Novinky v OFS

Petr Olšák

Elektrotechnická fakulta ČVUT, Praha
Email: `petr@olsak.net`

Abstrakt: OFS (Olšákův fontový systém) byl již na SLT prezentován. Během roku 2004 byl ale OFS pro plain významě přepracován a byly do něj přidány nové vlastnosti, které umožňují vytvářet on-line katalogy fontů nebo důkladné testy fontových rodin včetně kombinace s matematickou sazbou. Rovněž byly rozšířeny možnosti práce s makry závislými na kódování a implementována podpora TX fontů. Na přednášce budou v krátkosti tyto nové vlastnosti OFS předvedeny posluchačům.

1 Úvodem

Makro OFS jsem poprvé zveřejnil v roce 2001 a na SLT v Seči jsem toto makro prezentoval [3]. Pak se s OFS delší dobu nic nedělo. Teprve letos jsem se pokusil vytvořit pro OFS jednak interaktivní makro na testy celých fontových rodin a dále jsem začal pracovat na OkTEXu [4, 5], kde jsem se rozhodl OFS pro plain využít.

Tyto dva cíle mě donutily udělat v OFS několik docela zásadních změn. Změny se týkají výhradně verze OFS pro plain. Na LATEXovou verzi jsem vůbec nesáhl. Neznamená to ale, že by LATEXový uživatel neměl z této práce žádný užitek. Může si pomocí příkazu `csplain ofstest` interaktivně vytvářet vzorníky fontů. Myslím si, že v tomto případě mu může být jedno, jakým formátem je ten vzorník zpracován.

2 Interaktivní makro `ofstest.tex`

Po spuštění `csplain ofstest` nebo `tex ofstest` si program vyžádá vložení názvu definičního souboru. Například `a35` nebo `slido`. Pokud rovnou na příkazový řádek za slovo `ofstest` a za mezeru zapíšeme definiční soubor(y) do hranaté závorky, pak program už tento údaj nežádá znovu. Dále můžeme vybrat rodinu fontů, kterou chceme testovat. Potom lze příkazem `\help` vypsat další možnosti, co můžeme dělat. Komunikace může vypadat například takto (údaje, které vložil uživatel, jsou vyznačeny rámečkem):

```
$ csplain ofstest [ffonts]
This is TeX, Version 3.14159 (Web2C 7.3.7x)
encTeX v. Feb. 2003, the reencoding enabled.
(/usr/TeX/texmf/tex/ofs/ofstest.tex The format: csplain <Feb. 2000>.
The cs-fonts are preloaded and A4 size implicitly defined.
(/usr/TeX/texmf/tex/ofs/ofs.tex
OFS (Olsak's Font System) based on plain initialized. <May 2004>
```

Petr Olšák (editor): SLT 2004 – sborník semináře o Linuxu a TEXu, str. 1–10, 2004.

```
(/usr/TeX/texmf/tex/ofs/ofsdef.tex))
Czech hyphenation used (\language=5). \frenchspacing is set on.
(/usr/TeX/texmf/tex/ofs/ofs-8z.tex) (/usr/TeX/texmf/tex/ofs/ofs-8c.tex))
(/usr/TeX/texmf/tex/ofs/ffonts.tex) This is ofstest macro, version <May. 2004>

*** Type family name without brackets (or ? or *): Charter

*** What to do with family Charter ?
    (type command or \help): \help
commands:
 \list ..... List all variants of the family Charter
 \table(s) . Tables of all variants of the family Charter
 \abet ..... One line alphabet/digits sample for each variant
 \chars .... Print list of available characters including TeX sequences
 \text ..... One paragraph in all variants of the family Charter
 \mixed .... Paragraph with fonts combined from \rm, \bf, \it and \bi
 \math ..... Mathematics text combined by fonts from Charter
 \all ...... The same as \list \table \abet \chars \text \mixed \math
 \setsize .. Set size of fonts (current size is "at10pt")
 \cfam ..... Change current family
 FamName ... The same as \cfam FamName
 \rem ...... Remove current family or family given in parameter from \famlist
 ----------------------------
 \famlist .. Show list of all declared families (the same as \showfonts)
 \decl ..... Input next declaration file
 \remdecl .. Remove all families of given declaration file from famlist
 \help ..... This text
 \morehelp . Show more help information
 \fontusage  The help screen of the OFS
 \end ...... End of this session

*** What to do with family Charter ?
    (type command or \help): \abet \chars \mixed \math \end

 [Charter/at10pt]: \rm abet (/usr/TeX/texmf/tex/ofs/ffonts.tex) \bf abet
\it abet \bi abet
 [Charter/]: \chars (/usr/TeX/texmf/tex/ofs/ofs-8z.tex)
 [Charter/at10pt]: mixed text (/usr/TeX/texmf/tex/ofs/ofs-ps.tex)
 [Charter/at10pt] (\fomenc: PS) [10.0pt/7.0pt/5.0pt]: math text
[1]
Output written on ofstest.dvi (1 page, 7492 bytes).
Transcript written on ofstest.log.
```

V tomto příkladě jsme vybrali rodinu Charter (shodou okolností je touto rodinou fontů dělán sborník SLT) a vypsali jsme si pomocí `\help` možnosti, co můžeme podniknout dále. Z nabídky jsme pak vybrali pouze tisk zkrácené abecedy (`\abet`), výpis dostupných znaků v této rodině včetně jejich TEXových sekvencí (`\chars`), tisk vzorku, kde je kombinován základní řez s tučným a s kurzívou (`\mixed`), a konečně tisk vzorku s matematickou sazbou (`\math`). Výsledek testu je vidět na obrázku 1.

Obrázek je v tomto sborníku mírně zmenšený, takže údaje po stranách (že se jedná o 10bodové písmo) nejsou správné. Samozřejmě, originální výstup z makra `ofstest.tex` je ve správné velikosti a údaje o rozměrech souhlasí.

OFS test 15. 5. 2004 1

[Charter/10pt] (enc: 8z), declared in ffonts.tex:

() \rm ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz

(Bold) \bf **ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz**

(Italic) \it *ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz*

(BoldItalic) \bi ***ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz***

Charter (8z)
\\` *: *̀ \\' *: *́ \v *: *̌ \u *: *̆ \= *: *̄ \r *: *̊ \^ *: *̂ \\. *: *̇ \H *: *̋ \~ *: *̃ \\" *: *̈
\c *: *̧ \k *: *̨ \m *: *̦ \b *: *̱ \d *: *̣ \dotlessi: ı \dotlessj: [?] \\" i: ï \\" \i: ï
\SS: SS \AE: Æ \OE: Œ \O: Ø \ss: ß \ae: æ \oe: œ \o: ø \ellipsis: … \promile: ‰
\varhyphen: - \flqq: « \frqq: » \clqq: „ \crqq: “ \clq: ‚ \crq: ‘ \elqq: “ \erqq: ”
\elq: ‘ \erq: ’ \exclamdown: ¡ \questiondown: ¿ \dag: † \ddag: ‡ \section: § \paragraph: ¶
\ellipsis: … \textbullet: • \sterling: £ \currency: ¤ \Lslash: Ł \lslash: ł \Eth: [?]
\eth: [?] \texttimes: [?] \textdiv: [?] \\` A: À \\' A: Á \\" A: Ä \\` a: à \\' a: á \\" a: ä
\v C: Č \v c: č \v D: Ď \v d: ď \\' E: É \v E: Ě \\' e: é \v e: ě \\' I: Í \\' i: í \\' \i: í
\v L: Ľ \\' L: Ĺ \v l: ľ \\' l: ĺ \v N: Ň \v n: ň \\' O: Ó \^ O: Ô \\" O: Ö \\' o: ó \^ o: ô
\\" o: ö \\' R: Ŕ \v R: Ř \\' r: ŕ \v r: ř \v S: Š \v s: š \v T: Ť \v t: ť \r U: Ů \\' U: Ú
\\" U: Ü \r u: ů \\' u: ú \\" u: ü \\' Y: Ý \\' y: ý \v Z: Ž \v z: ž \k A: Ą \^ A: Â \u A: Ă
\k a: ą \^ a: â \u a: ă \\' C: Ć \c C: Ç \\' c: ć \c c: ç \k E: Ę \\" E: Ë \k e: ÿ \\" e: ë
\^ I: Î \^ i: î \^ \i: î \\' N: Ń \\' n: ń \H O: Ő \H o: ő \\' S: Ś \c S: Ş \\' s: ś \c s: ş
\c T: Ţ \c t: ţ \H U: Ű \H u: ű \\' Z: Ź \\. Z: Ż \\' z: ź \\. z: ż

Modifications { 8z:charter }

Charter 10pt/12.0pt

Příklad. Nyní zjistíme, zda je možné kombinovat základní řez s tučným (resp. polotučným) řezem a s kurzívou v rodině písma *Charter*. Pro vyznačování je vhodné použít *kurzívu* a už méně **tučnou variantu** a prakticky vůbec se nehodí použít ***tučnou a skloněnou variantu***. Zcela nevhodné je vyznačovat podtržením nebo prostrkáním. To lidé s dobrým vychováním nedělají. *Vyznačení má být takové, aby při čtení bylo vyznačené místo zřetelně odlišné, ale při pohledu z dálky zůstal text odstavce stejnoměrně šedý.* Tomu nejlépe vyhovuje kurzíva, ale ne ve všech rodinách písma je zdařilá kuzíva k dispozici.

Charter (PX) [10.0pt/7.0pt/5.0pt]

Poznámka. Funkci Gamma v bodě x značíme $\Gamma(x)$ a počítáme ji podle vzorce:

$$\Gamma(x) = \int_0^\infty e^{-t} t^{x-1} \mathrm{d}t \quad (x > 0).$$

Speciálně pro $x = n \in \mathbb{N}$ je $\Gamma(n) = (n-1)\,!$ a pro $\alpha \in (0,1)$ je

$$\Gamma(\alpha)\,\Gamma(1-\alpha) = \frac{\pi}{\sin \pi\alpha}.$$

Definice. Nechť $\boldsymbol{A}$ je čtvercová matice s n sloupci a řádky a s prvky $a_{i,j}$. Pak číslo

$$\det \boldsymbol{A} = |\boldsymbol{A}| = \sum_{\Pi=(j_1,j_2,\dots,j_n)} \operatorname{sgn}\Pi \cdot a_{1,j_1}\, a_{2,j_2} \cdots a_{n,j_n}$$

nazýváme *determinantem* matice $\boldsymbol{A}$.

Obrázek 1. Výstup z `ofstest.tex`. Charter, příkazy: `\abet`, `\chars`, `\mixed`, `\math`

Funkci všech příkazů, které jsou v interaktivním prostředí k dispozici, nebudu podrobně rozebírat. Stručný význam těchto příkazů je vytištěn na terminál pomocí příkazu `\help` a je zahrnut v předchozí ukázce.

Před tiskem můžeme měnit interaktivně některé parametry. Například pomocí `\def\fotenc{8t}` dáme najevo, že chceme testovat fonty v kódování `8t`. Pomocí příkazu `\setsize` můžeme zadat jiný základní rozměr fontu než `10pt` a příkazem `\baselineskip=...` můžeme měnit velikost řádkování. Konečně příkazem `\def\textlang{en}` dáváme najevo, že chceme vzorky v anglickém textu a nikoli českém.

Výchozí hodnoty makra `ofstest.tex` jsou závislé na tom, zda je použit formát `csplain` nebo formát `plain`. V prvním případě je výchozím jazykem

	OFS test 15. 5. 2004 1
	[AntykwaTorunska/10pt] (enc: 8z), declared in pantyk.tex:
(Regular) \rm	ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz
(Bold) \bf	**ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz**
(Italic) \it	*ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz*
(BoldItalic) \bi	***ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz***
(Light) \lr	ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz
(LightItalic) \li	*ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz*
(Medium) \mr	ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz
(MediumItalic) \mi	*ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz*
	[AntykwaTorunskaCaps/10pt] (enc: 8z), declared in pantyk.tex:
(Regular) \rm	ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ
(Bold) \bf	**ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ**
(Italic) \it	*ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ*
(BoldItalic) \bi	***ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ***
(Light) \lr	ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ
(LightItalic) \li	*ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ*
(Medium) \mr	ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ
(MediumItalic) \mi	*ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ*
	[AntykwaTorunskaCond/10pt] (enc: 8z), declared in pantyk.tex:
(Regular) \rm	ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz
(Bold) \bf	**ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz**
(Italic) \it	*ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz*
(BoldItalic) \bi	***ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz***
(Light) \lr	ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz
(LightItalic) \li	*ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz*
(Medium) \mr	ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz
(MediumItalic) \mi	*ABCDEOPQRSVWXYZ 0123456789 &%?!-Žžščěáéťů, acbdefghijklmnopqrstuvwxyz*
	[AntykwaTorunskaCondCaps/10pt] (enc: 8z), declared in pantyk.tex:
(Regular) \rm	ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ
(Bold) \bf	**ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQŔSTUVWXYZ**
(Italic) \it	*ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ*
(BoldItalic) \bi	***ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQŔSTUVWXYZ***
(Light) \lr	ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ
(LightItalic) \li	*ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ*
(Medium) \mr	ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ
(MediumItalic) \mi	*ABCDEOPQRSVWXYZ 0123456789 &%?!-ŽžŠČĚÁÉŤŮ, ACBDEFGHIJKLMNOPQRSTUVWXYZ*

Obrázek 2. Výstup z `ofstest.tex`. Katalog volně dostupných rodin AntykwaTorunska

ukázek čeština a výchozí kódování je `8z` (alias IL2). Při formátu `plain` je výchozím jazykem angličtina a kódování je implicitně nastaveno na `8t` (podle Corku).

Makro `ofstest.tex` nemusí testovat jen jedinou rodinu. Napíšeme-li místo jména rodiny hvězdičku, provede se pak zvolený příkaz na všechny rodiny, které jsou načteny v deklaračních souborech. Seznam těchto rodin je dostupný pomocí známého příkazu OFS `\showfonts` nebo příkazem `\famlist`. Makro dále umožní za chodu načítat další deklarační soubory (příkaz `\decl`) a ze seznamu rodin vymazávat některé rodiny. Až nám po takové činnosti zůstanou v seznamu `\listfam` jen rodiny, které chceme podrobit testu, pak teprve použijeme příkazy `\abet`, `\text`, `\mixed` a další.

Na obrázku 2 je výstup, který se dá využít jako vzorníček fontů. Zvolil jsem deklarační soubor `pantyk` (Polish Antykwa) a příkazem `\remdecl defaults` jsem odstranil z `\listfam` implicitní rodiny OFS. Zůstaly tam jen rodiny Antykwy Toruńské, z nichž jsem si udělal vzorníček příkazem `\abet`. Takové vzorníčky stovek rodin, které mám na počítači k dispozici, jsem si vytiskl a vybírám z nich písma podle aktuální potřeby (vzorníčky nechám kolovati).

Všimněme si, že OFS pro plainTEX si udržuje v paměti informace o původních názvech jednotlivých variant fontů. V uvedené ukázce vidíme varianty `Light`, `LightItalic`, `Medium` a `MediumItalic`. Ve vzorníčku máme také uvedeny příkazy na přepnutí do zvolené varianty.

Závěrem bych chtěl poznamenat, že interaktivní makro `ofstest.tex` se dá použít i neinteraktivním způsobem. V plainTEXovém dokumentu můžeme napsat například:

```
\input ofstest [a117]
\cfam Bodoni        \tables \text \mixed
\cfam LetterGothic  \tables \text \mixed
⟨...atd.⟩
```

Makro samo pozná, že je zavoláno v neinteraktivním režimu a potlačí nutkání klást jakékoli dotazy.

3 Výjimky z kódování pro každou rodinu fontů

Ukazuje se, že každá rodina fontů je vybavena mírně odlišnou sadou znaků. Tu chybí netečkované j, tam zas je navíc k dispozici Euro atd. Není rozumné pro každou takovou výjimku vytvářet kódování s novým názvem. OFS raději pracuje s pořád stejným kódováním, ale je nově schopno registrovat všechny výjimky z kódování jednotlivých rodin a udržovat si podle toho představu, které znaky jsou momentálně k dispozici a které ne. Myslím si, že touto vlastností například NFSS z LATEXu nedisponuje.

Za základ pro kódování `8z` jsem vzal encoding vektor `xl2.enc`, který vytvořil pan Wagner. Vzhledem k tomuto kódování pak každá rodina fontů může mít nějaké výjimky. Třeba rodině Charter chybí znaky `\dotlessj`, `\Eth`, `\eth`, `\textimes` a `\texdiv`. Je to názorně vidět na výpisu při testu příkazem `\chars` (obrázek 1). Naopak třeba rodina fontů Lido má navíc (proti základnímu kódování `8z`) znaky `\degree`, `\euro`, `\trademark` atd.

V deklaračních souborech můžeme výjimky seskupit do skupin příkazem `\modifydef`, jednotlivé výjimky pak tam daklarujeme pomocí `\characterdef`, `\characterdel`, `\accentdef` a `\accentdel` a konečně jednotlivým rodinám můžeme přidělit určité skupiny výjimek příkazem `\modifyenc`. Programátor maker se může ptát na existenci znaku ve fontu příkazem `\knowchar`. Podrobněji je o těchto příkazech pojednáno v dokumentaci [2].

Například rodina Charter má pro kódování `8z` přidělenu skupinu výjimek s názvem `8z:charter`. O této skutečnosti nás rovněž informuje výpis `\chars` na obrázku 1 v odstavci `Modifications`. Kdyby tato skupina výjimek obsahovala znaky navíc, byly by v tomto odstavci vykresleny. Výpis `\chars` ovšem zatajuje, že nejsou dosažitelné některé akcentované znaky. Například v rodině Charter chybí znak ű. Proto je pro něj deklarována výjimka pomocí `\accentdel`. Jakmile je výjimka registrována, OFS automaticky začne realizovat sekvenci „`\H u`“ pomocí implicitního akcentu, takže i tento znak nám bude fungovat (ačkoli bude v TEXu složen ze dvou znaků pomocí primitivu `\accent`).

4 Automatické čtení kódovacích souborů

Aby mohl OFS udržovat přehled o kontrolních sekvencích pro jednotlivé znaky a tím také o výjimkách z kódování, musí mít načteny odpovídající soubory `ofs-`⟨*kódování*⟩`.tex`. Už ve verzi z roku 2001 tyto soubory existovaly, ale uživatel si je načítal podle vlastního uvážení sám příkazem `\input`. Protože se v OkTEXu bude pracovat s množstvím různých kódování a makra mezi těmito kódováními budou sofistikovaně a mnohdy automaticky přepínat, rozhodl jsem se OFS vybavit možností automatického načítání souborů `ofs-`⟨*kódování*⟩`.tex` v okamžiku, kdy to je nutné.

Abych neiritoval přílišnou inteligencí makra OFS ty uživatele, kteří mají rádi konzervativní přístup a raději si soubory načtou sami, je implicitně automatické načítání kódovacích souborů vypnuto. Uživatel si může tuto funkci zapnout pomocí příkazu `\loadingenc=1`. Teprve po takovém příkazu máme zaručeno, že OFS udržuje přehled o TEXových sekvencích vztahujících se ke kódování použitých fontů.

Automatické načtení souboru `ofs-`⟨*kódování*⟩`.tex` může proběhnout ve velmi nepříznivé době: například v horizontálním módu, uvnitř skupiny nebo dokonce ve stavu, kdy má uživatel nastaveny `\catcode` speciálním způsobem. Aby přesto proběhlo načtení souboru bezkonfliktně, rozhodl jsem se toto načtení řešit následujícími kroky:

- Před načtením kódovacího souboru založí OFS novou skupinu.
- Pro jistotu v této skupině nastaví všem znakům standardní kategorie a znaku `@` nastaví kategorii 11.
- Nastaví `\endlinechar=-1`, takže další čtení ze souboru ignoruje prázdné řádky (neprovede se tedy žádné nadbytečné `\par`) a ignoruje mezery, které vznikají na koncích řádků (nedochází k zavlečení mezer v horizontálním módu).
- Provede `\globaldefs=1`, aby všechny následující definice z kódovacího souboru zůstaly zachovány i po opuštění skupiny.
- Načte kódovací soubor.
- Ukončí skupinu, takže `\globaldefs`, `\endlinechar` a kategorie znaků mají zase stejný význam, jaký měly před načtením souboru.

5 Novinky v deklaračních souborech

Deklarační soubory vytvořené pro staré OFS z roku 2001 jsou a budou stále použitelné, tj. jazyk těchto souborů je zpětně kompatibilní. V nové verzi OFS jsem ale přidal další vlastnosti a je tedy možno použít nové příkazy.

Mezi již zmíněné novinky patří možnost přidělit každé rodině skupinu výjimek z kódování. To ale zdaleka není všechno, co lze nově využít ve verzi OFS 2004.

Každá rodina fontů může být registrována příkazem `\registerenc` k použití jen pro určitá kódování. Máme tak zaručeno, že například po

```
\def\fotenc{aaa} \setfonts [Times/]
```

se nebude OFS marně snažit najít neexistující metriku `ptmraaa.tfm`, ale vynadá nám, že kódování `aaa` není pro rodinu Times registrováno. Tato vlastnost je bohatě využita v OkTEXu, viz [5].

Každý deklarační i kódovací soubor může začínat příkazem

```
\protectreading ⟨jméno souboru⟩⟨mezera⟩% This is part of OFS package
```

Tím záměrně zkomplikujeme čtení tohoto souboru v balíčcích, kde není příkaz `\protectreading` definován. V nové verzi OFS je tento příkaz definován tak, že si poznamená ⟨*jméno souboru*⟩ do paměti a při příštím čtení stejného souboru se chová jako `\endinput`, tj. zaručí, že níže zapsané deklarace budou přečteny jen jednou. Podobnou vlastnost má LATEXový příkaz `\ProvidesPackage`.

Nově je zavedena též možnost předeklarovat již deklarované rodiny fontů prostým zopakováním příkazu `\ofsdeclarefamily [`⟨*Rodina*⟩`]`, ovšem s jinými parametry. To dává možnost uživateli přizpůsobit i implicitní rodiny fontů vlastním potřebám.

6 Dvojí čtení stejného souboru

Kam psát výjimky z kódování jednotlivých rodin fontů? Slušelo by se, aby byly tyto výjimky v deklaračních souborech společně s deklarovanými rodinami.

V rámci OkTEXu plánuji načíst pokud možno všechny dostupné deklarační soubory rodin fontů už při generování formátu. Pak by ale byly výjimky přečteny už při generování formátu. Ovšem to není dobré, pokud chceme šetřit paměť TEXu. Rozhodl jsem se tedy použít následující trik:

Výjimky z kódování lze zapisovat do deklaračních souborů za `\endinput`, takže při prvním čtení těchto souborů budou ignorovány. Ovšem v době, kdy bude rodina aktivována příkazem `\setfonts`, může OFS otevřít deklarační soubor ještě jednou, tentokrát přeskočí obvyklé deklarace i `\endinput` a přečte seznam výjimek z kódování. Pro tyto účely jsou nově vytvořeny příkazy `\modifyread` a `\modifytext`. Jak přesně pracují, je popsáno v dokumentaci [2] a příklad jejich použití najde čtenář v novém deklaračním souboru `slido.tex`.

7 Makro `\ofshexbox`

V plainTEXu je například znak § řešen pomocí makra `\mathhexbox`. Při vší úctě ke Knuthovi si dovolím poznamenat, že to je docela nekoncepční, protože textový znak je deklarován tak, že jeho funkčnost závisí na momentálním nastavení matematických fontů. Rozhodl jsem se tedy vytvořit pro takové účely nové makro `\ofshexbox`, které závisí jen na čtevřici deklarovaných textových fontů, jež tvoří rodinu. OFS si navíc pohlídá velikost těchto fontů, tj. zaručí, že vždy budou použity ve správné velikosti. Pomocí příkazu

```
\ofshexboxdef ⟨rodina⟩{⟨metrika-rm⟩}{⟨metrika-bf⟩}{⟨metrika-it⟩}{⟨metrika-bi⟩}
```

lze deklarovat ⟨*rodinu*⟩ jako čtveřici metrik, která bude pak při použití makra `\ofshexbox`⟨*rodina*⟩⟨*hexa-kód*⟩ použita. Která konkrétně metrika bude použita, závisí na aktuální variantě fontů. Při `\bf` bude použita ⟨*metrika-bf*⟩, atd. Je-li aktuální jiná varianta než `\bf`, `\it`, `\bi`, bude použita ⟨*metrika-rm*⟩.

Následující ukázka z dokumentace ilustruje, jak se můžeme jednou pro vždy vyrovant s tím, že některé rodiny znak Euro obsahují a některé ne:

```
\ofshexboxdef {TS1}{tcrm1000}{tcbx1000}{tcti1000}{tcbi1000}
\characterdef \euro * {\ofshexbox{TS1}BF}
```

Použité metriky `tc*1000.tfm` obsahují znak Euro na pozici BF. Tyto metriky z balíčku EC fontů jsou kódované podle tzv. Text Companion Encoding, v LaTeXu označované jako TS1. V ukázce jsme deklarovali znak `\euro` *pro všechny rodiny fontů, ve kterých znak není deklarován* pomocí makra `\ofshexbox`. Znak se tedy v rodinách, kde Euro není přítomno, vyloví z metrik `tc*1000.tfm`, a přitom bude záviset na aktuální variantě (tučná/kurzíva/tučná kurzíva) a navíc bude vždy použit v aktuální velikosti fontů. Pokud ale znak Euro je v rodině fontů obsažen, pak se přednostně použije tento znak.

Aby to celé fungovalo, je samozřejmě potřeba deklarovat pro každou rodinu odpovídající skupinu výjimek ze základního kódování. Pak bude mít uživatel záruku, že pokud napíše `\euro`, vždycky se mu vytiskne znak €. Přitom se nemusí starat, zda znak v aktuálně použitém fontu je nebo není.

8 Nové deklarace v OFS

Do nové verze OFS jsem přidal deklarace rodin `CM*` v kódování `8t` (podle Corku). Tyto rodiny odkazují na EC fonty a respektují všechny stupně zvětšení (různé metriky pro různé velikosti).

Dále jsem definoval nové kódování `8c`, které odpovídá tzv. „Text Companion encoding“ v LaTeXu známé jako TS1. Všem implicitním rodinám fontů `CMRoman`, `CMSans`, `CMTypewriter`, `Times`, `Helvetica`, `Courier` jsem přidělil rozšiřující metriku v kódování `8c`. Tím je například vyřešen znak `\euro` v rodinách `CM*`. Rodiny `Times`, `Helvetica` a `Courier` mají sice ve standardních distribucích TeXu připraveny metriky `8c`, ale tyto metriky zdaleka neobsahují všechny znaky podle definice tohoto kódování (například `\euro` chybí).Tyto rodiny mají tedy v OFS přiděleny výjimky z kódování `8c`.

Vzhledem k tomu, že metriky `tc*.tfm` patří rodině `CMRoman` v kódování `8c`, je možné příklad z předchozí sekce s příkazem `\ofshexbox` přepsat takto:

```
\ofshexboxdef {TS1}{cmr8c}{cmbx8c}{cmti8c}{cmbxti8c}
\characterdef \euro * {\ofshexbox{TS1}BF}
```

Čtenář, který zná dokumentaci k OFS, si jako cvičení může zkusit najít rozdíl mezi deklarací `\ofshexboxdef` z předchozí sekce a touto ukázkou. (Odpověď: při práci s různými velikostmi se podle předchozí sekce použije znak euro vždy z metriky `tc*1000.tfm` příslušně zvětšené pomocí `at`. Na druhé straně tato nová deklarace vybere podle požadované velikosti odpovídající metriku, třeba `tcrm1095.tfm`, `tcrm1200.tfm` atd. vždy navíc přesně zvětšenou pomocí `at`.)

9 Deklarační soubory pro matematické fonty

OFS z roku 2001 deklarovalo jen matematické kódování PS (PostSriptové znaky z fontu Symbol) a `CM` (Znaky z Computer Modern fontů). Později přibyl doplňkový soubor, kde byla deklarace pro fonty MathTimes.

V roce 2004 jsem nejprve vylepšil kódování PS tak, že znaky $\sum$, $\int$ a $\prod$ mají v display módu zvětšené varianty a dále jsem deklaroval další matematická kódování:

- Po načtení souboru `amsfn.tex` může uživatel použít `\def\fomenc{AMS}` a využít desítky speciálních matematických symbolů známých především z $\mathcal{AMS}$TEXu.
- Po načtení souboru `txfn.tex` může uživatel použít jednu ze dvou možností: `\def\fomenc{TX}` — pro matematiku se použijí TX fonty, tj. volně přístupná úplná sada matematických znaků vizuálně kompatibilních s rodinou Times a tedy i s mnoha dalšími běžnými písmy dynamické antikvy. Druhou možností je `\def\fomenc{PX}` — podobně jako při kódování PS se kombinuje kurzíva aktuální rodiny fontů s matematickými znaky z TX fontů.
- Po načtení souboru `mtfn.tex` může uživatel použít `\def\fomenc{MT}`. Pak se použijí metriky komerční sady matematických fontů MathTimes.

V tuto chvíli už mohu prozradit, že jsem na začátku tohoto článku zatajil, že obrázek 1 nekoresponduje zcela přesně s výpisem, který obrázku předchází. Test `\math` byl ve skutečnosti spuštěn až po zadání `\input txfn` a `\def\fomenc{PX}`. Na obrázku 1 tedy vidíme kombinaci volně dostupné rodiny Charter s volně dostupnými TX fonty pro matematickou sazbu. Tyto matematické fonty vypadají hodně dobře a navíc obsahují nadmnožinu všech znaků z Computer Modern i z $\mathcal{AMS}$ fontů dohromady. Doporučuji používat!

10 Nekompatibilita

Při přechodu na novou verzi jsem byl bohužel nucen v několika bodech ustoupit od exaktní zpětné kompatibility. Tyto body nyní přesně vyjmenuji. Věřím, že se jedná o změny, které většina uživatelů vůbec nepocítí.

Jazyk deklaračních souborů textových fontů byl pouze rozšířen, tj. staré konstrukce jsou stále použitelné, ale přibyly nová slovíčka. Deklarační soubory pro novou verzi OFS nejsou tedy zpracovatelné ve verzi staré. Typickým novým slovíčkem je `\protectreading`, které se nyní snažím používat ve všech nových deklaračních souborech.

Jazyk deklaračních souborů matematických fontů byl bohužel mírně změněn. Může se tedy stát, že pokud si někdo vytvořil deklaraci matematické rodiny, nebude mu v novém OFS fungovat. Jste-li autory maker, která deklarují matematické rodiny fontů, pak doporučuji přečíst sekci 3.7 v dokumentaci [2], kde jsou změny důkladně popsány.

Další problémy s kompatibilitou mohou nastat na úrovni použití maker pro fonty Štormovy písmolijny. Tam jsem měl v deklaracích rodin docela nekoncepčně zavedeny i kódovací soubory pro kódování `6s`, zatímco při použití standardních rodin (např. ze souboru `a35.tex`) musel uživatel zavést kódovací soubory manuálně. Rozhodl jsem se kódovací soubory implicitně nezavádět nikde, takže jsem příslušný `\input` vymazal i z deklaračních souborů pro Štormovy fonty. Pokud uživatel chce načítat kódovací soubory, může v nové verzi OFS psát `\loadingenc=1`.

V původní deklaraci Štormových fontů bylo navíc aktivováno makro, které rozšiřovalo sadu matematických fontů o aktuální Štormův font, na který pak byly odkazy u těch matematických znaků, které Štorm do svých písem obvykle zahrnul (například ∞). To ale nefungovalo spolehlivě (ne ve všech Štormových fontech byly matematické znaky zastoupeny stejnou měrou), takže jsem tuto vlastnost zrušil úplně. Pro uživatele, kteří to někdy využili, jsem ponechal makro `\stmath`, jehož explicitním spuštěním se matematická sada fontů rozšíří stejně, jako ve starých deklaracích Štormových fontů.

Citelnou změnu v deklaracích mohou pocítit uživatelé, kteří používali makra definovaná na koncích kódovacích souborů a týkající se uvozovek: `\singleuv`, `\doubleuv`, `\doublefuv`, atd. Tato makra jsem z deklarací vyhodil, protože podle mého názoru patří do stylového souboru pro daný jazyk. Zařadil jsem je proto do OkTEXu. Pokud je uživatel potřebuje, může si je ze souborů `ofs-8?.tex` a `ofs-6s.tex` obkreslit. Zůstala tam, jsou ale zakomentovaná.

Šetření pamětí v příkazu `\showfonts` mě vedlo k odstranění interního příkazu `\listfamilies`. Pokud si uživatel udělal makra, která s tímto příkazem pracovala, pak toto přestane fungovat. Například mé makro `ofscatal.tex` přestalo z tohoto důvodu pracovat. Rozhodl jsem se toto makro už nepředělávat a odsunout je na smetiště dějin. Makro `ofstest.tex` totiž umí taky dělat katalogy fontů, viz například obrázek 2.

11 Obnovení dokumentace včetně anglické

Českou dokumentci jsem upravoval průběžně před provedením změn ve vlastních makrech OFS. Anglickou dokumentaci jsem pak upravil se zpožděním. Vydatně mi při tom pomohl student Tomáš Komárek. Děkuji.

Reference

1. `ftp://math.feld.cvut.cz/pub/olsak/ofs`.
2. Petr Olšák. *OFS: Olšákův fontový systém*. 2001, 2004. Dokumentace k balíku je v souborech `ofsdoc.tex`, `ofsdoc.pdf`.
3. Petr Olšák. *Makro OFS*. in: SLT 2002 – sborník semináře o Linuxu a TEXu, str. 79–92.
4. `ftp://math.feld.cvut.cz/pub/olsak/oktex`.
5. Petr Olšák. *Projekt OkTEX*. in: SLT 2004 – sborník semináře o Linuxu a TEXu.