

Manuál k \LaTeX

Petr Olšák

Hněvanice, Morávka, Praha – léto 2002

Autor $\text{T}_{\text{E}}\text{X}$ u je profesor Donald Knuth.

$\text{T}_{\text{E}}\text{X}$ 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 $\mathcal{C}\mathcal{S}\text{T}_{\text{E}}\text{X}$. Jedná se o referenční manuál k tomuto balíku. Text je k dispozici ve formátech `tex` ($\mathcal{C}\mathcal{S}\text{plain}$), 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 $\mathcal{C}\mathcal{S}\text{T}_{\text{E}}\text{X}$ package. It is a reference manual of the $\mathcal{C}\mathcal{S}\text{T}_{\text{E}}\text{X}$. It is available in formats `tex` ($\mathcal{C}\mathcal{S}\text{plain}$), 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

© RNDr. Petr Olšák, 2002

Obsah

1. Úvod	4
2. $\mathcal{C}\mathcal{S}$ fonty	5
2.1 Odlíšnosti od CM fontů	7
2.2 Struktura METAFONTových zdrojových souborů $\mathcal{C}\mathcal{S}$ fontů	8
2.3 Alternativní hyphenchar	9
2.4 Anabáze UNIXových hard-linků	9
2.5 Virtuální fonty přidané do balíčku $\mathcal{C}\mathcal{S}$ fontů	10
2.6 Historie a budoucnost $\mathcal{C}\mathcal{S}$ fontů	11
3. <code>cspfonts.tar.gz</code> – 35 základních PostScriptových fontů	13
3.1 Historie a budoucnost balíčku <code>cspfonts.tar.gz</code>	16
4. Formát $\mathcal{C}\mathcal{S}$ plain	16
4.1 Překódování v input procesoru $\mathcal{T}\mathcal{E}\mathcal{X}$ u	16
4.2 Inicializace formátu	18
4.3 Výchozí nastavení v $\mathcal{C}\mathcal{S}$ plainu	19
4.4 Použití PostScriptových fontů v $\mathcal{C}\mathcal{S}$ plainu	20
4.5 Použití fontů kódovaných podle Corku (T1 kódování) v $\mathcal{C}\mathcal{S}$ plainu	22
4.6 Nové řídicí sekvence $\mathcal{C}\mathcal{S}$ plainu oproti Knuthovu plainu	23
4.7 Přidání vzorů dělení slov dalších jazyků do $\mathcal{C}\mathcal{S}$ plainu	25
4.8 $\mathcal{C}\mathcal{S}$ plain a $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$	26
4.9 <code>pdf$\mathcal{T}\mathcal{E}\mathcal{X}$</code> + $\mathcal{C}\mathcal{S}$ plain = <code>pdf$\mathcal{C}\mathcal{S}$plain</code>	26
4.10 Historie a budoucnost $\mathcal{C}\mathcal{S}$ plainu	27
5. Formát $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$	28
5.1 Různé $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ y	28
5.2 Záhlaví $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ ového dokumentu	31
5.3 Vlastnosti $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u	32
5.4 Vlastnosti stylových souborů <code>czech.sty</code> a <code>slovak.sty</code>	33
5.5 PostScriptové fonty v $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u	34
5.6 Použití fontů kódovaných podle Corku (T1 kódování) v $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u	35
5.7 <code>pdf$\mathcal{T}\mathcal{E}\mathcal{X}$</code> + $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ = <code>pdf$\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$</code>	35
5.8 Historie a budoucnost $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u	35
6. Reference	37

1. Úvod

V tomto manuálu se pokusím vyčerpávajícím způsobem popsat vlastnosti $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$. Z tohoto důvodu to asi nebude jednoduché čtení pro $\mathcal{T}\mathcal{E}\mathcal{X}$ ové začátečníky. Těm doporučuji nejprve přečíst [4].

$\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ je sada $\mathcal{T}\mathcal{E}\mathcal{X}$ ových maker, fontů, vzorů dělení slov a doplňujícího software pro podporu české a slovenské sazby v $\mathcal{T}\mathcal{E}\mathcal{X}$ u. Je vytvořen tak, aby mohl být použit na libovolné $\mathcal{T}\mathcal{E}\mathcal{X}$ ové distribuci na libovolném operačním systému.

Dříve byla slovem $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ označována také kompletní $\mathit{em}\mathcal{T}\mathcal{E}\mathcal{X}$ ová distribuce doplněná o zmíněná makra, fonty a vzory dělení. To bylo v roce 1993, kdy sdružení $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}$ rozesílalo pod tímto názvem $\mathit{em}\mathcal{T}\mathcal{E}\mathcal{X}$ ovou distribuci na disketách svým členům. V té době většina členů používala DOS, takže $\mathit{em}\mathcal{T}\mathcal{E}\mathcal{X}$ pro DOS byl pro ně vyhovující. V dnešní době uživatelé pracují s nejrůznějšími operačními systémy a s různými distribucemi $\mathcal{T}\mathcal{E}\mathcal{X}$ u pro tyto systémy. Z toho důvodu se v tomto manuálu přidržíme jen užšího významu slova $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$, definovaného v předchozím odstavci.

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

Základní balíčky $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u 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.
cspspfonts.tar.gz ... Virtuální fonty základních 35 PostScriptových
                  fontů v kódování podle CSfontů.
```

Pokud někdo řekne, že má na své distribuci $\mathcal{T}\mathcal{E}\mathcal{X}$ u instalován $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$, pak to znamená, že má určité instalovány tyto čtyři balíky. Kromě toho k $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u řadíme i následující doplňující software, který už nemusejí mít všichni uživatelé $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u instalován:

```
csfonts-t1.tar.gz .. Varianta CSfontů ve formátu PostScript Type1.
entex.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.
vlana.tar.gz      ... Zdrojové soubory v jazyce C a v jazyce cweb
                  programu vlana 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 (T_EX directory standard), takže pokud vaše distribuce T_EXu používá tento standard, pak rozbalení balíčků nad adresářem `texmf` povede automaticky k zařazení všech souborů na správná místa v `texmf` stromu.

Starší verze uvedených balíčků najdete na

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

Dokumentaci k C_ST_EXu (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 emT_EXu s podporou C_ST_EXu (v podadresáři `emtex`), minimalizovanou distribuci emT_EXu+C_ST_EXu pro pouhé tři instalační diskety (v podadresáři `minitex`), návody a software na instalaci web2c T_EXu (v podadresáři `web2c`) a RPM balíky teT_EXu podporující C_ST_EX pro různé distribuce Linuxu (v podadresáři `tetex-rpm`). Tyto podadresáře vznikaly postupně v poslední dekádě minulého tisíciletí podle potřeby a zájmu o uvedené distribuce T_EXu. 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 T_EXu obsahujících C_ST_EX – 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. C_Sfonty

C_Sfonty jsou konzervativním rozšířením CM fontů Donalda Knutha. Tím je míněno, že každý C_Sfont 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. C_Sfont 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 C_Sfontu. Druhý znak na pozicích 0B až 0F se vyskytuje namísto prvního znaku ve fontech bez ligatur typu `fi` (`csr5`, `cstt*` a `csslftt*`), druhý znak na pozicích 20, 3C, 3E, 5C, 5F a 7B až 7D najdeme ve strojopisných fontech (`cstt*` a `csslftt*`) 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 _j	ffl _z
1x	ı	ı	`	´	˘	˙	-	°	˚	ß	æ	œ	ø	Æ	Œ	Ø
2x	¬	!	”	#	\$£	%	&	'	()	*	+	,	-	.	/	
3x	0	1	2	3	4	5	6	7	8	9	:	;	i<	=	i>	?
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	[“\]	^	˘
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														% ₀₀		
9x									À				-	˘	«	»
Ax						Ĺ			Š		Ť				Ž	
Bx						ı			à	š		ť			ž	
Cx	Ř	Á			Ä	Ĺ			Č	É			Ě	Í		Ď
Dx			Ň	Ó	Ô		Ö		Ř	Ů	Ú		Û	Ý		
Ex	ı	á			ä	ı			č	é			ě	ı		ď
Fx			ň	ó	ô		ö		ř	ů	ú		ü	ý	„	“

Následuje seznam všech CM fontů. Pokud není vpravo uveden alternativní název \mathcal{C} fontu, jedná se o matematický font, který nemá v \mathcal{C} fontech alternativu. Také ji nepotřebuje.

CM font

CSfont

cmr17, 12, 10, 9, 8, 7, 6, 5
cmbx12, 10, 9, 8, 7, 6, 5
cmsl12, 10, 9, 8
cmtt12, 10, 9, 8
cmsl12, 10, 9, 8
cmss17, 12, 10, 9, 8
cmssi17, 12, 10, 9, 8
cmssdc10, cmssbx10
cmssqi8, cmssq8
cmdunh10, cmbxsl10, cmb10
cmff10, cmfib10

csr17, 12, 10, 9, 8, 7, 6, 5
csbx12, 10, 9, 8, 7, 6, 5
cssl12, 10, 9, 8
cstt12, 10, 9, 8
cssl12, 10, 9, 8
csss17, 12, 10, 9, 8
csssi17, 12, 10, 9, 8
csssdc10, csssbx10
csssqi8, csssq8
csdunh10, csbxsl10, csb10
csff10, csfib10

cmti12, 10, 9, 8, 7
cmbxti10, cmitt10
cmu10, cmfi10

csti12, 10, 9, 8, 7
csbxti10, csitt10
csu10, csfi10

cmcsc10, cmtcsc10

cscsc10, cstcsc10

cminch10

csinch10

Dvojice P. a P, – `csr10`: –0,0027777pt, `cmr10` není.

Dvojice F., F., V., V., W. a W, – `csr10`: –0,0055555pt, `cmr10` není.

Dvojice Av a Aw – `csr10`: –0,011111pt, `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 zao-krouhlení, která způsobí odlišnost výšek od výšek v `cmr10` maximálně o 0,007779pt. Jedná se o tyto znaky:

- Γ až Ω , \mathcal{A} , \mathcal{E} a všechny kapitálky: v `csr10` jsou menší o 0,00773pt.
- Nadržítka (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 \mathcal{C} fontů

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 \mathcal{C} fontech je takových hlavních souborů celkem 121. Všechny mají stejný dvouřádkový obsah:

```
input ccode
use_driver;
```

V souboru `ccode.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 `ccode.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 \mathcal{C} fontů. Kromě souboru `kmroman.mf` jako alternativy k souboru `roman.mf` z CM fontů najdeme analogické alternativy s názvy `kmtexit.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, í, ú, ý,
- `csacutu.mf` generuje Á, É, Í, Ó, Ř, 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, û, à, í, é, ð, ï, ä, ö, ü,
- `csotheru.mf` generuje Ô, Û, À, É, Ð, Ì, 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 \mathcal{C} fontech 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 $\text{T}_{\text{E}}\text{X}$ dělit slovo. Bez tohoto nastavení by $\text{T}_{\text{E}}\text{X}$ 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 \mathcal{C} fontu můžete pro vlastní potřeby modifikovat například programy `tftopl` a `pltotf`. Takto modifikovanou metriku ovšem nesmíte distribuovat pod stejným názvem, jaký má původní metrika \mathcal{C} fontu.

2.4 Anabáze UNIXových hard-linků

Jak jsme uvedli v předchozí sekci, hlavní METAFONTové soubory \mathcal{C} fontů 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 `cs-fonts.tar.gz` a dal k dispozici internetové veřejnosti.

Pokud je systém, na který se balíček `cs-fonts.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 rozbálí z těch 121 souborů jen jeden, přičemž používají jakýsi `Wintar`. Odpovídám: nechtě si tito uživatelé stěžují u svého dodavatele implementace programu `tar`. Odmítám kvůli Windowsovým uživatelům balit \mathcal{C} Sfonty jinak a opustit tak možnost šetření inodů na UNIXových systémech. Navíc se mi doneslo, že Windowsovi 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 \mathcal{C} Sfontů

V balíčku `csfonts.tar.gz` jsou přítomny dvě skupiny virtuálních fontů. Jedny mapují \mathcal{C} Sfonty na CM fonty a druhé mapují CM fonty na \mathcal{C} Sfonty. 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 \mathcal{C} Sfonty 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 \mathcal{C} S \TeX u obvykle používá \mathcal{C} Sfonty, 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 \mathcal{C} Sfonty jsou nadmnožinou CM fontů a bitmapy \mathcal{C} Sfontů pravděpodobně na disku už vygenerované jsou. Stačí pro zobrazení znaků CM fontů čerpat z bitmap \mathcal{C} Sfontů.

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 \mathcal{C} Sfonty. Takže `dvi`-ovladač bude tyto znaky nakonec čerpat z \mathcal{C} Sfontů. K žádnému zkreslení informace přitom nedochází, protože všechny znaky CM fontů jsou v odpovídajících \mathcal{C} Sfontech na stejných pozicích a vypadají úplně stejně.

Pokud \mathcal{C} Sfonty 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 \TeX ových distribucích, kde většina uživatelů \mathcal{C} Sfonty 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 \mathcal{C} Sfontů.

Druhá skupina virtuálních fontů s označením `cs2cm` není po rozbalení balíčku s \mathcal{C} Sfonty 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{C} ST \mathcal{E} Xu dokument odkazující na \mathcal{C} Sfonty, a nyní jej chceme ve formátovaném tvaru (jako soubor `dvi`) poslat například do zahraničí. Přitom předpokládáme, že příjemce nebude mít ve své \mathcal{T} \mathcal{E} Xové distribuci \mathcal{C} Sfonty.

Virtuální fonty `cs2cm` mapují všechny znaky \mathcal{C} Sfontů 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 \mathcal{T} \mathcal{E} Xové 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{C} Sfontech jsou kresleny společně s písmenem s velkou 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 \mathcal{T} \mathcal{E} Xu v UNIXu můžete pro uvedenou konverzi `dvi` souboru z \mathcal{C} Sfontů 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{C} Sfontů

Tvary akcentů \mathcal{C} Sfontů byly vytvořeny a implementovány v jazyce METAFONTu Petrem Novákem ve spolupráci s českými typografy někdy na začátku 90. let. Autor přenechal \mathcal{C} Sfonty \mathcal{C} STUGu, 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. Pracoval tam možnost nastavení kódování fontu a vytvořil makra umožňující mít všechny hlavní `mf` soubory se stejným dvouřádkovým obsahem.

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

V roce 1993 jsem převzal údržbu \mathcal{C} Sfontů 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 Ě, Y, Ý). Vě, Vř, Vö, Vü (analogicky pro F, W) a redukoval jsem přílišné záporné kerny: Té, Tó, Tů, Tí, Tá, Tú (analogicky pro Ě, Y, Ý).

Pak jsem vývoj \mathcal{C} fontů zmrazil podobným způsobem, jako Knuth přestal měnit CM fonty. Prioritním požadavkem je, aby dokument opírající se o \mathcal{C} fonty 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 \mathcal{C} fontů přidal virtuální fonty podporující náhradu Computer Modern fonty a naopak.

V roce 1998 se podařilo dohodnout s autorem $\text{te}\TeX\text{u}$ Thomassem Esserem, aby zařadil do své distribuce \mathcal{C} fonty a celý $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$. Od této chvíle jsou distribuce odvozené z $\text{te}\TeX\text{u}$ implicitně vybaveny \mathcal{C} fonty.

V roce 1998 jsem také pro potřeby výstupu do formátu PDF vytvořil variantu \mathcal{C} fontů, 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 \mathcal{C} fontů METAFONTem. Na mnoha místech jsem byl nucen přistoupit k mikrotypografickým kompromisům – v drobnostech se kresby některých znaků \mathcal{C} fontů z Type1 liší od svých originálních METAFONTových protějšků. Proto jsem distribuci Type1 \mathcal{C} fontů 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 \mathcal{C} fonty byly původně konfigurovány jen pro `pdfTEX`.

Rozhodnutí ponechat `dvips` pracovat implicitně s bitmapami naráželo na problémy. Neustále dokola se uživatelé 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 $\text{te}\TeX\text{u}$ rozhodl, že bude \mathcal{C} fonty 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 \mathcal{C} fontech a s výsledkem jsem nebyl vůbec spokojen: výsledné `pfb` soubory byly asi pětikrát větší než ty moje „ručně“ vyrobené. Proto jsem zatím alpha verzi Type1 formátu \mathcal{C} fontů z roku 1998 neopustil.

Do budoucna bych velmi rád do \mathcal{C} fontů přidal znak euro a paragraf. Taková změna by byla zpětně kompatibilní, takže bych se jí nebránil. METAFONTové zdroje pro paragraf ověřené na všech \mathcal{C} fontech 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 \mathcal{C} fontů 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{C} 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{C} 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{C} TeXu.

Kódování těchto metrik je inspirováno kódováním \mathcal{C} fontů, ale chybí matematické znaky ze začátku tabulky a znaky, které v běžných PostScriptových fontech nenajdeme (např. pozice 20: škrátka 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{C} TeX:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x													fi	fl		
1x	ı		`	´	˘	˙	-	°	˚	ß	æ	œ	ø	Æ	Œ	Ø
2x		!	”	#	\$	%	&	'	()	*	+	,	-	.	/
3x	0	1	2	3	4	5	6	7	8	9	:	;	i<	=	i>	?
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	[“\]	^	·
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	...	†	‡	•	£	¶								%o		
9x									À				-	ˆ	«	»
Ax			˘	Ł	ł	Ł		§	Š		Ť				Ž	
Bx	°			ł		ł			à	š		ť			ž	
Cx	Ř	Á	Â		Ä	Ĺ		Ç	Č	É		È	Ë	Í	Î	Ď
Dx			Ň	Ó	Ô		Ö		Ř	Ů	Ú		Û	Ý		
Ex	í	á	â		ä	Í		ç	č	é		ë	ě	í	î	ď
Fx			ň	ó	ô		ö		ř	ů	ú		ü	ý	„	“

Fonty rodiny Courier mají kódování inspirováno \mathcal{C} 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

`cspfonts.tar.gz`, které je nazváno 8z. Tuto nejednotnost kódování známe už u $\mathcal{C}\mathcal{S}$ 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 $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u, 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 $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ em.

Sada metrik balíčku `cspfonts.tar.gz` zahrnuje následující fonty:

Metrika	odkazuje na	Odpovídající PostScriptový font
<code>pagk8z</code>	<code>rpagk</code>	AvantGarde-Book
<code>pagko8z</code>	<code>rpagko</code>	AvantGarde-BookOblique
<code>pagd8z</code>	<code>rpagd</code>	AvantGarde-Demi
<code>pagdo8z</code>	<code>rpagdo</code>	AvantGarde-DemiOblique
<code>pagkc8z</code>	* <code>rpagk</code>	AvantGarde-Book Caps & Small Caps
<code>pagdc8z</code>	* <code>rpagd</code>	AvantGarde-Demi Caps & Small Caps
<code>pbkl8z</code>	<code>rpbkl</code>	Bookman-Light
<code>pbkli8z</code>	<code>rpbkli</code>	Bookman-LightItalic
<code>pbkd8z</code>	<code>rpbkd</code>	Bookman-Demi
<code>pbkdi8z</code>	<code>rpbkdi</code>	Bookman-DemiItalic
<code>pbklc8z</code>	* <code>rpbkl</code>	Bookman-Light Caps & Small Caps
<code>pbkdc8z</code>	* <code>rpbkd</code>	Bookman-Demi Caps & Small Caps
<code>pcrr8u</code>	<code>rpcrr</code>	Courier
<code>pcrrro8u</code>	<code>rpcrrro</code>	Courier-Oblique
<code>pcrb8u</code>	<code>rpcrb</code>	Courier-Bold
<code>pcrbo8u</code>	<code>rpcrbo</code>	Courier-BoldOblique
<code>pcrrc8u</code>	* <code>rpcrr</code>	Courier Caps & Small Caps
<code>pcrbc8u</code>	* <code>rpcrb</code>	Courier-Bold Caps & Small Caps
<code>phvr8z</code>	<code>rphvr</code>	Helvetica
<code>phvro8z</code>	<code>rphvro</code>	Helvetica-Oblique
<code>phvb8z</code>	<code>rphvb</code>	Helvetica-Bold
<code>phvbo8z</code>	<code>rphvbo</code>	Helvetica-BoldOblique
<code>phvrc8z</code>	* <code>rphvr</code>	Helvetica Caps & Small Caps
<code>phvbc8z</code>	* <code>rphvb</code>	Helvetica-Bold Caps & Small Caps
<code>phvrn8z</code>	* <code>rphvrn</code>	Helvetica Narrow
<code>phvron8z</code>	* <code>rphvron</code>	Helvetica-Oblique Narrow
<code>phvbn8z</code>	* <code>rphvbrn</code>	Helvetica-Bold Narrow
<code>phvbon8z</code>	* <code>rphvbon</code>	Helvetica-BoldOblique Narrow
<code>phvbnc8z</code>	* <code>rphvbrn</code>	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 počestěny 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 verzi 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 nespolehá 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 `cspfonts.tar.gz`

Balíček `cspfonts.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 `cspfonts.tar.gz` obsaženy metriky pana Wagnera.

Konečně v roce 1999 jsem musel po konzultaci s Karlem Berryem 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 `cspfonts.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`), popsaneho 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. $\mathcal{C}\mathcal{S}$ fonty a fonty zaváděné pomocí `\input ctimes`, `\input cbookman` atd. tuto vlastnost mají.

Překódování textu do vnitřního kódování můžete udělat ručně před spuštěním TEX u (to moc nedoporučuji) nebo je tato konverze implementována do input procesoru samotného TEX u. Jak je to uděláno závisí na použité distribuci TEX u. 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 $\mathcal{C}\mathcal{S}$ plainem musím na výstupu dostat: $\mathcal{Z}=\mathcal{Z}$. Tímto příkladem jsem naznačil dvě věci. Za prvé: některá makra mohou být závislá na zvoleném vnitřním kódování TEX u (zde makro `\char174`, které má vytisknout písmeno \mathcal{Z}). Pokud použiju v systému povel `csplain` bez doplňujících maker, pak předpokládám, že kód 174 znamená písmeno \mathcal{Z} .

Za druhé: pokud vpravo od rovnítko vidím v editoru, kterým zpracovávám vstupní dokument, písmeno \mathcal{Z} , pak se tento znak musí dostat do vnitřních částí TEX u 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 TEX u. Důležité je, že to ve svém editoru *vidím jako \mathcal{Z}* a očekávám na základě této vizuální informace jednotné chování $\mathcal{C}\mathcal{S}$ plainu na všech systémech, kde existují editory, kterými se mohu do zdrojového textu podívat a vidět \mathcal{Z} (ačkoli různé operační systémy mohou toto \mathcal{Z} kódovat různě).

Konverzi do jednotného vnitřního kódování je možné realizovat v různých distribucích TEX u různě. Pročtěte si dokumentaci k použité distribuci. V $\text{e}\text{m}\text{T}\text{E}\text{X}$ u se pro tyto účely používá tzv. TCP tabulka zaváděná při inicializaci formátu. Ve velmi starých web2c distribucích TEX u (léta 1992–1997) můžete najít tzv. Škarvadovu záplatu, která nastavovala kódování podle proměnné prostředí systému. Později se pro překódování používal `encT EX` (rok 1997), který nastavoval `xord/xchr` vektor input procesoru TEX u pomocí přidáných primitivů (viz [6] a [5]). Dnes (po roce 1998) se v distribucích web2c, `te TEX` , `T EX live` 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 TEX u 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 $\mathcal{C}\mathcal{S}$ plain nepřípustné, a proto je nutné v TEX u 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 `i12-cs.tcx` v UNIXových distribucích, přestože tabulka pouze deklaruje překódování „jedna ku jedné“.

Věnujte prosím při implementaci $\mathcal{C}\mathcal{S}$ plainu do jiných TEX ových 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 $\mathcal{C}\mathcal{S}$ plainu, vyzkoušejte test `cstrip` (viz [8]).

4.2 Inicializace formátu

Pokud se vám podařilo \mathcal{C} Splainem zpracovat tento dokument, pak máte formát \mathcal{C} Splain již inicializovaný a můžete směle tuto sekci přeskočit. Pokud používáte nejnovější verzi $\text{T}_{\text{E}}\text{X}$ live, pak stačí napsat na příkazový řádek `csplain` dokument a pokud tak činíte poprvé, formát se automaticky vytvoří. Také můžete inicializovat formát \mathcal{C} Splain 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 $\text{T}_{\text{E}}\text{X}$ u v inicializačním módu, kdy $\text{T}_{\text{E}}\text{X}$ 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} Splain (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 $\text{T}_{\text{E}}\text{X}$ ové distribuci. Níže uvádím několik příkladů.

- **te $\text{T}_{\text{E}}\text{X}$, $\text{T}_{\text{E}}\text{X}$ Live, vstupní kódování ISO-8859-2:**

inicializace: `tex -ini csplain.ini`

obsah skriptu `csplain`:

```
tex -fmt=csplain -default-translate-file=il2-cs $@
```

spuštění \mathcal{C} Splainu: `csplain dokument`

- **te $\text{T}_{\text{E}}\text{X}$, $\text{T}_{\text{E}}\text{X}$ Live, 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} Splainu: `csplain dokument`

- **web2 $\text{T}_{\text{E}}\text{X}$ s enc $\text{T}_{\text{E}}\text{X}$ em, vstupní kódování ISO-8859-2:**

inicializace: `tex -ini csplain.ini`

instalace povelu `csplain`: `ln -s tex csplain`

spuštění \mathcal{C} Splainu: `csplain dokument`

- **web2 $\text{T}_{\text{E}}\text{X}$ s enc $\text{T}_{\text{E}}\text{X}$ em, vstupní kódování podle Kamenických:**

inicializace: `tex -ini '\let\enc=k \input csplain.ini'`

instalace povelu `csplain`: `ln -s tex csplain`

spuštění \mathcal{C} Splainu: `csplain dokument`

- **em $\text{T}_{\text{E}}\text{X}$, 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 povelů 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 L^AT_EXu, zatímco uživatel C_Splainu si vystačí s povely `\chyp` a `\shyp`. V budoucnu přestěhuji tyto styly z archivního souboru `cspplain.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 C_Splainu a pro srovnání tytéž parametry v originálním plainu:

C _S plain	plain
<code>\hsize = 159.2mm</code>	<code>\hsize = 6.5in (165.1mm)</code>
<code>\vsize = 239.2mm</code>	<code>\vsize = 8.9in (226.06mm)</code>
<code>\hoffset = 0mm</code>	<code>\hoffset = 0in</code>
<code>\voffset = 0mm</code>	<code>\voffset = 0in</code>

Výchozí fonty zavedené do formátu C_Splain jsou C_Sfonty. Pro matematiku jsou zavedeny původní Computer Modern fonty. Na rozdíl od plainu formát C_Splain nezávádí do paměti při inicializaci desítky dalších fontů označených jako `\preloaded`, protože další fonty se dají zavést povelom `\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 C_Splainu nepoužité.

Matematická sazba funguje v C_Splainu zcela stejně, jako v originálním plainu. Je to díky tomu, že implicitně zavedené C_Sfonty 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 C_Splainu

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

soubor	Rodina fontů
<code>cavantga.tex</code>	Avantgarde Book
<code>cbookman.tex</code>	Bookman
<code>chelvet.tex</code>	Helvetica
<code>cncent.tex</code>	New Century
<code>cpalatin.tex</code>	Palatino
<code>ctimes.tex</code>	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{C} Sfonty 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{C} Sfontů. 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 (\TeX ová 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 (\TeX ová 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 \TeX u) 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{C} Splainu distribuován soubor `cmt.tex`, takže pokud na začátku dokumentu uvedete

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


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
\chypb
```

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.

`\cmaccentsmessage`

Způsobí výpis informace do logu a na terminál o použití příkazu `\cmaccents`. Pokud vás tato zpráva obtěžuje, nastavte `\let\cmaccentsmessage=\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 `û`. Pokud slovo obsahuje akcentovaný znak zapsaný některou z těchto sekvencí, pak pro ně funguje i automatické dělení slov.

`\csaccentsmessage`

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

`\clqq`

Vytiskne levé české uvozovky („).

`\crqq`

Vytiskne pravé české uvozovky (“).

`\chyp`

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`).

`\ehyp`

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 (ASCII 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 popsáním 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 $\mathcal{C}\mathcal{S}$ plain a $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$

$\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ můžete používat v plainu i $\mathcal{C}\mathcal{S}$ plainu bez nutnosti generovat kvůli tomu formátový soubor. Máte-li dokument v $\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u, napište jednoduše na jeho začátek:

```
\input amstex
```

a dále můžete dokument zpracovat příkazem $\mathcal{C}\mathcal{S}$ plain.

4.9 pdf $\mathcal{T}\mathcal{E}\mathcal{X}$ + $\mathcal{C}\mathcal{S}$ plain = pdf $\mathcal{C}\mathcal{S}$ plain

Pokud vygenerujete formát $\mathcal{C}\mathcal{S}$ plain pdf $\mathcal{T}\mathcal{E}\mathcal{X}$ em, je potřeba jej nazvat jinak, aby byl odlišen od formátu $\mathcal{C}\mathcal{S}$ plain pro originální $\mathcal{T}\mathcal{E}\mathcal{X}$. Proto je v $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u zvoleno jméno `pdfcsplain.fmt`. Dále je potřeba připravit příkaz `pdfcsplain`, který spustí pdf $\mathcal{T}\mathcal{E}\mathcal{X}$ s formátem `pdfcsplain.fmt` a zajistí případné překódování na úrovni input procesoru pdf $\mathcal{T}\mathcal{E}\mathcal{X}$ u stejně, jako to dělá příkaz `csplain`. Pokud uživatel nepoužil ve svých makrech konstrukce měnící sazbu při použití pdf $\mathcal{T}\mathcal{E}\mathcal{X}$ u, 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 $\mathcal{T}\mathcal{E}\mathcal{X}$ u 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 $\mathcal{T}\mathcal{E}\mathcal{X}$ ových distribucích se situace může trochu lišit.

Poznamenejme, že v `teTeXu` a odvozených distribucích většinou stačí napsat `pdfcspain` dokument, a pokud jste `pdfcspain` 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 nejnnutnější 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 $\mathcal{C}\mathcal{S}$ plainu stejný názor jako já.

$\mathcal{C}\mathcal{S}$ plain 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 $\mathcal{C}\mathcal{S}$ plainu 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/ $\mathcal{C}\mathcal{S}$ plainu si vytváří vlastní stále znovu používaná makra, která mu musí fungovat ve všech i budoucích verzích $\mathcal{C}\mathcal{S}$ plainu. Proto se snažím $\mathcal{C}\mathcal{S}$ plain pokud možno neměnit.

Lákavá je například změna definice uvozovek v $\mathcal{C}\mathcal{S}$ plainu jednoduše na

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

aby fungoval automatický kerning s oběma stranami uvozovek. Tuto změnu ale nikdy v $\mathcal{C}\mathcal{S}$ plainu 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 $\mathcal{C}\mathcal{S}$ plainu, a že si ji každý může zařadit do svých maker.

Změnu v $\mathcal{C}\mathcal{S}$ plainu z <Feb. 2000> považuji za asi největší, kterou jsem byl ochoten udělat. K zařazení alternativního kódování Cork (které samozřejmě není a nikdy nebude v $\mathcal{C}\mathcal{S}$ plainu 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 $\mathcal{C}\mathcal{S}$ plain, takže si nemusejí vytvářet své vlastní formáty.

5. Formát $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$

5.1 Různé $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ y

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

V $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u 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 $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u, které jsou starší než rok a půl, generování končí chybou a na terminálu se objeví varování, že máte instalován příliš starý $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 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 \LaTeX em 2_{ϵ} . Pojmem \LaTeX tedy budeme rozumět \LaTeX 2_{ϵ} . I za těchto okolností budeme nuceni rozlišovat mezi třemi „druhy“ \LaTeX ů:

- 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 používáný.

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 používáný.

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 \CSLaTeX u nazván `cslatex.fmt` a nikoli `latex.fmt`, aby bylo možno jej odlišit od běžně používaného babelizovaného \LaTeX u. Z toho důvodu je v balíčku \CSLaTeX u 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 \TeX u. Tento princip je tedy shodný s \CSplainem .

Je patrné, 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 \LaTeX u 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 \TeX ové 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 \TeX u 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 $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, 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 $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u pak bude stylový soubor `czech.sty` nebo `slovak.sty` při použití formátu $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 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 $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u 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 $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u 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 nevedeme na příkazové řádce žádný název formátu.

S uvedenou konfigurací je možné bezproblémové soužití babelizovaného $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u s $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ em v jediné distribuci. $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ pak obvykle voláme skriptem `cslatex`, který provede `tex -fmt cslatex`, zatímco babelizovaný $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ voláme skriptem `latex`, což spustí `tex -fmt latex`.

Pokud nelze ve vaší $\mathcal{T}\mathcal{E}\mathcal{X}$ ové distribuci konfigurovat prohledávací algoritmy podle názvu formátu, pak je možné soužití babelizovaného $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u s $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ em za předpokladu, že budete dodržovat následující zásady:

- Babelizovaný $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ vygenerujte v době, kdy není v distribuci přítomen $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. Tak máte jistotu, že `iniTeX` nenažde soubor `hyphen.cfg` z $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u, ale z balíku Babel.

- $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ pak vygenerujte nad adresářem `cslatex`. Aktuální adresář má totiž ve všech distribucích $\mathcal{T}\mathcal{E}\mathcal{X}$ přednost před ostatními adresáři v `texmf` stromu. Proto si iní $\mathcal{T}\mathcal{E}\mathcal{X}$ přečte soubory `fonttext.cfg` a `hyphen.cfg` z balíčku $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$.
- Po instalaci $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u 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 $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ (o tom pojednám podrobněji v následující sekci). Na druhé straně soubory `czech.sty` a `slovak.sty` z $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u budete potřebovat velmi často, takže tyto soubory nemažte.

5.2 Záhlaví $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ ového dokumentu

Pokud používáte $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, 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 $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ bude pracovat stejně jako vanilla $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$.

Pokud naopak používáte babelizovaný $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$, 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 $\CS\LaTeX$ u není v babelizovaném \LaTeX u řešena synchronizace vstupního kódování dokumentu s vnitřním kódováním \LaTeX u. \LaTeX implicitně pracuje s vnitřním kódováním podle CM fontů (tzv. `OT1`), což pro většinu jazyků není užitečné. Je tedy třeba při použití babelizovaného \LaTeX u 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 $\CS\LaTeX$ u) žádné jiné vnitřní kódování vhodné pro češtinu nebo slovenštinu.

Dále je potřeba v babelizovaném \LaTeX u 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 $\CS\LaTeX$ u

Uvedeme podrobně rozdíly mezi vanilla \LaTeX em a $\CS\LaTeX$ em.

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

$\CS\LaTeX$ 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 $\CS\LaTeX$ u 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`. $\CS\LaTeX$ dále definuje mechanismy přepínání mezi těmito vzory dělení poté, co uživatel přepne vnitřní kódování z `T1` na `IL2` nebo naopak standardními \LaTeX ovými prostředky. Kvůli tomu je v $\CS\LaTeX$ u předefinováno interní makro \LaTeX u `\DeclareFontEncoding`, protože vanilla \LaTeX nepředpokládá, že je po přepnutí kódování nutné přepnout i vzory dělení slov.

V $\CS\LaTeX$ u 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 \CS plainu 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 \TeX u je proto ve skriptu `cslatex` resp. dávice `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 \TeX u způsobem závislým na použité distribuci. Například v \emTeX u 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 $\CSL\TeX$ u pomocí `\usepackage{czech}` způsobí následující změny:

- Nastaví se vnitřní kódování \LaTeX u na IL2, takže se implicitně použijí \mathcal{C} 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 dvojitě 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 \LaTeX u, která jsou změněna po načtení `czech.sty` na české názvy:

Původní název	<code>czech.sty</code>	<code>slovak.sty</code>
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 tieď

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

T1

místo implicitního vnitřního kódování \LaTeX 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 \LaTeX . Pokud je načten z babelizovaného \LaTeX , 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 \LaTeX , 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 \LaTeX

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

$\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u přítomny potřebné `fd` soubory. Následující tabulka ukazuje parametry příkazu `\usepackage` pro rodiny fontů ze standardní skupiny 35 PostScriptových fontů.

Rodina fontů	parametr
Avantgarde Book	<code>avant</code>
Bookman	<code>bookman</code>
Helvetica	<code>helvet</code>
New Century	<code>newcent</code>
Palatino	<code>palatino</code>
Times Roman	<code>times</code>

5.6 Použití fontů kódovaných podle Corku (T1 kódování) v $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u

Máte-li vzory dělení v $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u 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 $\mathcal{C}\mathcal{S}$ fontů fonty kódované podle Corku. Vnitřní kódování T1 inicializujete zápisem:

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

Protože ale input procesor $\mathcal{T}\mathcal{E}\mathcal{X}$ u při použití příkazu `cslatex` konvertuje vstupní kódování dokumentu na ISO 8859-2, je v tomto případě nutné navázat na to další konverzi pomocí balíčku `inputenc` takto:

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

Zde je potřeba vždy psát `[latin2]`, ať je vstupní kódování dokumentu jakékoli, protože vstupní procesor $\mathcal{T}\mathcal{E}\mathcal{X}$ u 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 $\mathcal{T}\mathcal{E}\mathcal{X}$ u a kódované podle Corku.

5.7 $\text{pdf}\mathcal{T}\mathcal{E}\mathcal{X} + \mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X} = \text{pdf}\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$

V případě spojení $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u s $\text{pdf}\mathcal{T}\mathcal{E}\mathcal{X}$ em platí vše naprosto stejně, jako bylo řečeno v sekci „ $\text{pdf}\mathcal{T}\mathcal{E}\mathcal{X} + \mathcal{C}\mathcal{S}\text{plain} = \text{pdf}\mathcal{C}\mathcal{S}\text{plain}$ “. Nahraďte v této sekci slovo $\mathcal{C}\mathcal{S}\text{plain}$ slovem $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ a slovo $\text{pdf}\mathcal{C}\mathcal{S}\text{plain}$ slovem $\text{pdf}\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ a přečtěte si tuto sekci ještě jednou.

5.8 Historie a budoucnost $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u

$\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ vytvořil zhruba v roce 1992 Jiří Zlatuška. Od něj pochází myšlenka načtení vzorů dělení stejného jazyka v různých kódováních a předefinování vnitřního $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ ového makra `\DeclareFontEncoding`. Na své implementaci $\mathcal{T}\mathcal{E}\mathcal{X}$ u tehdy provozoval mimo jiné fonty kódované v KOI-8, takže přepínání vnitřního kódování

L^AT_EXu 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 zaneprázdněnosti v jiné oblasti tímto problémem později zřejmě nezabýval.

Po schůzce tvůrců $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u v roce 1993 převzal starost o $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ podle dohody Zdeněk Wagner, který vytvořil definice kódování IL2. Vytvořil také pro $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ definiční soubory `fd` jednak pro $\mathcal{C}\mathcal{S}$ fonty a jednak pro PostScriptové fonty z balíčku `cspsfonts.tar.gz`. Od něj také pochází implementace $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u pro L^AT_EX 2.09. V em_TE_Xové distribuci $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u je stále tato implementace obsažena (pod označením latex209).

Soubor `czech.sty` má asi podstatně delší historii než $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. 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 $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u. Slovenskou část včetně vzorů dělení slov vytvořila Janka Chlebíková. Soubor `slovak.sty` je přesnou kopií souboru `czech.sty` s výjimkou slovensky specifických částí.

V duchu licence GPL převzal zhruba v roce 1997 údržbu $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u 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 $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ em od této doby nejde bez chyb formátovat český překlad úvodu do L^AT_EXu, který je pod názvem balíčku `csuvodlat.tar.gz` součástí dokumentace $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u. Je to názorná ukázka toho, co může způsobit změna kódu, která nerespektuje zpětnou kompatibilitu. Pan Šnajdr rovněž napsal `html` dokumentaci k $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u, která popisuje instalaci $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u 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 L^AT_EX a tím pádem $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ jevil od začátku malý zájem, protože celý projekt je závislý na L^AT_EXu samotném. Nemám tedy jistotu, jaké změny v něm současný L^AT_EX-team udělá a jaké budou existovat do budoucna potíže se zpětnou kompatibilitou. To je zásadní odlišnost od Knuthova plainu a T_EXu samotného. Raději jsem se tedy do L^AT_EXových věcí nemíchal.

V roce 1999 jsem nicméně přidal pár řádek maker do souboru `czhyphen.tex` tak, aby byl použitelný v babelizovaném L^AT_EXu. Do té doby totiž tato větev L^AT_EXu používala Lhotkovy vzory dělení, zatímco v $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u jsme už dávno měli daleko kvalitnější Ševečkovy vzory dělení. Tyto novější vzory dělení jsou totiž napsány za použití T_EXových 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 $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u zůstává původní soubor `czhyphen.tex`. Sjednocení názvů těchto souborů by stejně nevyřešilo nejednotnost vývoje $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u a babelizovaného L^AT_EXu.

Na výborové schůzi v roce 1999 jsem dostal za úkol prověřit možnost spojení babelizovaného L^AT_EXu s $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ em. Neustálé dotazy začínajících uživatelů, kteří

si pletou tyto dva \LaTeX y, nás utvrzují v tom, že by se pro sloučení mělo něco udělat.

Analyzoval jsem proto makra Babelu a udělal návrh na možné zapracování funkcionality \CSLaTeX u do Babelu. Domnívám se, že \CSLaTeX klidně může přestat existovat, ale babelizovaný \LaTeX musí bezpodmínečně převzít všechny vlastnosti \CSLaTeX u tak, aby dokumenty dříve zpracovávané \CSLaTeX em byly naprosto stejně a bez jediné úpravy zpracované novým babelizovaným \LaTeX em. 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 jediné zásah do jádra \LaTeX u samotného, aby dokázal při změně kódování fontů přepnout automaticky i vzory dělení. Zlatuška kvůli tomu předefinoval makro jádra \LaTeX u `\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 \LaTeX ové jádro neřeší, považuji totiž za chybu \LaTeX u. V roce 1999 jsem tedy požádal \LaTeX -team, aby zapracoval změnu v duchu Zlatuškovy návrhu do \LaTeX ového jádra. Můj návrh nebyl \LaTeX -teamem akceptován. Za těchto okolností nejsem schopen zapracovat funkcionalitu \CSLaTeX u do Babelu, protože to prostě nejde. Uživatelé \LaTeX u se budou muset nadále potýkat s tím, že jejich oblíbený formát trpí určitou schizofrenií.

Společně se sloučením \CSLaTeX u 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 reliktvů z dob minulých, plno zcela nepoužívaných větví ve složitém větvení pomocí `\if` a stávají se totálně nepřehlednými. Protože ale ke sloučení \CSLaTeX u 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 \CSLaTeX em 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 \CSLaTeX em. Jak jsem už vysvětlil, jsem ochoten převzít iniciativu jen tehdy, když bude \LaTeX ové jádro umět přepínat mezi různě kódovanými vzory dělení stejného jazyka. Přitom členové \LaTeX -teamu jsou toho názoru, že to možná bude zapracováno až do \LaTeX u 3.

6. Reference

- [1] Donald Knuth. *The \TeX book*. 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. *\TeX book naruby*. Konvoj, Brno 2001, ISBN 80-7302-007-6.

- [4] Petr Olšák. *První setkání s T_EXem*. 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 $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}\mathcal{U}$, 3/1997, strany 109–118.
- [6] Petr Olšák. *Rozšíření T_EXu encT_EX*. 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 $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}\mathcal{U}$. Je volně šířen na `ftp://math.feld.cvut.cz/pub/cstex/base/cstrip.tar.gz`.
- [9] Petr Olšák. *Makro OFS*. T_EXové makro OFS pro práci s rozsáhlými kolekcemi fontů je volně šířeno na `ftp://math.feld.cvut.cz/pub/olsak/ofs/`.