při práci s PostScriptovými fonty v CSLaTeXu stačí použít standardní nástroje NFSS. Například pro zapnutí do fontů rodiny Times Roman stačí napsat do záhlaví dokumentu `\usepackage{times}`. Protože CSLaTeX při `\usepackage{czech}` implicitně pracuje s vnitřním kódováním podle ISO 8859-2, použijí se v tomto případě virtuální fonty z balíčku `cspsfonts.tar.gz`. Z toho důvodu jsou v balíčku

CSLaTeXu přítomny potřebné `fd` soubory. Následující tabulka ukazuje parametry příkazu `\usepackage` pro rodiny fontů ze standardní skupiny 35 PostScriptových fontů.

```
Rodina fontů      parametr
--------------------------
Avantgarde Book   avant
Bookman           bookman
Helvetica         helvet
New Century       newcent
Palatino          palatino
Times Roman       times
```

5.6 Použití fontů kódovaných podle Corku (T1 kódování) v CSLaTeXu

Máte-li vzory dělení v CSLaTeXu načteny i pro kódování T1 (viz příkaz `\DeclareLanguage` v souboru `hyphen.cfg`), pak můžete místo implicitního vnitřního kódování `IL2` použít kódování `T1`. V takovém případě se automaticky použijí místo CSfontů fonty kódované podle Corku. Vnitřní kódování `T1` inicializujete zápisem:

```
\usepackage[T1]{czech} % nebo \usepackage[T1]{slovak}
```

Protože ale input procesor TeXu při použití příkazu `cslatex` konvertuje vstupní kódování dokumentu na ISO 8859-2, je v tomto případě nutné navázat na to další konverzí pomocí balíčku inputenc takto:

```
\usepackage[latin2]{inputenc}
```

Zde je potřeba vždy psát `[latin2]`, ať je vstupní kódování dokumentu jakékoli, protože vstupní procesor TeXu nám toto kódování převedl na ISO 8859-2.

Po zapnutí vnitřního kódování na `T1` lze použít PostScriptové fonty stejným způsobem jako předtím (například `\usepackage{times}`). Nyní se ale použijí metriky dodávané v mezinárodních distribucích TeXu a kódované podle Corku.

5.7 pdfTeX + CSLaTeX = pdfCSLaTeX

V případě spojení CSLaTeXu s pdfTeXem platí vše naprosto stejně, jako bylo řečeno v sekci „pdfTeX+ CSplain = pdfCSplain“. Nahraďte v této sekci slovo CSplain slovem CSLaTeX a slovo pdfCSplain slovem pdfCSLaTeX a přečtěte si tuto sekci ještě jednou.

5.8 Historie a budoucnost CSLaTeXu

CSLaTeX vytvořil zhruba v roce 1992 Jiří Zlatuška. Od něj pochází myšlenka načtení vzorů dělení stejného jazyka v různých kódováních a předefinování vnitřního LaTeXového makra `\DeclareFontEncoding`. Na své implementaci TeXu tehdy provozoval mimo jiné fonty kódované v KOI-8, takže přepínání vnitřního kódování

LaTeXu si vlastně udělal pro svoje potřeby. Jiří Zlatuška je také autorem maker `\splithyphens` a `\standardhyphens`. Veškerá makra napsal dobře dokumentovaná pro použití v systému docstrip. Svou práci zveřejnil pod licencí GPL mimo jiné podle jeho slov proto, že pokud to bude někoho zajímat, tak to může dále udržovat a zvelebovat podle svých vlastních představ. On sám se kvůli své zaneprázněnosti v jiné oblasti tímto problémem později zřejmě nezabýval.

Po schůzce tvůrců CSTeXu v roce 1993 převzal starost o CSLaTeX podle dohody Zdeněk Wagner, který vytvořil definice kódování `IL2`. Vytvořil také pro CSLaTeX definiční soubory `fd` jednak pro CSfonty a jednak pro PostScriptové fonty z balíčku `cspsfonts.tar.gz`. Od něj také pochází implementace CSLaTeXu pro LaTeX 2.09. V emTeXové distribuci CSTeXu je stále tato implementace obsažena (pod označením latex209).

Soubor `czech.sty` má asi podstatně delší historii než CSLaTeX. Pochází z dílny Olina Ulricha. který se zřejmě inspiroval podobným stylovým souborem pro německý jazyk. Olin rovněž vytvořil makra `\csprimeson` a `\csprimesoff`. Zdeněk Wagner pak převzal Olinův stylový soubor a upravil jej pro provoz v CSLaTeXu. Slovenskou část včetně vzorů dělení slov vytvořila Janka Chlebíková. Soubor `slovak.sty` je přesnou kopií souboru `czech.sty` s výjimkou slovensky špecifických částí.

V duchu licence GPL převzal zhruba v roce 1997 údržbu CSLaTeXu Jaroslav Šnajdr. Udělal několik úprav stylových souborů `czech.sty` a `slovak.sty` včetně přechodu na novou definici uvozovek, uvnitř jejichž argumentu nefungují verbatim konstrukce. Tím kuriózně způsobil, že CSLaTeXem od této doby nejde bez chyb formátovat český překlad úvodu do LaTeXu, který je pod názvem balíčku `csuvodlat.tar.gz` součástí dokumentace CSTeXu. Je to názorná ukázka toho, co může způsobit změna kódu, která nerespektuje zpětnou kompatibilitu. Pan Šnajdr rovněž napsal `html` dokumentaci k CSLaTeXu, která popisuje instalaci CSLaTeXu ze zdrojových souborů použitím docstripu. Použijete-li ale balíček `cslatex.tar.gz`, pak nemusíte docstrip aplikovat, protože vedle zdrojových souborů jsou tam už přítomny i všechny soubory, které vznikají po aplikaci docstripu.

Já osobně jsem o LaTeX a tím pádem CSLaTeX jevil od začátku malý zájem, protože celý projekt je závislý na LaTeXu samotném. Nemám tedy jistotu, jaké změny v něm současný LaTeX-team udělá a jaké budou existovat do budoucna potíže se zpětnou kompatibilitou. To je zásadní odlišnost od Knuthova plainu a TeXu samotného. Raději jsem se tedy do LaTeXových věcí nemíchal.

V roce 1999 jsem nicméně přidal pár řádek maker do souboru `czhyphen.tex` tak, aby byl použitelný v babelizovaném LaTeXu. Do té doby totiž tato větev LaTeXu používala Lhotkovy vzory dělení, zatímco v CSLaTeXu jsme už dávno měli daleko kvalitnější Ševečkovy vzory dělení. Tyto novější vzory dělení jsou totiž napsány za použití TeXových sekvencí, což je sice nezávislé na kódování češtiny, ale balíček Babel to implicitně nedokáže strávit a očekává vzory dělení v kódování `T1`. Upravený soubor jsem nazval Babelovsky: `czhyph.tex`, zatímco v CSTeXu zůstává původní soubor `czhyphen.tex`. Sjednocení názvů těchto souborů by stejně nevyřešilo nejednotnost vývoje CSLaTeXu a babelizovaného LaTeXu.

Na výborové schůzi v roce 1999 jsem dostal za úkol prověřit možnost spojení babelizovaného LaTeXu s CSLaTeXem. Neustálé dotazy začínajících uživatelů, kteří si pletou tyto dva LaTeXy, nás utvrzují v tom, že by se pro sloučení mělo něco udělat.

Analyzoval jsem proto makra Babelu a udělal návrh na možné zapracování funkcionality CSLaTeXu do Babelu. Domnívám se, že CSLaTeX klidně může přestat existovat, ale babelizovaný LaTeX musí bezpodmínečně převzít všechny vlastnosti CSLaTeXu tak, aby dokumenty dříve zpracovávané CSLaTeXem byly naprosto stejně a bez jediné úpravy zpracované novým babelizovaným LaTeXem. Kvůli tomuto požadavku musí babelizovaný LaTeX umět načítat vzory dělení stejného jazyka ve více kódováních, jako to nyní dělá CSLaTeX. Dospěl jsem k závěru, že čistým řešením tohoto problému je jedině zásah do jádra LaTeXu samotného, aby dokázal při změně kódování fontů přepnout automaticky i vzory dělení. Zlatuška kvůli tomu předefinoval makro jádra LaTeXu `\DeclareFontEncoding`. Tato záplata, či jinak řečeno odmítnutí původního kódu tohoto makra, je na úrovni Babelu podle mého názoru velmi nečisté řešení. Skutečnost, že přepínání vzorů dělení při přepnutí kódování fontů LaTeXové jádro neřeší, považuji totiž za chybu LaTeXu. V roce 1999 jsem tedy požádal LaTeX-team, aby zapracoval změnu v duchu Zlatuškova návrhu do LaTeXového jádra. Můj návrh nebyl LaTeX-teamem akceptován. Za těchto okolností nejsem schopen zapracovat funkcionalitu CSLaTeXu do Babelu, protože to prostě nejde. Uživatelé LaTeXu se budou muset nadále potýkat s tím, že jejich oblíbený formát trpí určitou schizofrenií.

Společně se sloučením CSLaTeXu s Babelem jsem připravoval zásadní revizi stylů `czech.sty` a `slovak.sty` – v podstatě jsem měl v úmyslu jejich totální přepsání. Tyto stylové soubory obsahují množství reliktů z dob minulých, plno zcela nepoužívaných větví ve složitém větvení pomocí `\if` a stávají se totálně nepřehlednými. Protože ale ke sloučení CSLaTeXu s Babelem nakonec nedošlo, upustil jsem zatím od plánu pracovat na těch stylových souborech. Není ale vyloučeno, že k tomu dojde v budoucnosti. V takovém případě počítám s tím, že makra `\splithyphens` a `\standardhyphens` přesunu z formátu do stylového souboru, kam přirozeně patří. Dokumenty, které tato makra používají, a přitom nemají v záhlaví `\usepackage{czech}` ani `\usepackage{slovak}`, pak nebudou fungovat. Předpokládám, že takových dokumentů není mnoho, protože CSLaTeX a stylové soubory jsou většinou používány současně.

Protože pan Šnajdr se přestal CSLaTeXem zabývat, byl jsem nucen v roce 2002 zanést do stylových souborů jednu opravu podle požadavku pana Kubena. Neznamená to ale, že bych se ujal iniciativy nad CSLaTeXem. Jak jsem už vysvětlil, jsem ochoten převzít iniciativu jen tehdy, když bude LaTeXové jádro umět přepínat mezi různě kódovanými vzory dělení stejného jazyka. Přitom členové LaTeX-teamu jsou toho názoru, že to možná bude zapracováno až do LaTeXu 3.

6. Reference

[1] Donald Knuth. *The TeXbook.* Addison Wesley Publishing Company. Eleventh printing, revised, May 1991, ISBN 0-201-13447-0

[2] Petr Olšák. *Typografický systém TeX.* Konvoj, Brno 2000, ISBN 80-85615-91-6

[3] Petr Olšák. *TeXbook naruby.* Konvoj, Brno 2001, ISBN 80-7302-007-6.

[4] Petr Olšák. *První setkání s TEXem.* Volně šířený dokument ve formátech `tex`, `ps`, `pdf` na `ftp://math.feld.cvut.cz/pub/cstex/doc/prvni.*`, 22 stran.

[5] Petr Olšák. *Putování písmene ř z klávesy na papír.* Zpravodaj CSTUGu, 3/1997, strany 109–118.

[6] Petr Olšák. *Rozšíření TEXu encTEX.* Rozšíření ve formě změnového souboru k `tex.web` je volně šířeno na `ftp://math.feld.cvut.cz/pub/olsak/enctex/`.

[7] Petr Olšák. *Program* `a2ac`. Program včetně zdrojových kódů v jazyce C je volně šířen na `ftp://math.feld.cvut.cz/pub/olsak/a2ac/`.

[8] Petr Olšák. *Test* `cstrip`. Test je nepovinnou součástí CSTEXu. Je volně šířen na `ftp://math.feld.cvut.cz/pub/cstex/base/cstrip.tar.gz`.

[9] Petr Olšák. *Makro OFS.* TEXové makro OFS pro práci s rozsáhlými kolekcemi fontů je volně šířeno na `ftp://math.feld.cvut.cz/pub/olsak/ofs/`.