When you're typing a manuscript for TEX, you need to know what symbols are available. The plain TEX format is based on the Computer Modern fonts, which provide the characters needed to typeset a wide variety of documents.
Donald Ervin Knuth: *The TEXbook.*

Kapitola pátá: Jak TEX pracuje s fonty

Petr Olšák

K tomu, abychom mohli v dalších kapitolách mluvit o národních zvláštnostech v instalacích TEXu a o zařazování obrázků do dokumentu psaného v TEXu, je třeba se aspoň stručně zmínit o tom, jak je to s fonty.

Pod pojmem „font" budeme považovat soubor jistých údajů, které se vztahují k písmenům jedné abecedy jednoho konkrétního vzhledu. Kromě písmen abecedy se do fontu zahrnují i další symboly s abecedou související (například číslice, interpunkce apod.). Navíc v TEXu existují speciální fonty, které žádnou abecedu neobsahují, pouze tvoří skupinu matematických symbolů. Údaje uložené ve fontu můžeme rozdělit následujícím způsobem:

- údaje o tvaru jednotlivých symbolů
- údaje o rozměrech symbolů
- údaje o mezerách mezi slovy
- další metrické údaje o vzájemných polohách objektů z fontu (např. v matematické sazbě informace o polohách indexů apod.)
- další údaje vztahující se k dané skupině symbolů (např. údaje o ligaturách apod.)

Skupina údajů, uvedená v prvním bodě není záležitostí TEXu samotného. Problematikou toho, jak symboly konkrétně vypadají, se zabývají až další programy, které zpracovávají výstup TEXu (soubor `.dvi`). O tom jsme se již zmínili v předchozí kapitole. Údaje o každém fontu jsou tedy rozděleny do dvou částí. První z nich (tak zvané metriky fontu) jsou uloženy ve zvláštním souboru s příponou `.tfm`, který TEX načítá v případě, že s daným fontem pracuje. Tam jsou uloženy informace, které jsme zde uvedli ve druhém až pátém bodu.

Údaje z prvního bodu jsou uloženy v souborech, obsahující bitové mapy tvarů písmen (většinou soubory `.pk`). S těmito soubory samotný TEX vůbec nepracuje.

Ve formátu k TEXu, jakým je například plainTEX nebo LATEX, jsou již metrické údaje některých fontů jednou provždy načteny a je definováno několik uživatelských příkazů, kterými lze sazbu v těchto fontech přepínat. S takovými příkazy jsme se už setkali v předchozích kapitolách. Například příkaz `\bf` přepne do polotučného fontu, nebo `\it` způsobí tisk kurzívou.

Chceme-li použít nějaký nestandardní font, který není definován ve formátu, lze použít primitivní příkaz `\font`. Ukážeme si to na příkladu. Berte tento příklad jako experimentální text; v žádném případě ne jako návod na to, jak by měly vypadat tištěné dokumenty.

```
\font\tucny=cmssbx10
\font\vysoky=cmdunh10

Tady píšu standardním románským písmem,
\it      zde jsem použil kurzívu, definovanou ve formátu (příkaz it),
\vysoky  nyní píšu vysokým písmem, Které Vynikne U Písmen Velké Abecedy,
\tucny   teď používám polotučné písmo bez patek
\rm      a nakonec zase standardní románské písmo.
\bye
```

Výsledek po zpracování dopadne takto:

Tady píšu standardním románským písmem, *zde jsem použil kurzívu, definovanou ve formátu (příkaz it),* nyní píšu vysokým písmem, Které Vynikne U Písmen Velké Abecedy, **teď používám polotučné písmo bez patek** a nakonec zase standardní románské písmo.

Příkaz `\font\tucny=cmssbx10` načte údaje o fontu, uloženém v souboru `cmssbx10.tfm` a vytvoří nový přepínač pro tento font s názvem `\tucny`. Tento přepínač se chová stejně, jako známé přepínače `\it`, `\bf` apod. Je třeba ale upozornit na to, že tyto přepínače jsou ve formátech definovány poněkud komplikovaněji, protože se od nich očekává obecnější využití (např. v matematické sazbě, nebo pro užití v různých velikostech).

Z uvedeného příkladu vyplývá, že k tomu, abychom mohli používat písma různých vzhledů, stačí znát názvy a významy všech souborů s příponou `.tfm`. Tyto názvy jsou pro začátečníka velmi těžko dešifrovatelné, protože se většinou jedná o zkratku pojmů, používaných v odborné terminologii v oblasti sazby. Proto většina uživatelů tento způsob zavádění fontů nepoužívá a vystačí si s příkazy, které jsou definovány ve formátu. Pro zajímavost se pokusíme „dešifrovat“ název souboru `cmssbx10`.

- `cm` znamená, že jde o písmo rodiny Computer Modern, jehož autorem je Donald Knuth
- `ss` je zkratka z francouzského „Sans Serif“, což je v sazbě běžně používaný pojem pro písmo bez patek
- `bx` je zkratka „Bold eXtended“, tj. polotučné písmo s „rozšířeným“ horizontálním rozměrem
- konečně desítka znamená, že se jedná o desetibodové písmo.

Nyní si něco povíme o velikostech fontů. Nechci zabíhat do zbytečných detailů z oblasti sazby, ale přesto bych rád upozornil na některé pojmy. Potřebujeme je znát například při instalaci TEXu na disk s omezenou kapacitou (každý disk má omezenou kapacitu). Fonty totiž v instalaci zabírají nejvíce diskové paměti. Zvažujeme-li, co instalovat a co nikoli, je třeba tomu alespoň trochu rozumět.

První věcí je otázka, co to znamená desetibodové písmo. Tiskařský bod je pevně stanovená měrná jednotka, která je velká zhruba třetinu milimetru (přesněji 72.27 bodů je rovno jednomu palci). V dobách rozvoje knihtisku totiž ještě nebyl zaveden jednotný měrný systém (SI), nicméně tiskaři potřebovali, aby jejich sady písem byly nějak kompatibilní. Proto si zavedli svou měrnou jednotku, která se používá dodnes. Tato jednotka sice není na celém světě jednotná, ale to nás začne zajímat až tehdy, když se začneme elektronické sazbě věnovat profesionálně. Dále na desetibodovém písmu nehledejte nic, co by mělo přesně rozměr deseti bodů. Je to zhruba šířka velkého písmene M, ale přesněji se jedná o velikost kuželky, ale ani to není vždy pravidlem. Co to je kuželka? Nebudu odpovídat, to bych zabíhal do zbytečných podrobností.

Nejčastěji používaný font v TEXu se jmenuje `cmr10`, tedy Computer Modern font Romana ve velikosti 10 bodů. Na něm si ukážeme problematiku rozměrů fontů. Nejdříve si řekneme, jak lze toto písmo zvětšit v libovolném poměru. Pomocí klíčového slova `scaled` uvedeného za příkazem `\font` lze udat libovolný poměr zvětšení. Číslo 1000 znamená poměr jedna ku jedné, číslo 800 například zmenšení na 8 desetin a třeba číslo 3500 je tři a půl násobné zvětšení. Podívejme se na tento příklad:

```
\font\veliky=cmr10 scaled 3500

Toto je {\veliky velmi} velké písmo.
\bye
```

Na výstupu dostaneme:

Toto je velmi velké písmo.

Pokud tento příklad sami vyzkoušíte na své instalaci TEXu, nebude Vám to pravděpodobně fungovat. Program TEX samotný se ještě nebude na nic zlobit, takže „překlad“ proběhne v pořádku. Program totiž vezme údaje pro font `\veliky` z naprosto stejného souboru `cmr10.tfm`, jako při definici standardního fontu. Není pro něj problém pronásobit všechny metrické údaje z tohoto souboru požadovaným koeficientem.

Problémy nastanou v okamžiku, kdy vyvoláte prohlížeč, případně tiskový program. Tyto programy už pracují s bitovými mapami fontů. Například Mattesovy programy Vám oznámí, že

```
Warning 1205: font cmr10 (...\cmr10<1050>) not found
New file name: ␣
```

Znamená to, že tento program nenašel bitovou mapu fontu v příslušném zvětšení 1050. Toto číslo je již závislé nejen na požadovaném zvětšení, ale i na požadované rozlišovací schopnosti výstupního zařízení. Zde jsme použili tiskárnu s rozlišením 300dpi (dot per inch) a vyžadujeme 3.5 násobné zvětšení. Proto program hledá bitovou mapu s „rozlišením“ $3.5 \times 300 = 1050$. Taková bitová mapa ale většinou není v systému instalovaná. Proto se program ptá na jiné jméno souboru, kde by bitovou mapu našel. Většinou je třeba takovou konverzaci s tiskovým ovladačem násilně ukončit a bitovou mapu nejdříve připravit pomocí programu METAFONT. O vlastnostech a způsobu použití programu METAFONT si povíme v některé z příštích kapitol.

Nicméně zvětšování písma je při přípravě dokumentu dosti častá operace (například pro nadpisy) a není možno nechat na uživateli, aby se setkával s výše uvedeným chybovým hlášením a musel pak zápolit s programem METAFONT. Většina uživatelů TEXu o existenci programu METAFONT neví skoro nic, případně jen „něco z doslechu“. Formáty, jako je například LATEX, přitom při sestavování nadpisů se zvětšenými fonty pracují. Proto bylo dohodnuto, že jistá předem stanovená zvětšení fontů se budou vždy ve standardních instalacích TEXu vyskytovat.

O jaká zvětšení se jedná? Každý další stupeň zvětšení je o pětinu větší, než ten předchozí, tedy je pronásoben koeficientem 1.2 . Tento koeficient nespadl z nebe, ale odpovídá jistým zvyklostem v typografii.

Na příkladu fontů pro laserovou tiskárnu s rozlišením 300dpi si uvedeme, která rozlišení se tedy v instalaci vyskytují. Kromě základního zvětšení jedna ku jedné (300) je tam zvětšení 1.2 (360), dále zvětšení $1.2 \times 1.2 = 1.44$ (432). Vynásobíme-li toto zvětšení znova koeficientem 1.2, dojdeme postupně k číslům (518), (622), (746). Podíváme-li se do instalace systému TeX, pak v případě, že jsou použity pro bitové mapy soubory formátu `pk`, najdeme tam určitě adresáře, v jejichž názvech se objevují uvedená čísla. Struktura adresářů může vypadat třeba takto:

`\tex\hpfonts\dpi300\cmr10.pk` … *bitová mapa fontu cmr10 v základním zvětšení* 1:1
`\dpi360\cmr10.pk` … *bitová mapa fontu cmr10 ve zvětšení* 1:1.2
`\dpi432\cmr10.pk` … *bitová mapa fontu cmr10 ve zvětšení* 1:1.44
`\dpi518\cmr10.pk` … *bitová mapa fontu cmr10 ve zvětšení* 1:1.728
`\dpi622\cmr10.pk` … *bitová mapa fontu cmr10 ve zvětšení* 1:2.076
`\dpi746\cmr10.pk` … *bitová mapa fontu cmr10 ve zvětšení* 1:2.488
`\dpi329\cmr10.pk` … *bitová mapa fontu cmr10 ve zvětšení* 1:$\sqrt{1.2}$

Větší zvětšení se v instalacích zpravidla nevyskytuje. V posledním řádku je uvedeno „poloviční“ zvětšení, které se při sazbě často používá. Uživatel TeXu přitom těmto číslům vůbec nemusí rozumět a většinou se o ně vůbec nestará. Napíše-li například v LaTeXu příkaz pro nadpis kapitoly:

```
\chapter{Výsledky měření}
...
```

LaTeX použije pro text „Výsledky měření“ vhodný font vhodně zvětšený tak, aby toto zvětšení odpovídalo některému z výše zmíněných čísel. Před tento text připojí (např. v jiném fontu) slovo „kapitola“ s odpovídajícím pořadovým číslem kapitoly. Navíc si text „Výsledky měření“ uloží do paměti a případně jej na dalších stránkách použije pro plovoucí záhlaví (zase k tomu zvolí odpovídající font). To ovšem záleží na tom, zda si uživatel plovoucí záhlaví přeje. Tuto skutečnost uživatel vyjádří příslušným příkazem LaTeXu.

Může se zdát, že uvedený způsob je jedinou metodou zvětšování a zmenšování písma. Není tomu tak! Mnozí z vás se například setkali se slovem „petit“ jako s pojmem pro drobné písmo, použité třeba pro poznámky pod čarou. Tento pojem v typografii znamená osmibodovou velikost písma. Kupodivu pro petit najdeme speciální soubor `cmr8.tfm`. Kdybychom pro petit zvolili font `cmr10` zmenšený na velikost 0.8, nedostaneme stejný výsledek, jako když použijeme přímo font `cmr8`. Osmibodový font totiž obsahuje jiné metrické informace a i tvary písmen jsou jiné. Když byl font navrhován, autor fontu už dopředu počítal, že bude použit pro sazbu petitu, tedy dosti drobného písma, a proto například pro lepší čitelnost navrhl mezi jednotlivými písmeny relativně větší mezery. Podívejte se na ilustraci tohoto problému. První řádek obsahuje font `cmr8`, tedy skutečný petit, zatímco druhý řádek je napsán fontem `cmr10` zmenšeným na velikost petitu.

pište v petitu
pište v petitu

pište v petitu
pište v petitu

Vlevo máte oba řádky navíc hodně zvětšeny, abyste mohli postřehnout rozdíly. Všimněte si například toho, že nejslabší tahy v písmu jsou ve skutečném petitu relativně silnější a mezery jsou větší. Síla tahů je asi patrnější v malé verzi, kde dochází při digitalizaci na 300dpi k celkem značnému zaokrouhlení.

Z uvedeného příkladu plyne, proč pro různé velikosti písma máme zcela odlišné fonty. V instalaci TeXu například najdete fonty `cmr*`, kde `*` postupně nabývá hodnot 5, 6, 7, 8, 9, 10, 12 a 17. Bitové mapy většiny těchto fontů najdete také skoro ve všech adresářích, které odpovídají jednotlivým zvětšením (300, 360, atd.). Důvod je prostý. Co kdyby náhodou uživatel použil v nadpise matematický index? Například při základní velikosti deset bodů je velikost indexu první úrovně rovna sedmi bodům a druhá úroveň má 5 bodů (o to vše se nemusíme starat; je to definováno ve formátu). Když nyní uživatel použije index v nadpise, který je zvětšený 1.44 krát, pak systém musí najít například soubor `cmr7.pk` v adresáři `dpi432`.

Máme zde vlastně jakési křížové zvětšení a zmenšení, které bohužel nešetří diskovou pamětí, ale chceme-li mít kvalitně vysázený dokument, nedá se nic dělat. Přitom jsme mluvili zatím pouze o fontu `cmr`. Mimo to existují i jiné fonty (polotučné, skloněné, vše s patkami, něco i bez patek, skloněné a zároveň tučné apod.). Myslím si, že pak nikoho nepřekvapí, že souborů s fonty může být v instalaci TeXu i několik tisíc. Zde záleží hodně na tom, co vše chcete používat ve své instalaci.

Nechci sice čtenáře unavovat dalšími technickými detaily, ale přesto bych se chtěl zmínit ještě o jedné věci, ve které je TeX charakteristický. Jedná se o tzv. ligatury neboli „slitky“. V jednom časopise jsem zahlédl slovníček pojmů z oblasti DTP a mohl jsem se tam dočíst, že ligatura je spojení dvou nebo více písmen do jednoho tvarového celku, který je sázen, jakoby šlo o jediné písmeno, například „fi“. Úsměvné na tom bylo, že se jim ten příklad moc nevyvedl. Článek byl totiž vysázen systémem, pro který je ligatura technicky nedosažitelná.

Podívejme se nyní na to, co dělá s ligaturami TeX. Ten najde v souboru `.tfm` tabulku ligatur. Jedná se o dvojice (nebo i trojice apod.) symbolů, pro které má definován v tabulce ligatur speciální odkaz na místo ve fontu, kde se ligatura (slitek) nalézá. Je přitom uložena ve fontu stejně, jako by se jednalo o samostatné písmeno. Vše vysvětlí následující příklad.

Uvažujme, že TeX zpracovává slovo `filosofie` ze vstupního souboru. Narazí na písmeno `f`. Pohlédne do tabulky ligatur a najde tam dvojice `fi`, `fl`, `ff` a trojice `ffi`, `ffl`. Proto zatím písmeno `f` nesází, ale podívá se, zda náhodou ve vstupu nenásleduje některé z písmen `i`, `l` nebo `f`. Je to pravda. TeX tedy zjistil, že je třeba vysázet ligaturu `fi` a podívá se do tabulky ligatur na kód, kde se ve fontu ligatura nalézá. Vysadí potom symbol s tímto kódem, jako by se jednalo o jediný symbol. V tomto případě jde o symbol „fi“. Pak TeX přečte ze vstupu písmeno `l`. Toto písmeno se nikde v tabulce ligatur nevyskytuje na prvním místě a proto jej TeX okamžitě vysází jako symbol s kódem, který je roven ASCII kódu písmene `l`. Tento algoritmus je v činnosti při sazbě každého písmene. Že to má svůj význam vidíte vlevo na zvětšené ukázce. V prvním řádku jsou pro dvojice `fi` použity ligatury, zatímco ve druhém řádku je tento algoritmus TeXu potlačen. Je vidět, že tvar písmene `f` je ve slitku navržen poněkud jinak, než při výskytu téhož písmene samostatně.

filosofie
filosofie

Nejčastěji používané ligatury jsou: fi, fl, ff, ffi, ffl. Poslední dvě se vyskytují pouze v cizích jazycích, např. ve slově „difficult“. Nejčastější slovo s ligaturou v časopisech se softwarovou tematikou je slovo „firma“ a „grafika“. Všimněte si, že tato slova jsou v časopisech sázena různě, ale nikdy ne s ligaturou. Výjimku tvoří časopis Bajt, který je sázen v TeXu. Zajímavý příklad najdete třeba v lednovém čísle Softwarových novin na straně 18 v článku „SWS v Bratislavě“. Zde v prostředním sloupci v třetím řádku odspodu je tento text:

… naší firmy, ale všech, neboť i další firmy využí-…

Zde je zajímavý úkaz. V prvním výskytu slova „firma“ je písmeno `i` přisunuto blíže k písmenu `f`, zatímco ve druhém výskytu nikoli. To vše na jediném řádku. Je možné, že sazeč si v jednom výskytu slova „firma“ všiml přítomnosti dvojice písmen `f` a `i` a chtěl proto „simulovat“ ligaturu přisunutím uvedených dvou písmen k sobě, zatímco na druhé slovo „firma“ zapomněl. Pravděpodobnější ale je, že systém zaokrouhluje rozměrové údaje tak katastrofálně, že to je dokonce vidět. To by si TeX nikdy nedovolil. Připomeňme si, že rozměrové údaje zpracovávané TeXem jsou reprezentovány celými čísly s jednotkou, která odpovídá velikosti menší než je vlnová délka světla. Jedná se o velikost 1/65536 tiskařského bodu. Zaokrouhlovací chyby TeXu jsou proto vlastně nezměřitelné už z fyzikálního principu.

Nezlobte se na mě, že stále kritizuji vzhled tohoto časopisu. Ostatní časopisy vypadají stejně, nebo ještě hůř. Příklady z tohoto časopisu uvádím pouze proto, že předpokládám, že jej má čtenář po ruce. Stále nedovedu pochopit, proč profesionálové, zabývající se elektronickou sazbou, váhají mezi nákupem Pagemakeru a Ventury, přičemž ani na vteřinu nevezmou v úvahu, že existuje TeX. Taky mě mrzí, když se někdo pokouší srovnávat TeX s textovými procesory. Řeči o tom, že něco lze napsat v ChiWriteru, Word Perfectu nebo v T602, takže TeX není potřeba, dokazují naprosté nepochopení věci. Čtenář tohoto seriálu už určitě ví, že srovnávat simulátory psacích strojů se systémem pro dokonalou elektronickou sazbu je absurdní nesmysl.

Kromě ligatur TeX načítá ze souboru `.tfm` „tabulku výjimek“. Za normálních okolností totiž klade TeX symboly za sebou (nebo nad sebe) podle jakýchsi „boxů“. Rozměry těchto boxů jsou definovány pro každý symbol v souboru `.tfm`. Nicméně některé dvojice písmen vyžadují porušení tohoto pravidla. Potřebují přisadit blíž k sobě, nebo naopak odsunout dál od sebe. Tento požadavek TeX provede podle zmíněné tabulky výjimek.

Na závěr si ukážeme jeden zajímavý příklad použití ligatur. Jedná se o sazbu azbuky. Vyvoláme font `wncyr10` a napíšeme něco rusky:

```
\font\bukvy=wncyr10
\catcode`\'=13    \def'{\char126 }

\bukvy Ya reshil podgotovit' nashu knigu v sisteme {\rm\TeX}.
\bye
```

Ve fontu `wncyr` je dvojice `Ya` označena jako ligatura tvaru „ Я“ a pro dvojici `sh` je požita ligatura „ш“. Dále jsme označili vstupní symbol `'` jako aktivní příkaz (pomocí příkazu ``\catcode`\'=13``) a jeho význam jsme definovali jako `\char126`, což je kód měkkého znaku ve fontu. Proto na výstupu dostaneme:

Я решил подготовить нашу книгу в системе TeX.

Ve fontu je uložena spousta dalších ligatur obdobným způsobem. Ruštinu potom lze psát podle mezinárodní transkripce azbuky. Tento způsob využijeme pro krátké citace ruských titulů. Příprava delších textů vyžaduje buď pevné nervy, nebo cvik. Těžko si ale představíme, že by byl ochoten takto připravovat dokument rodilý Rus. Proto je třeba řešit problém jinak. O tom si povíme příště v kapitole věnované modifikacím TeXu na různá národní prostředí.