

Uwagi nt. wykorzystania fontów truetype

Tomasz Przechlewski

Już od jakiegoś czasu znana firma Microsoft Corp. udostępnia nieodpłatnie wiele fontów komputerowych (<http://www.microsoft.com/typography/>). Są one oczywiście zapisane w promowanym przez Microsoft formacie TrueType. Z fontów tych można korzystać posługując się T_EX-em. Od razu uprzedzam, że posługiwanie się fontami TrueType nie jest takie bezproblemowe jak w wypadku fontów w formacie typu pierwszego (PostScriptowych), jednakże w zamian mamy możliwość wykorzystania kilkudziesięciu darmowych fontów w ponad dziesięciu krojach. Sześć z tych krojów, a mianowicie: Times New Roman, Arial, Courier New, Verdana, Georgia i Trebuchet jest udostępnione w czterech podstawowych odmianach (normalna, kursywa, gruba, gruba kursywa). Pozostałe kroje (Andale Mono, Impact, Arial Black, Comic Sans) występują w 1–2 odmianach. Jest także jeden font symboliczny (Webdings).

1 Przygotowanie pliku .tfm

T_EX-owi jest dokładnie wszystko jedno czy składa za pomocą fontów .pk, .pfb, TrueType czy opentype. Jedyną rzeczą, którą T_EX potrzebuje jest informacja o wymiarach znaków, podcięciach itp. przechowywana w pliku .tfm. Pliki takie możemy otrzymać z plików .ttf wykorzystując, np. program `ttf2afm` dołączony do dystrybucji programu `pdftex` (najnowszą wersję znaleźć można w <http://www.cstug.cz/tex/local/cstug/thanh>). W wyniku prostego uruchomienia programu¹:

```
ttf2afm verdana.ttf > verdana.afm
```

zostanie utworzony plik AFM. Do tego samego celu możemy także wykorzystać program `ttftot42` (<ftp://ftp.giga.or.at/pub/nih/ttftot42/>, autor: Dieter Baron). Ponieważ program ten korzysta z biblioteki freetype (<http://www.freetype.org>) musimy ją uprzednio zainstalować. W systemie Linux skompilowanie oraz zainstalowanie biblioteki i programu nie stanowi problemu. W systemie „W” może być gorzej. Ponieważ program

¹ Uwaga: wstępne wersje programu `ttf2afm` zawierały błędy, ale wydaje się, że w wersji 0.12r `pdftex`-a większość została usunięta.

ten został udostępniony w chwili pisania tego tekstu, autorowi nie są znane adresy archiwów, z których można by pobrać wykonywalne wersje programu `ttftot42` na inne platformy systemowe. Można mieć pewność, że w chwili gdy ten tekst *będzie czytany*, takowe wersje będą już udostępnione – zainteresowani powinni rozpocząć poszukiwania od obu wzmiankowanych w tym akapicie adresów.

Jeżeli korzystamy z programu `ttftot42` to piszemy:

```
ttftot42 verdana.ttf
```

W rezultacie otrzymamy dwa pliki, jeden z rozszerzeniem .afm i drugi z rozszerzeniem .t42, który zawiera font tzw. typu 42, będący specjalnie „opakowanym” na potrzeby interpretatora PostScripta fontem TrueType. Pliki te będą potrzebne na etapie drukowania dokumentu, o czym dalej w tekście².

Otrzymany w rezultacie plik AFM konwertujemy do formatu TFM za pomocą programu `afm2tfm` z dystrybucji `dvips-a`. Znacznie lepiej do takiej konwersji nadaje się program `toil` autorstwa B. Jackowskiego i kolegów z f-my BOP. Posługując się `toil`-em możemy w szczególności w wygodny sposób zmienić wektor kodowania fontu, co jest nieuniknione jeżeli posługujemy się językiem polskim. Z tego powodu opiszę wyłącznie jak należy wygenerować (z otrzymanych fontów AFM) pliki TFM z wykorzystaniem pakietu `toil`.

Pakiet `toil` to kilka skryptów napisanych w języku AWK plus sterujące nimi pliki .bat i kilka plików „konfiguracyjnych”. Całość jest gotowa do zainstalowania w systemach DOS/Windows. Dla systemów Unix-owych zmodyfikowałem nieco oryginalne skrypty AWK-owe a skrypty .bat zamieniłem na skrypty napisane w shell-u.

Niestety fonty Microsoftu nie są wolne od niedoróbek, co nieco komplikuje sprawę. W szczególności w fontach z kroju Georgia nazwa polskich

² Jest jeszcze rodzimy program `ttf2pf` (<ftp://ftp.GUST.org.pl/TeX/GUST/BachoTeX98/>, autor: f-ma BOP). Jest to plik PostScriptowy i uruchamiający go DOS-owy plik .bat. Do konwersji wykorzystywany jest `ghostscript`. Program `ttf2pf` działa przeciętnie dużo gorzej niż `ttftot42`: często się „wysypuje” a pliki .pfa są dużo większe od tych wyprodukowanych przez `ttftot42`. Tym niemniej wspominamy o nim gdyż można mieć nadzieję, że podobnie jak sam `ghostscript`, kolejne wersje `ttf2pf` będą dużo lepsze.

znaków „ż” i „Ž” to „zdotaccent” i „Zdotaccent” (prawidłowo), podczas gdy w innych krojach (testowałem Times, Arial, Courier i Verdana) jest to „zdot/Zdot”. Dlatego generując plik .tfm dla fontu Georgia, powinniśmy wykorzystać gotowy skrypt:

```
a2t-qx-zacc georgia
```

natomiast w przypadku pozostałych fontów należy uruchomić skrypt a2t-qx-zdot, np.:

```
a2t-qx-zdot arial
```

Jeżeli chcemy wygenerować font z innym, zdefiniowanym przez nas, układem znaków to winniśmy skorzystać ze skryptu a2t. Skrypt należy wywołać z trzema parametrami, np:

```
a2t verdana e.qq l.qq
```

Powyższe oznacza pobranie wektora kodowania zawartego w pliku e.qq a schematu ligatur z pliku l.qq. Sposób deklarowania kodowania i ligatur jest w pakiecie toil, tak prosty, że pozwolimy sobie pominąć go w tym tekście. Osoby zainteresowane nie powinny mieć z tym żadnych problemów – wystarczy zajrzeć do plików dostarczonych w pakiecie, np. e.qx czy l.qx.

Drugą z zauważonych przeze mnie niedoróbek darmowych fontów Microsoftu jest brak we wszystkich fontach, za wyjątkiem krojów Times i Arial, definicji parametru „HXheight”, który służy później do określenia jednostki odległości ex w T_EX-u. Zarówno program ttft2afm jak i ttftot42 generuje plik AFM pozbawione tej ważnej informacji³.

Pakiet toil okazał się nieodporny na taką niedoróbkę. Nawiasem mówiąc w specyfikacji formatu truetype znajduje się rekomendacja, że „HXheight” powinno być zdefiniowane jako ele-

ment tablicy PCLT. Tablica jest opcjonalna ale cyt. „highly recommended”. Jak widać firmie nie udało się zastosować do własnych zaleceń.

W związku z tym zmodyfikowałem nieco działanie toil-a: w przypadku gdy, w nagłówku pliku AFM nie ma określenia wartości XHeight, to jest ona wyznaczana jako wysokość litery „x” fontu. W tym przypadku na ekranie pojawia się komunikat⁴:

```
XHeight not found in the .afm preamble!
XHeight set equal the height of 'x'!
```

Po wygenerowaniu plików .tfm wstawiamy je do odpowiedniego katalogu, np. /texmf/fonts/tfm/public/ms/. Użytkownicy L^AT_EX-a potrzebują jeszcze plików .fd i .sty. Zestaw takich plików można znaleźć w pakiecie pod roboczą nazwą msf-lqx, znajdującego się w niezastąpionym archiwum ftp. GUST.org.pl/TeX/GUST/contrib/#####.

Przykładowo aby złożyć dokument w ten sposób, żeby tytuły i śródtytuły były składane krojem Arial, tekst główny krojem Times New Roman a imitacja pisma maszynowego krojem Courier New należy w preambule umieścić następujące polecenia:

```
\documentclass[twoside]{article}
...
\usepackage{sfheaders}
\usepackage{mstac}
\usepackage[LQX]{fontenc}
...
\begin{document}
```

Po kompilacji pliku źródłowego L^A/T_EX-em otrzymamy plik .dvi. Ale co dalej?

2 Drukowanie

Sterownik dvips potrafi zamienić plik .dvi na plik PostScriptowy, dołączając w razie potrzeby odpowiednie fonty w formatach .pk oraz .pfb lub .pfa.

⁴ Zmieniony toil informuje także o brakujących, w stosunku do zdanego układu, znakach. Zadając układ QX okaże się że w foncie Arial brakuje 8 znaków: ligatur fk, ff, ffi, ffl oraz znaków dotlessj, ell, anglearc i diameter. Znając awersję Microsoftu do ligatur należało tego oczekiwać. W foncie Trebuchet oprócz w/w brakuje jeszcze alpha, beta, delta, Pi, Sigma, Iogonek, Uogonek, iogonek oraz uogonek. Jest za to dotlessj!

³ Mniej dokuczliwe niedoróbki to różne wektory kodowania fontów (w obrębie jednego kroju). Np. font Times ma 652 znaki a pozostałe odmiany tego kroju o jeden znak (smiggring) mniej. Trebuchet (306/307 znaków) w odmianach normalnej i półgrubej nie ma zdefiniowanego znaku foursuperior a w pozostałych odmianach ma. Wektor kodowanie fontu Verdana (892 znaki) w odmianie normalnej, różni się od pozostałych odmian w kilkudziesięciu miejscach! Font ten zresztą jako jedyny w tym kroju ma zdefiniowane pary kernowe, pomimo tego, że oficjalnie pismo jest tak projektowane żeby nie było potrzeby definiowania podcięć! W przypadku pozostałych fontów kodowanie znaków w obrębie poszczególnych krojów jest jednakowe.

Innych formatów dvips nie zna dlatego w przypadku fontów truetype ta sprawdzona ścieżka odpada. W chwili obecnej mamy do dyspozycji co najmniej⁵ trzy inne możliwości:

- zamianę fontów truetype na fonty typu 42,
- drukowanie bezpośrednio z ghostscript-a,
- wykorzystanie programu pdftex.

Poniżej omówimy po kolei każdą z tych ścieżek wskazując na jej słabe i mocne strony.

2.1 Fonty typu 42

Jak wspomniano program `ttftot42` nie tylko generuje plik `.afm`, ale także plik z rozszerzeniem `.t42`. Jego zawartością jest font w formacie typu 42, który dvips jest w stanie dołączyć do pliku `.ps` przy zamianie z formatu DVI.

Mając do dyspozycji np. font `verdana.t42`, postępujemy tak jak w wypadku fontów PostScriptowych, tj. umieszczamy go w katalogu, w którym zostanie odnaleziony przez dvips-a, a do pliku `psfonts.map` dodajemy linijkę⁶:

```
verdana Verdana "enc-verdana \
ReEncodeFont" <verdana.enc <verdana.t42
```

W pakiecie `msf-lqx` znajduje się plik `MS0.map` zawierający konfigurację dla wszystkich fontów. Wystarczy go dołączyć do pliku `psfonts.map`, lub uruchamiać dvips-a z opcją `-u`, np.:

```
dvips -j0 -u MS0.map test.dvi
```

Opcja `-j0` oznacza, że nie możemy korzystać z częściowego ładowania fontów. W wersji 5.83 dvips-a włączenie opcji `-j1` nie przeszkadzało, z tym, że było przez dvips-a ignorowane.

Tak otrzymany plik `.ps` prawidłowo wydrukujemy się za pomocą ghostscript-a, ale niekoniecznie na *każdym* urządzeniu postscriptowym. Szczególnie posiadacze interpretatorów wbudowanych, za które kiedyś zapłacili ciężki pieniądz, mogą przeżyć bolesne rozczarowanie. Trzeba bowiem pamiętać

⁵ Piszę co najmniej, bo istnieją sterowniki komercyjne, które umożliwiają wykorzystanie fontów truetype. Być może też istnieje oprogramowanie freeware, którego nie znam.

⁶ Ukośna kreska na końcu pierwszego wiersza „/” jest znakiem kontynuacji i wskazuje, że całość powinna być zapisana w pliku `psfonts.map` w jednej linijce, co było niemożliwe w Biuletynie z uwagi na zbyt wąską szpalcę.

tać, że typ 42 to nowe rozszerzenie wprowadzone przez Adobe i wiele interpretatorów starszych typów nie potrafi ich poprawnie wydrukować.

2.2 Ghostscript

Jeszcze mniej przenośne od poprzedniego, ale prostsze rozwiązanie problemu wydruku można osiągnąć odpowiednio konfigurując plik `Fontmap` ghostscript-a. W przypadku wykorzystywania tego podejścia, plik konfiguracyjny `psfonts.map` powinien zawierać (dla fontu Verdana) linijkę:

```
verdana Verdana "enc-verdana \
ReEncodeFont" <verdana.enc
```

W rezultacie otrzymamy plik `.ps` *bez dołączonego* fontu. W pakiecie `msf-lqx` znajduje się plik `MS1.map` zawierający odpowiednie wpisy konfiguracyjne dla wszystkich fontów. Wystarczy jego zawartość dołączyć do pliku `psfonts.map`, lub uruchamiać dvips-a z opcją `-u`, np.:

```
dvips -u MS1.map test.dvi
```

Teraz musimy dokonać odpowiednich zapisów w pliku `Fontmap`, np. dla fontu Verdana należy wpisać:

```
/Verdana (/fonts/truetype/verdana.ttf);
```

Ścieżka `/fonts/truetype/` jest oczywiście przykładowa. Jeżeli wstawimy font do katalogu, który jest przeszukowany przez ghostscript-a, to oczywiście nie musimy podawać ścieżki.

Teraz możliwe jest oglądanie/drukowanie dokumentu za pomocą ghostscript-a (lub dowolnej nakładki typu `gsview` czy `gv`).

2.3 Pdftex

Program `pdftex` był już opisywany w numerze 9 Biuletynu GUST. Od tego czasu jego możliwości zwiększyły się, a jedną z tych nowych funkcji jest umiejętność korzystania z fontów truetype.

Wykorzystanie fontów truetype jest możliwe tylko podczas pracy w trybie rozszerzonym, w którym `m.in.` zamiast pliku `.dvi` tworzony jest plik w formacie PDF. Plik `.pdf` możemy następnie obejrzeć lub wydrukować w programie rozumiejącym ten format, np. Adobe Acrobat lub ghostscript.

W tym celu należy tylko zmodyfikować plik `pdftex.cfg`, dodając linijkę

```
+map msttf.map
```

Plik `msttf.map` wchodzi w skład pakietu `msf-lqx` i zawiera niezbędne zapisy konfiguracyjne. Przykładowo dla pliku Verdana, zawiera on linijkę:

```
verdana Verdana "enc-verdana \
  ReEncodeFont" <verdana.ttf <verdana.enc
```

Jak widać stosowana jest notacja prawie, że identyczna do tej z pliku `psfonts.map`.

Testy przeprowadzone dla wersji 0.12h programu `pdftex` wypadły pomyślnie, tj. dokumenty były wyświetlane i drukowane poprawnie. Należy jednak pamiętać, że wersje 0.12 są oznaczone jako beta co oznacza, że mogą wystąpić problemy. I rzeczywiście w wersji 0.12r autor zapewne poprawiając coś tam innego popsuł obsługę fontów `truetype`. Należy zatem zachować czujność.

◇ Tomasz Przechlewski
ekotp@univ.gda.pl