Makro OFS

Petr Olšák

Elektrotechnická fakulta ČVUT, Praha
Email: `olsak@math.feld.cvut.cz`

Abstrakt: OFS (Olšákův fontový systém) je balíček TeXových maker, který umožní získat větší přehled nad rozsáhlými kolekcemi fontů a umožní s nimi poměrně snadnou manipulaci. Balík byl vyvinut pro získání přehledu nad fonty z Typokatalogu Střešovické písmolijny [1]. Na základě poptávky LaTeXových uživatelů byl balík napsán ještě jednou pro LaTeX, kde využívá NFSS a pokouší se je trošičku vylepšit. Základní uživatelské příkazy OFS jsou pak v obou prostředích (plain i LaTeX) stejné. Na přednášce předvedu použití OFS na uživatelské i konfigurační úrovni.

Úvodem

Makro OFS jsem si napsal hlavně proto, abych se vyznal ve stovkách fontů, které pocházely ze Střešovické písmolijny a pro které jsem před rokem udělal TeXovou podporu. Protože jsem plainista, šlo mi hlavně o to udělat makro co nejvíce srozumitelné uživateli plainu, který potřebuje přesně vědět, co to makro dělá. Tento požadavek například LaTeXové NFSS nesplňuje.

Později se začali o OFS zajímat též někteří uživatelé LaTeXu. Tož jsem se přemohl a pokusil OFS napsat ještě jednou, tentokrát pod LaTeXem s využitím NFSS. Cílem tohoto přepsání bylo hlavně to, aby se uživatelské příkazy pro vyhledávání a přepínání rodin fontů zcela shodovaly v LaTeXové verzi s verzí plainovou. LaTeXová verze přesto umí podstatně méně věcí, než plainová, protože pokud by měla umět vše, musel bych NFSS zcela odmítnout a fontový modul LaTeXu si napsat po svém. Tím bych ale popřel LaTeX jako takový, takže jsem zůstal u respektování principů NFSS. Věci, které se v NFSS dají dělat velmi těžko, jsem raději nedělal.

Z těchto důvodů jsem se v LaTeXu nepouštěl ani do podpory matematických fontů. Koncepce matematiky v NFSS mi připadá jako plainistovi poněkud nesrozumitelná. Navíc velké kolekce fontů, které byly hlavní motivací OFS, jsou většinou textové. Pokud chce někdo použít matematickou sadu fontů v LaTeXu, použije `\usepackage{styl}` a ten styl je k té matematické sadě většinou dodáván. Nepovažoval jsem tedy řešení matematiky v LaTeXu za prioritní. Na druhou stranu matematické rodiny v plainu se pomocí OFS doplňují a zavádějí ve všech velikostech velmi elegantně. Věřím, že alespoň nějakému plainistovi (kromě mě) se toto řešení bude hodit.

Makro OFS jsem zveřejnil na [2] pod TeX-like licencí, tj. je volně k mání, ale změny pod stejným názvem nesmí dále šířit nikdo jiný, než autor. Do TeXových

Jan Kasprzak, Petr Sojka (editoři): SLT 2002 – sborník semináře o Linuxu a TeXu, str. 79–92, 2002.

distribucí jsem makro zatím neprosadil, protože bohužel chybí anglická dokumentace. Domnívám se, že kdyby anglická dokumentace byla, nebyl by se zařazením do TEXových distribucí problém a makro by mohlo využít daleko více lidí. Bohužel, nejsem v angličtině natolik zdatný, abych tento problém rychle překonal. Pro letošní letní prázdniny jsem sice plánoval, že se pokusím anglickou dokumentaci napsat, jenže přišla voda…

K OFS existuje samozřejmě česká dokumentace [3] podrobě popisující chování makra. Navíc jsem se o OFS zmínil už v článku o TEXové podpoře Štormových fontů [4]. Dovolil jsem si zde přesto znovu k tomuto tématu vrátit. Nechci být nyní tak technicky exaktní, jako v manuálu [3], ale zase mám zde více místa k rozepsání možností makra, než jsem měl ve článku [4], který byl především o Štormových fontech.

2 Základy uživatelského rozhraní

Uživatelské rozhraní je shodné v LATEXové verzi OFS i v plainové s výjimkou snad zavedení makra OFS, které v plainu provedeme jednoduše pomocí `\input ofs [kolekce, fontů]`, zatímco v záhlaví LATEXového dokumentu píšeme `\usepackage [kolekce, fontů] {ofs}`.

Abych v tomto textu nemusel každou chvíli větvit svůj výklad na situaci vhodnou pro plain a pro LATEX, rozhodl jsem se zde předpokládat, že pracujeme pouze s plainem (například s CSplainem). LATEXový uživatel si bude muset dohledat specifika verze OFS pro svůj formát v dokumentaci [3].

Příkazem `\fontusage` dostaneme na terminál a do logu základní informace o uživatelských příkazech:

```
$ tex ofs \\fontusage
This is TeX, Version 3.14159 (Web2C 7.3beta5)
(/usr/local/share/texmf/tex/csplain/ofs.tex
OFS (Olsak's Font System) based on plain initialized. <Oct. 2002>
(/usr/local/share/texmf/tex/csplain/ofsdef.tex))
\fontusage: ============== Olsak's Font System, usage: =================
\input ofs [sjannon, sdynamo, a35] ... for example
\showfonts  ... shows all loaded font families (by previous \input)
\setfonts [Family/] ... local switch to the new family, after this, the
  \rm, \bf, \it, bi will switch to the variants. The current size is used.
\setfonts [/size] ... local switch to the new size of fonts, the family is
  not changed. The "size" has the following possible formats:
    at<dimen>       ... the same as \font\something=file at<dimen>
    <dimen>         ... the same as at<dimen>
    <number>        ... the same as at<number>pt
    scaled<number>  ... the same as \font\something=file scaled<number>
    mag<decimal-number> fonts will be magnified by given coefficient
                        depend on current size of the fonts.
\setfonts [Family/size] ... switch to the new family at given size
\setfonts [Family-vr/]  ... switch to the specified font, the current size
  is used. The "vr" is acronym for variant (bf for example).
\setfonts [Family-vr/size] ... switch to the specified font.
\fontdef\name [Family/size] ... same as \gdef\name{\setfonts[Family/size]}
```

```
  The "Family" or "size" parameter may be empty.
\fontdef\name [Family-vr/size] ... \name is fixed-font switch iff:
  "size" is no empty and no mag<dec-number>.
  Fixed-font switch "\name" is implemented as \global\font\name=file.
\setmath [size/size/size] ... set math it/rm as current it/rm + use PS Symbol
\nofontmessages, \logfontmessages, \displayfontmessages, \detailfontmessages
  ... the levels of log.
*
```

Vidíme, že příkaz `\showfonts` nám ukáže kolekce fontů. Pokud jsme nepoužili „previous `\input`“, dopadne základní kolekce (implementovaná přímo uvnitř OFS) takto:

```
*\showfonts
OFS (1.0): The list of known font families:
defaults:
   [CMRoman/]           \rm, \bf, \it, \bi, \sl
   [CMSans/]            \rm, \bf, \it,  -
   [CMTypewriter/]      \rm,  - , \it,  - , \sl
   [Times/]             \rm, \bf, \it, \bi
   [Helvetica/]         \rm, \bf, \it, \bi, \nrm, \nbf, \nit, \nbi
   [Courier/]           \rm, \bf, \it, \bi
```

Názvy fontových rodin jsou zde uvedeny v hranatých závorkách a vedle jsou uvedeny přepínače variant, které pro danou rodinu je možné použít.

Vidíme, že nejběžnější textové fonty Computer Modern jsou zaneseny v OFS do tří rodin `CMRoman`, `CMSans` a `CMTypewriter`. Myslím, že netřeba dodávat, co to znamená. V rodinách `CMSans` a `CMTypewriter` není k dispozici jinak obvyklá varianta `BoldItalic`, pro kterou je vyhrazen přepínač `\bi`. V rodině `CMTypewriter` nenajdeme ani přepínač `\bf` pro tučnou variantu, ale zato můžeme použít „nadstandardní“ variantu `\sl` (slanted), která je k dispozici i v rodině `CMRoman`.

Deklarace dalších písmových rodin jsou zanášeny do deklaračních souborů s příponou `tex`[1]. V každém distribuci TEXu doporučuji udržovat soubor volající všechny deklarační soubory `allfonts.tex`. Například na mém počítači vypadá tento soubor takto:

```
$ cat `kpsewhich allfonts.tex`
%%% All OFS families declared on this TeX
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%                           Petr Olsak
\input a35       % PostScript 35
\input ffonts    % Another free fonts
\input btfonts   % Bitstream fonts
\input skatalog  % Stromtype foundry, 89 families
```

Když tedy napíšu `$ tex allfonts \\showfonts \\end | less` , dostanu na svém počítači výpis zhruba tří set písmových rodin, každá obvykle ve čtyřech

[1] V LATEXovém OFS se jedná o příponu `sty`.

variantách. Důležité je, že se v tom vyznám a nemusím vzpomínat, jak se která metrika jmenuje. Následuje jen část výpisu.

```
OFS (1.0): The list of known font families:
defaults:
   [CMRoman/]           \rm, \bf, \it, \bi, \sl
   [CMSans/]            \rm, \bf, \it,  -
   [CMTypewriter/]      \rm,  - , \it,  - , \sl
   [Times/]             \rm, \bf, \it, \bi
   [Helvetica/]         \rm, \bf, \it, \bi, \nrm, \nbf, \nit, \nbi
   [Courier/]           \rm, \bf, \it, \bi
a35.tex:
   [AvantGarde/]        \rm, \bf, \it, \bi
   [Bookman/]           \rm, \bf, \it, \bi
   [NewCentury/]        \rm, \bf, \it, \bi
   [Palatino/]          \rm, \bf, \it, \bi
   [ZapfChancery/]      \rm,  - , \it,  -
   [ZapfDingbats/]      \rm,  - ,  - ,  -
   [Symbol/]            \rm,  - , \it,  -
ffonts.tex:
   [Charter/]           \rm, \bf, \it, \bi
...
sjannon.tex:
   [JannonAntikva/]     \rm, \bf, \it, \bi, \mr, \mi
   [JannonText/]        \rm, \bf, \it, \bi, \mr, \mi
   [JannonCaps/]        \rm, \bf, \it, \bi
...
sdynamo.tex:
...
   [DynaGroteskLE/]     \rm, \bf, \it, \bi
   [DynaGroteskD/]      \rm, \bf, \it, \bi
   [DynaGroteskR/]      \rm, \bf, \it, \bi
...
stitul.tex:
   [Alcoholica/]        \rm, \bf, \it,  -
   [Monarchia/]         \rm, \bf,  - ,  -
   [MonarchiaText/]     \rm, \bf,  - ,  -
   [Clichee/]           \rm, \bf, \it, \bi
   [Regula/]            \rm,  - , \it,  -
   [Splendid/]          \rm, \bf, \it,  - , \script, \sans
   [Cobra/]             \rm, \bf,  - ,  -
   [ExcelScript/]       \rm,  - ,  - ,  - , \ext
   [ExcelScriptText/]   \rm,  - ,  - ,  - , \ext
   [Zeppelin/]          \rm, \bf,  - ,  - , \lr, \coll
   [Negro/]             \rm,  - ,  - ,  -
   [Farao/]             \rm, \bf,  - ,  - , \kr, \coll
...
slido.tex:
   [Lido/]              \rm, \bf, \it, \bi, \crm, \cbf
```

Ve výpisu je uveden i TEXový soubor, kde se deklarace odpovídajících rodin skutečně nachází. Vidíme tedy, že `a35.tex` už obsahuje deklarace rodin, ale například soubor `skatalog.tex` zřejmě obsahuje jen další `\input sjannon`,

`\input sdynamo`, `\input stitul` atd. Když si pro svůj dokument vyberu třeba rodiny `JannonText`, `DynaGroteskR` a `Farao`, pak místo toho, abych psal

```
\input ofs [allfonts]
```

si vystačím s

```
\input ofs [sjannon, sdynamo, stitul]
```

což způsobí načítání podstatně méně souborů s deklaracemi fontových rodin.

Pokud potřebuji vědět, jak ty fonty vypadají, napíšu například:

```
$ pdfcsplain ofscatal [sjannon, sdynamo, stitul] ; acroread ofscatal.pdf
```

Soubor `ofscatal.tex`, který vytvoří katalog, byl zařazen do balíku OFS od verze `Oct 2002`.

3 Přepínač rodin a velikostí

Hlavním příkazem OFS na uživatelské úrovni je `\setfonts`. Jedná se především o přepínač rodin, viz výpis `\fontusage` v předchozí sekci. Přepínač má dva parametry v hranaté závorce oddělené lomítkem. Prvním parametrem je název rodiny a druhým požadovaná velikost fontů. Pokud některý parametr chybí, přepínač nebude jeho nastavení měnit. Přepínač samozřejmě funguje lokálně ve skupinách, jako bývá u fontových přepínačů obvyklé. Po přepnutí rodiny pracují přepínače variant (obvykle `\rm`, `\bf`, `\it`, `\bi`) pro novou rodinu. Varianta se po přepnutí rodiny inicializuje stejná, jako byla před přepnutím, tj. jsme-li zrovna v kurzívě rodiny Times a přepneme do Helveticy, zůstáváme ve variantě `\it` rodiny Helvetica, tj. skloněné písmo. Pokud ale při přepnutí rodiny aktuální varianta v nové rodině neexistuje, inicializuje se varianta `\rm`, kterou musí obsahovat každá rodina.

Příklady

```
\setfonts [JannonText/10.5]  % nastavím na začátku dokumentu
\setfonts [/14]\bf           % například pro nadpisy
\setfonts [/8]\rm            % pro poznámky pod čarou
\setfonts [DynagroteskR/]\it % třeba pro citáty
\setfonts [CMTypewriter/]    %  pro strojopis.
       % tento přepínač např. v kurzívě nastaví automaticky
       % variantu kurzíva-strojopis.
```

Výhodou tedy je, že si nemusím pamatovat názvy matrik (v LaTeXu pak nesrozumitelné zkratky rodin používané v NFSS), ale píšu název rodiny do dokumentu stejným způsobem, jak jej vidím v písmovém katalogu. Pokud udělám v názvu rodiny překlep (třeba nedodržím velká a malá písmena), příkaz `\setfonts` spustí `\showfonts`, tj. na obrazovce a v logu vidím seznam všech rodin, které mohu použít.

Velikost fontů mohu nastavit přímo udáním v jednotce `pt`, nebo připsáním jiné jednotky (např. `mm`), nebo jako u primitivu `\font` pomocí slova `scaled` (koeficient zvětšení se váže k základní velikosti fontu) a konečně i pomocí zcela nového prefixu mag, který udává zvětšení vzhledem ke zrovna použité velikosti fontů, například:

```
\def\maly{\setfonts [/mag0.8]}
stanu se {\maly menším a {\maly menším a {\maly ještě menším}}}
a už jsem se z toho dostal.
```

dopadne takto:

stanu se menším a menším a ještě menším a už jsem se z toho dostal.

Prefix mag pro velikost využijeme například v logu LATEX, které jsem si v plainu definoval jako

```
\def\LaTeX{L\kern-.2em\raise.45ex\hbox{\setfonts[/mag.7] A}\kern-.05em\TeX}
```

a funguje to v nadpisech (je tam tučné vyvýšené A odpovídající velikosti) i v poznámkách pod čarou. V LATEXu samotném mají toto logo definováno tak, že se pro vyvýšené A použije font v indexové velikosti. No jo, zde jim to náhodou prochází, ale co kdyby se (například v jiném v logu) hodilo pro některé písmeno použít velikost mag.8, která není jako indexová velikost použita? Bez OFS bychom to pak dělali dost obtížně, protože NFSS nám řešení nenabízí.

Dalším využitím prefixu mag je možnost korigovat nestejné střední výšky písma. Například v tomto sborníku je pro strojopis použit CMTypewriter. Pokud si všimnete v jiných příspěvcích, než tento, vidíte, že uvnitř odstavců ten strojopis s fontem Charter moc nesedí, protože strojopis má menší střední výšku písma. Stačí ale pro přepínač \tt použít \setfonts [CMTypewriter/mag1.1], a střední výška je v lati. To vidíme například v tomto příspěvku. A funguje to včetně zmenšených variant, které jsou v příspěvcích pro SLT použity v abstaktu, poznámce pod čarou, nebo třeba v údaji „Email“, uvedeném v záhlaví příspěvku.

Příkaz \setfonts může obsahovat i specifikaci varianty (viz výpis příkazu \fontusage). Pak se už nejedná o přepínač rodiny, ale fontu samotného. V takovém případě přepínač \setfonts neovlivní přepínače variant \rm, \bf a dalších, ale nastaví jen požadovaný jediný font. Například pro nadpisy by šlo psát \setfonts[-bf/14] místo \setfonts[/14]\bf, ale museli bychom mít jistotu, že v nadpisu nepoužijeme přepínač varianty, protože ten by vrátil font do velikosti aktuální rodiny. Tou velikostí je třeba 10pt.

Abychom v nadpisu mohli použít přepínač \it, a přitom se dostali do varianty \bi (což je žádoucí), je potřeba přepínač fontu v makru pro nadpis naprogramovat zhruba takto:

```
\def\nadpis #1{{\setfonts[/14]\bf \let\it=\bi #1}}
```

Kvůli tomuto jednoduchému obratu, který zvládne snad každý plainista, jsem se rozhodl nekomplikovat jádro OFS podobným způsobem jako NFSS a nezavádět tedy další „nezávislou souřadnici“ popisující duktus fontu. Pro speciální rodiny fontů (jako je třeba rodina DynaGrotesk) jsem ale snadno pomocí doplňujících maker vytvořil přepínač, který respektuje ještě o jednu „nezávislou souřadnici“ více, než zvládá NFSS. V tomto případě to nebyl jen duktus, ale i stupeň zúžení písma. Navíc tento přepínač dokáže „poskakovat“ po jednotkách vpřed a vzad podél zvolené souřadnice. To také NFSS nezvládá.

Často se hodí vytvářet zkratky pro přepínače rodin. K tomu slouží makro `\fontdef\přepínač[Rodina/velikost]`, které funguje podobně, jako použití konstrukce `\gdef\přepínač{\setfonts[Rodina/velikost]}`. V záhlaví tohoto dokumentu mám například uvedeno:

```
\input ofs [ffonts]              % Charter je ve skupině free fonts
\setfonts [Charter/10pt]         % výchozí rodina
\fontdef\tt [CMTypewriter/mag1.1]  % strojopis, korekce střední výšky
\fontdef\verbtt [CMTypewriter/8]   % strojopis pro display ukázky
\fontdef\small [Charter/9]         % zmenšení pro abstrakt a záhlaví
```

Možná čtenáře napadne, že jsem nemusel psát podruhé slovo `Charter` v deklaraci přepínače `\small`. Při kompletní změně základní rodiny dokumentu na jinou bych pak mohl změnit slovo `Charter` jen na jediném místě (u příkazu `\setfonts`) a byl bych hotov. Bohužel, vynechání rodiny v příkazu `\small` vede k problémům, protože tento příkaz je použit v záhlaví dokumentu, tj. v `\output` rutině. Pokud zde neuvedeme explicitně rodinu, dědí se rodina aktuálně použitá, která ovšem může být jakákoli, protože `\output` rutina je vyvolávaná z různých míst při zpracování dokumentu. Mít záhlaví jednou strojopisem a podruhé Charterem podle toho, odkud byla `\output` rutina zavolána, je samozřejmě nežádoucí.

Tento problém se dá řešit „vykřičníkovou" konvencí, kterou umí zpracovat příkaz `\fontdef`:

```
\fontdef\small [!/9] \addcmd\small {\rm} % pro abstrakt a záhlaví
```

Vykřičník je nahrazen aktuální rodinou už v době činnosti příkazu `\fontdef`, nikoli tedy až v době provádění přepínače. V `\output` rutině se musíme ještě postarat o potlačení dědičnosti aktuální varianty. Proto jsem použil ještě příkaz `\addcmd` (zaveden nově od verze OFS `Oct 2002`), který k existujícímu makru přidá další příkazy na jeho konec. Celá deklarace fontů v tomto dokumentu tedy vypadá zhruba takto:

```
\input ofs [ffonts]                   % Charter je ve skupině free fonts
\setfonts [Charter/10pt]              % výchozí rodina
\fontdef\tt [CMTypewriter/mag1.1]     % strojopis, korekce střední výšky
\fontdef\verbtt [CMTypewriter-rm/8]   % strojopis pro display ukázky
\fontdef\small [!/9]
\addcmd \small {\baselineskip11pt\rm} % zmenšení pro abstrakt a záhlaví
\fontdef\fontsekce [!/12]
\addcmd \fontsekce {\bf \let\it=\bi}  % pro nadpisy sekcí
\fontdef\fonttitul [!-bf/14.4]        % v titulu nebudeme přepínat varianty
```

Povedlo se tedy veškeré aktivity související s výběrem fontů soustředit na jedno místo dokumentu bez nutnosti používat názvy metrik.

4 Kódování fontů

OFS pro plain implicitně pracuje v kódování $\mathcal{CS}$fontů, ale můžete jej přepnout do jiného kódování. [2]

[2] OFS pro LaTeX přenechává starost o kódování zcela na NFSS.

Základem informace o kódování v OFS pro plain je makro `\fotenc`, které má implicitně hodnotu `8z`. To znamená, že rodiny `CMRoman`, `CMTypewriter` a `CMSans` budou pracovat s CSfonty. Kdyby uživatel nastavil `\def\fotenc{8t}`, začalo by OFS pracovat v případě těchto rodin s DC nebo EC fonty. Jaké je technické pozadí vysvětlíme v následující sekci.

Většina metrik PostScriptových fontů má varianty `*8t` (pro kódování podle Corku) a `*8z` (pro kódování podle CSfontů). Makro `\fotenc` tedy musí obsahovat koncovku metrik, které chceme použít. Tím volíme kódování. Příklad:

```
\input ofs
\setfonts [Times/]  text 1 % metrika: ptmr8z, tj. kódování CSfontů
\def\fotenc{8t}
\setfonts [/]       text 2 % metrika: ptmr8t  tj. kódování podle Corku
```

Máte-li své fonty v nějakém dalším kódování a koncovky metrik máte s názvy třeba `*8x`, pak není problém napsat `\def\fotenc{8x}` a můžete začít používat své vlastní metriky.

Pokud používáte kódování podle Corku, doporučuji definovat makro `\fotenc` jako `8t` před zavedením makra OFS. Při použití složitěji deklarovaných rodin se to hodí.

V balíku OFS jsou soubory `ofs-8z.tex` a `ofs-8t.tex`, ve kterých je deklarace akcentů a některých na kódování závislých maker, jako například `\promile`. Implicitně není načten ani jeden z těchto souborů, tj. jsou respektována makra pro akcenty z originálního plainu, která expandují na `\accent` podle kódování CMfontů.

Pokud je toto chování nevyhovující, můžete načíst pomocí `\input` jeden nebo oba dva výše zmíněné soubory (v libovolném pořadí – jejich definice se nehádají). Pokud načtete oba soubory, pak makra pro akcenty expandují na znaky podle kódování, které je aktuálně nastaveno v makru `\fotenc` (`8z` nebo `8t`).

Podívejme se do souborů `ofs-8z.tex` a `ofs-8t.tex` (výrazně kráceno):

```
%%% Default accents in CM
\accentdef  \` *    8z  {\accent 18 }              % grave
\accentdef  \' *    8z  {\accent 19 }              % acute
\accentdef  \v *    8z  {\accent 20 }              % caron
\accentdef  \u *    8z  {\accent 21 }              % breve
...
%%% Standard characters in plain (redefined here)
\def\aa{\r a}
\def\AA{\r A}
\characterdef \i    8z  16
\characterdef \j    8z  17
\characterdef \SS   8z  {SS}
\characterdef \AE   8z  29
...
%%% Extra characters from CS-fonts
\characterdef \promile      8z  141
\characterdef \varhyphen    8z  156
\characterdef \flqq         8z  158
```

```
\characterdef \frqq        8z  159
\characterdef \clqq        8z  254
\characterdef \crqq        8z  255
...
%%% Accented letters from CS fonts
\accentdef  \‘ A      8z  152
\accentdef  \’ A      8z  193
\accentdef  \" A      8z  196
\accentdef  \‘ a      8z  184
\accentdef  \’ a      8z  225
...
%%% Default accents in Cork
\accentdef  \‘ *    8t  {\accent 0 }
\accentdef  \’ *    8t  {\accent 1 }
\accentdef  \^ *    8t  {\accent 2 }
\accentdef  \  *    8t  {\accent 3 }
\accentdef  \" *    8t  {\accent 4 }
...
%%% Standard characters in plain (redefined here)
\def\aa{\r a}
\def\AA{\r A}
\characterdef \i    8t  25
\characterdef \j    8t  26
\characterdef \SS   8t  223
\characterdef \AE   8t  198
...
\characterdef \promile             8t {\%\char 24 }
\characterdef \textpertenthousand   8t {\%\char 24\char 24 }
...
%%% Accented letters from Cork encoding
\accentdef  \. i    8t ‘\i
\accentdef  \u A    8t 128
\accentdef  \k A    8t 129
\accentdef  \’ C    8t 130
\accentdef  \v C    8t 131
...
```

Myslím, že syntaxe a význam příkazů `\characterdef` a `\accentdef` je v této ukázce samovysvětlující. Pokud ne, odkazuji čtenáře do dokumentace [3], která do posledního detailu (včetně popisu jednotlivých fází expanze) vysvětluje činnost těchto příkazů.

LaTeXovému uživateli to může připomínat příkazy `\DeclareTextSymbol` a `\DeclareTextComposite` a jim podobné, které dělají zhruba totéž (jen poněkud komplikovaněji, nepřehledněji a těžkopádněji).

OFS také počítá s možností, že některé znaky jsou přidány do extra fontu, který s původním fontem vytváří uspořádanou dvojici. Takové dvojice metrik jsou použity například pro fonty Štormovy písmolijny, protože obsahují více než 256 znaků. Přitom by bylo škoda některé znaky nevyužít. Ke každé metrice těchto fontů (`*8z` i `*8t`) je přiřazena extra metrika `*6s`, obsahující zbylé znaky. Pomocí `\characterdef` a `\accentdef` můžeme deklarovat přítomnost těchto znaků v kódování `6s` (viz soubor `stormenc.tex`). Pokud aktuální metrika má k sobě extra font v kódování 6s a je požadován znak z tohoto fontu, OFS jej automaticky „vyloví“ pomocí přechodného přepnutí na extra metriku.

5 Deklarace rodin

Podívejme se nyní do souboru a35.tex, který deklaruje rodiny pro základních 35 PostScriptových fontů (kráceno):

```
%%% Times, Helvetica, Courier is in OFS defaults
\ofsdeclarefamily [AvantGarde] {% -------------------- AvantGarde
   \loadtextfam (Book)         pagk\fotenc;%
                (Demi)         pagd\fotenc;%
                (BookOblique)  pagko\fotenc;%
                (DemiOblique)  pagdo\fotenc;;%
   \def\TeX{T\kern-.08em\lower.3333ex\hbox{E}\kern-0.09emX}%
}
\ofsdeclarefamily [NewCentury] {% -------------------- NewCenturySchlbk
   \loadtextfam (Roman)        pncr\fotenc;%
                               pncb\fotenc;%
                               pncri\fotenc;%
                               pncbi\fotenc;;%
   \def\TeX{T\kern-.1667em\lower.5ex\hbox{E}\kern-.125emX}%
}
...
```

Zde je tedy deklarováno zobrazení mezi názvy rodin a metrikami.[3] Každá rodina má čtyři povinné metriky (pro běžné čtyři varianty). Metriky jsou zapsány pomocí \fotenc, aby fungovalo přepínání kódování. Mezi předposledním a posledním středníkem může být uvedeno extra kódování, pokud je font rozložen do dvou metrik (v tomto příkladě není). Příkazy uvedené v parametru \ofsdeclarefamily se provedou při každém přepnutí rodiny příkazem \setfonts. V ukázce tam máme alternativní definice loga \TeX, aby toto logo pěkně vypadalo v každém fontu.

Od verze OFS Oct 2002 je přidána možnost použít nepovinné parametry v argumentech příkazu \loadtextfam. Tyto parametry píšeme do kulaté závorky a značí název varianty, pokud je odlišný od běžného názvu. Tato informace se použije v logu a při tisku katalogu.

Podívejme se ještě do souboru sjannon.tex z podpory Štormových fontů (výpis je zde neúplný):

```
\ofsdeclarefamily [JannonAntikva] {% ------- Jannon Antikva
   \loadtextfam sjnr\fotenc;%
                sjnb\fotenc;%
                sjnri\fotenc;%
                sjnbi\fotenc;6s;%
   \newvariant2 \mr (Medium)       sjnm\fotenc;6s;%
   \newvariant3 \mi (MediumItalic)  sjnmi\fotenc;6s;%
```

[3] V OFS pro LaTeX takové soubory neexistují; tam se pouze deklaruje zobrazení mezi (dlouhými) názvy rodin v OFS a zkratkami rodin v NFSS.

```
}
\ofsdeclarefamily [JannonText] {% ----------- Jannon Text
   \loadtextfam sjnrg\fotenc;%
                sjnbg\fotenc;%
                sjnrig\fotenc;%
                sjnbig\fotenc;6s;%
   \newvariant2 \mr (Medium)       sjnmg\fotenc;6s;%
   \newvariant3 \mi (MediumItalic)  sjnmig\fotenc;6s;%
}
...
```

Vidíme, že navíc je zde deklarována extra metrika v kódování `6s` a že se zde deklarují doplňující přepínače variant `\mr` a `\mi` mimo základní čtyři varianty. Podrobnější vysvětlení použitých příkazů čtenář opět najde v dokumentaci [3].

Jak jsou pomocí expanze `\fotenc` mapovány metriky $\mathcal{CS}$ fontů resp. DC fontů, které v názvech koncovku `8z` ani `8t` nemají? To souvisí rovněž s tím, že tyto fonty mají pro různé velikosti různé metriky. V deklaračních souborech je pak tento problém vyřešen způsobem „dvě mouchy jednou ranou“ (viz soubor `ofsdef.tex`):

```
\registertfm cmr8z     -      csr10  % metrika pro všechny velikosti
\registertfm cmr8z  0pt-6pt   csr5
\registertfm cmr8z  6pt-7pt   csr6
\registertfm cmr8z  7pt-8pt   csr7
\registertfm cmr8z  8pt-9pt   csr8
\registertfm cmr8z  9pt-10pt  csr9
\registertfm cmr8z  10pt-12pt csr10
\registertfm cmr8z  12pt-17pt csr12
\registertfm cmr8z  17pt-*    csr17
...
\registertfm cmr8t     -   dcr10     % metrika pro všechny velikosti
...
\ofsdeclarefamily [CMRoman] {% ----------- Computer Modern Roman
   \loadtextfam cmr\fotenc;%
                cmbx\fotenc;%
                cmti\fotenc;%
                cmbxti\fotenc;;%
   \newvariant8 \sl (Slanted) cmsl\fotenc;;%
}
```

Centrálním příkazem je zde makro `\registertfm`, které mapuje neexistující metriky formálně vytvořené pomocí expanze `\fotenc` do skutečných metrik. Toto makro také umí registrovat různé metriky pro různé velikosti. Pokud tedy budeme chtít font daný pomocí `\setfonts[CMRoman-rm/12.5]`, vyvolá se při `\fotenc` s hodnotou `8z` metrika `csr12 at12.5pt`. „Metrika pro všechny velikosti“ se použije v případě, že je velikost fontu deklarovaná s prefixem `scaled`, takže `\setfonts[CMRoman-rm/scaled1250]` vyvolá metriku `csr10 scaled1250`. Ztrácíte-li přehled o tom, jaká metrika se ve skutečnosti použila, můžete zapnout logovací přepínač `\detailfontmessages`.

Podíváte-li se do souboru `ofsdef.tex` podrobněji, můžete si všimnout, že jsem metriky rodiny `CMRoman` a dalších `CM` rodin pro kódování `8t` poněkud odbyl. Registroval jsem je jako metriky DC fontů bez střídání metrik pro různé velikosti. Nechtělo se mi to totiž vypisovat a nevěděl jsem, zda uživatel tohoto kódování raději nepoužije EC fonty místo DC fontů. Ve zmíněném souboru jsou příklady, jak by se ta deklarace měla provést. Pokud ji někdo dopíše (například pro DC i EC fonty do zvláštních souborů), rád ji do balíku OFS zařadím. Osobně ale fonty kódované v `8t` nepoužívám, takže mě zatím nic nemotivuje to udělat.

6 Matematické fonty

Každý plainista ví, že fonty pro matematiku se sdružují do matematických rodin obsahující font pro základní, indexovou a index-indexovou velikost. Rovněž ví, že první čtyři matematické rodiny (s číslem 0 až 3) jsou s TeXem jistým způsobem pevně významově svázány a další rodiny se dají deklarovat.

Aby plainista při deklarování matematických rodin nemusel psát třikrát za sebou `\font` a potom ještě `\textfont\rodina=...`, `\scriptfont...`, atd, je v OFS k tomu vytvořena zkratka `\loadmathfam`, jak ukážu za chvíli.

Inicializaci matematických fontů v OFS pro plain provedeme pomocí příkazu `\setmath`. Dokud tento příkaz nepoužijeme, jsou matematické fonty ve stavu, jak je inicializoval plain. Ve třech parametrech příkazu `\setmath` (oddělených lomítky) dáváme najevo, jakou chceme základní velikost, jakou pro indexy a jakou pro indexy indexů. Jsou-li tyto parametry prázdné, je použito relativní zmenšení podle aktuální velikosti textového fontu pomocí prefixu `mag` takto:

```
\setmath [//] je totéž jako \setmath [mag1.0/mag.7/mag.5]
```

Příkaz `\setmath` vypočítá požadované velikosti a startuje makro pro zavedení fontů `\mathfonts`. Dále tento příkaz startuje makro pro inicializaci matematického kódování `\mathchars`. Tato makra si může plainista definovat jak chce, nicméně většinou využije již připravených maker `\defaultmathfonts` a `\defaultmathchars`, která jsou udělána tak, že spouštějí různé varianty kódu podle hodnoty maker `\fomenc` a `\mathversion`.

Při `\def\fomenc{PS}` (imlicitní hodnota: PostScriptové fonty) se zavedou fonty tak, že matematická kurzíva se ztotožní s aktuální textovou kurzívou a podobně rodina 0 pro číslice a textové symboly zůstává nastavena podle aktuální textové rodiny varianty `\rm`. Matematické symboly se berou (pokud to je možné) z běžně dostupného PostScriptového fontu `Symbol`. Zbytek (např. natahovací závorky) pak zůstává v Computer Modern. Matematické kódování je pro tuto situaci výrazně pozměněno obvyklými primitivy `\mathchardef` a podobnými, aby byly dosažitelné všechny znaky plainu. Například pro řecká písmena musela být zavedena nová rodina se skloněným fontem `Symbol` a příkazy typu `\alpha` jsou překódovány z původní matematické rodiny 1 na tuto novou rodinu.

Nastavíte-li `\def\fomenc{CM}`, pak příkaz `\setmath` zavede stejné matematické fonty z rodiny Computer Modern, jako v plainu. Příkaz `\setmath[//]` pak pouze aktualizuje velikosti těchto fontů podle velikosti aktuálního textového fontu.

Nakoupíte-li fonty MathTimes, dále načtete soubor `ofsmtdef.tex` (součástí OFS od verze `Jun. 2002`) a definujete `\def\fomenc{MT}`, pak příkaz

`\setmath` zavede kurzívu a rodinu 0 stejně jako při `\def\fomenc{PS}`, ale navíc použije pro *všechny* symboly včetně natahovacích závorek fonty MathTimes.

Kromě hodnoty `\fomenc` se příkaz `\setmath` větví i vzhledem k hodnotě makra `\mathversion`. Implicitně OFS počítá se dvěma hodnotami tohoto makra: `\def\mathversion{normal}` nebo `bold`. Při verzi `bold` jsou načteny do matematických rodin tučné alternativy fontů, pokud to jde. Vše názorně vysvětlí pohled do souboru `ofsdef.tex`:

```
\def\defaultmathfonts{\csname load\fomenc\mathversion math\endcsname}
\def\defaultmathchars{\csname set\fomenc mathchars\endcsname}
\def\mathfonts{\defaultmathfonts}
\def\mathchars{\defaultmathchars
   \let\mathchars=\relax % to protect the twice math-setting
}
\def\loadPSnormalmath{%
  \loadmathfam 0[tenrm/]%             Actual Roman font
  \loadmathfam 1[tenit/]%             Actual Italic font
  \loadmathfam 2[/cmsy]%              Standard symbols from CM
  \noindexsize\loadmathfam 3[tenex/]%  Standard extra symbols from CM
  \chardef\itfam=1 \chardef\bifam=5
  \loadmathfam \bffam [tenbf/]%       Actual Bold font
  \loadmathfam \bifam [tenbi/]%       Actual Bold Italic
  \newmathfam\symbfam
  \loadmathfam \symbfam [/psyr]%      PostScript Symbol
  \newmathfam\symbofam
  \loadmathfam \symbofam [/psyro]%    PostScript Symbol Oblique
}
\def\loadPSboldmath{%
  \loadmathfam 0[tenbf/]%             Actual Bold font
  \loadmathfam 1[tenbi/]%             Actual Bold-Italic font
...
}
\def\loadCMnormalmath{%
  \loadmathfam 0[/cmr8z]%                   Roman font
  \loadmathfam 1[/cmmi]%                    Math Italic font
  \loadmathfam 2[/cmsy]%                    Standard symbols from CM
  \noindexsize\loadmathfam 3[/cmex10]%      Standard extra symbols from CM
...
}
\def\loadCMboldmath{%
  \loadmathfam 0[/cmbx8z]%                    Roman font
  \loadmathfam 1[/cmmib10]%                   Math Italic font
...
}
\def\setPSmathchars{%
  \fontmessage{\ofsmessageheader Math codes are set for PS encoding}%
  \mathcode`\,="602C
  \mathcode`\.="002E
  \delcode`\<="\hex\symbfam E130A
  \delcode`\>="\hex\symbfam F130B
  \delcode`\|="\hex\symbfam 7C30C
  \edef\langle{\delimiter"4\hex\symbfam E130A }%
  \edef\rangle{\delimiter"5\hex\symbfam F130B }%
  \mathchardef\alpha    "0\hex\symbofam 61
```

```
  \mathchardef\beta     "0\hex\symbofam 62
  \mathchardef\gamma    "0\hex\symbofam 67
...
}
```

Chceme-li přidat nějaké další matematické rodiny (v terminologii NFSS matematické abecedy), pak můžeme postupovat třeba takto:

```
\def\mathfonts{\defaultmathfonts
  \newmathfam\bbfam
  \loadmathfam \bbfam [/bbold12]%  Dvojitá vertikální kresba
  \def\bb{\fam\bbfam}%
}
\def\mathchars{\defaultmathchars
  \mathchardef\balpha    "0\hex\bbfam 0B
  \mathchardef\bbeta     "0\hex\bbfam 0C
  ...
}
```

Příklad z konce sekce 3 nyní obohatíme o možnost práce s matematikou v libovolných velikostech:

```
\input ofs [ffonts]                  % Charter je ve skupině free fonts
\setfonts [Charter/10pt]             % výchozí rodina
\setmath[//]                         % inicializace matematiky
\fontdef\tt [CMTypewriter/mag1.1]    % strojopis, korekce střední výšky
\fontdef\verbtt [CMTypewriter-rm/8]  % strojopis pro display ukázky
\fontdef\small [!/9]                 % zmenšení pro abstrakt a záhlaví
\addcmd \small {\baselineskip11pt \rm \def\mathversion{normal}\setmath[//]}
\fontdef\fontsekce [!/12]            % pro nadpisy sekcí
\addcmd \fontsekce {\bf \let\it=\bi \def\mathversion{bold}\setmath[//]}
\fontdef\fonttitul [!-bf/14.4]       % titul
```

Nyní i v abstraktech, poznámkách pod čarou a záhlavích je matematika ve všech velikostech všech možných indexů zmenšená odpovídajícím způsobem. Pokud by se matematika měla použít v názvu sekce, pak bude také tučná a správně veliká.

Reference

1. `www.pismolijna.cz`, `www.cstug.cz/stormtype`.
2. `ftp://math.feld.cvut.cz/pub/olsak/ofs`.
3. Petr Olšák. *OFS: Olšákův fontový systém*. 2001. Dokumentace k balíku je v souborech `ofsdoc.tex`, `ofsdoc.pdf`.
4. Petr Olšák. *Jak TeX k fontům ze Střešovic přišel*. Zpravodaj Československého sdružení uživatelů TeXu, 4/2001, pp 153–180.