Pakiet TOIL czyli oswajanie fontów POSTSCRIPT-owych

Bogusław Jackowski

Na spotkaniu w Bachotku wiosną 1997 roku został udostępniony pakiet TOIL przeznaczony do instalowania fontów POSTSCRIPT-owych dla potrzeb składu TEX-owego.

Główna idea pakietu, wykorzystującego program AWK oraz METAFONT, polega na tym, żeby z pliku AFM utworzyć plik źródłowy dla METAFONT-a, który zawierałby informację metryczną, czyli rozmiary znaków, pary kernowe, ligatury oraz korekty kursywy (kształty znaków są na tym etapie nieistotne). Po przetworzeniu pliku wynikowego METAFONT-em uzyskujemy plik *.tfm, niezbędny do składania dokumentów TEX-em. Oprócz pliku *.tfm tworzone są również dodatkowe pliki dla sterownika DVIPS, mianowicie psfonts.map, oraz plik zawierający informację o aktualnym układzie znaków, tzw. plik kodowania (*encoding file*).

Podobne funkcje pełni program AFM2TFM, należący do pakietu dystrybucyjnego sterownika DVIPS. Program ten umie nawet więcej, bo obsługuje – w odróżnieniu od TOIL-a – fonty wirtualne (por. artykuł Alana Hoeniga w niniejszym numerze). Po co więc tworzyć nowe, uboższe narzędzie? Dwa grzyby w barszcz?

Tak bardzo uboższy od programu AFM2TFM to pakiet TOIL nie jest – ma przede wszystkim spore możliwości konfigurowania, i to jest jego mocny punkt. A poza tym z fontów wirtualnych, jak pokazuje praktyka, korzystają głównie wirtuozi...

Plik konfiguracyjny pakietu TOIL określa, gdzie i jakie pliki wejściowe mają być poszukiwane oraz gdzie i jakie pliki wynikowe mają się znaleźć. Ponadto pozwala na wygodne – zdaniem autorów – definiowanie własnych układów znaków i własnych zestawów ligatur. Jeśli uwzględnić jeszcze, że użytkownik ma do dyspozycji różne „gwizdki i dzwonki”, na przykład możliwość zignorowania pojawiających się w niektórych fontach kernów między cyframi, to myślę, że czytelnik się zgodzi, iż dwa grzyby w barszcz to w tym wypadku niekoniecznie za dużo.

Aktualnie pakiet TOIL działa pod systemem MS-DOS, ale trwają próby przeniesienia pakietu na inne platformy, na których istnieją implementacje AWK-a i METAFONT-a (Linux, Windows NT, OS/2).

Po zainstalowaniu pakietu korzystanie zeń jest bardzo proste. Plik wsadowy o nazwie a2t.bat robi co trzeba, to znaczy tworzy plik *.tfm, psfonts.map oraz jeśli trzeba plik kodowania *.enc. Polecenie systemowe ma następującą postać:

A2T ⟨*nazwa fontu*⟩ [⟨*parametry dodatkowe*⟩]

Parametr ⟨*nazwa fontu*⟩ oznacza tu dyskową nazwę (w odróżnieniu od POSTSCRIPT-owej). Podkreślić należy, że to, gdzie szukać danego fontu, zapisane jest w pliku konfiguracyjnym toil.cfg pakietu TOIL i że font może być mozolnie (po angielsku „toil” to tyle co „mozół”) poszukiwany w rozmaitych podejrzanych o zawieranie go kartotekach. Może wydawać się dziwne, że ktoś może nie wiedzieć, gdzie jaki font się znajduje, ale załóżmy, że mamy fonty spolonizowane przez firmę ThETA (kilkaset fontów na trzech płytach CD) plus fonty TEX-owe (CM, PL, EC, Euler), fonty firmy URW dystrybuowane wraz z Ghostscriptem, fonty dostarczane z niektórymi sterownikami, itp., itd., i załóżmy, że w ramach unikania bałaganu przygotowujemy specjalną strukturę kartotek i tam starannie w osobnych kartotekach układamy poszczególne fonty... Łatwo sobie wyobrazić, że już po paru miesiącach przestanie być oczywiste, gdzie którego fontu poszukiwać. Ma więc sens zapisanie kartotek fontowych raz na zawsze (no, może raz na jakiś czas) w pliku konfiguracyjnym.

Opcjonalne ⟨*parametry dodatkowe*⟩ mają postać:

-L[ ]⟨*nazwa pliku definiującego ligatury*⟩
-E[ ]⟨*nazwa pliku definiującego kodowanie*⟩
-M[ ]⟨*nazwa (tymczasowego) pliku* METAFONT-*owego*⟩
-N[ ]⟨*nazwa pliku kodowania dla sterownika* DVIPS⟩
-X[ ]⟨*przeskalowanie poziome fontu*⟩
-I[ ]⟨*dodatkowe pochylenie fontu*⟩

(Spacja w nawiasach kwadratowych oznacza opcjonalny odstęp, duże i małe litery są utożsamiane.)

Istnieje jeszcze parametr -G o wartości 0 lub 1, określający czy utworzyć lokalnie nowy plik psfonts.map (wartość 0) czy też skorzystać z istniejącego już, globalnego (stąd G) pliku psfonts.map (wartość 1), wskazywanego przez zmienną systemową TEXCONFIG.

Jeżeli chodzi o polskie znaki diakrytyczne, to trudno się spodziewać, żeby w ogólności w fontach

PostScript-owych miały one przypisane kody zgodne z wymaganiami pakietów TEX-owych, czyli układ zgodny z układem fontów PL czy EC. Raczej będą to układy znaków z punktu widzenia TEX-a egzotyczne, takie jak np. CP-1250.

Aby pożenić ogień z wodą, czyli TEX-a ze światem windowsowym, po licznych, trwających ładnych parę lat debatach, wypracowany został tzw. „układ QX” określający zestaw znaków oraz kody znaków dla potrzeb zastosowań windowsowych i TEX-owych. Oczywiście kody polskich znaków w układzie QX do zastosowań TEX-owych są zgodne z układem fontów PL, czyli są przystosowane do wykorzystania od ręki w MEX-u. Szczegółowe porównanie zestawów znaków w różnych układach zostało zamieszczone na stronie 28.

Wraz z pakietem TOIL dystrybuowany jest plik E.QX zawierający informację niezbędną dla wygenerowania pliku *.tfm z polskimi znakami w układzie QX. W pewnym sensie uzupełnieniem pliku E.QX jest plik L.QX zawierający definicje stosownych ligatur dla układu QX. Aby zainstalować font zawierający polskie znaki diakrytyczne, np. udostępniony również na spotkaniu w Bachotku wiosną 1997 roku font PostScript-owy Quasi-Palladio, należałoby użyć polecenia systemowego:

A2T ⟨*nazwa fontu*⟩ -L L.QX -E E.QX

(W istocie Quasi-Palladio jest dystrybuowany wraz ze wszystkimi „dodatkami” niezbędnymi dla wykorzystania tego fontu w składzie TEX-owym.)

Ze względu na domniemanie, iż powyższe polecenie będzie dość często wykorzystywane, pakiet TOIL zawiera plik wsadowy a2tqx.bat, realizujący powyższe polecenie.

Z okazji Nowego Roku wielu oswojonych fontów w imieniu autorów pakietu życzy Czytelnikom autor artykułu.

Podziękowania

Jestem wdzięczny Staszkowi Wawrykiewiczowi za uporczywe zachęcanie mnie i innych do prezentowania poglądów w formie pisanej.

◇ Bogusław Jackowski
B.Jackowski@GUST.ORG.PL

Z kolekcji Keesa van der Laana

```
%!PS-Adobe- More Square madness,
% cgl Feb 97
%%BoundingBox: -80 -80 80 80
/r 50 def
/s .5858 r mul def /ms s neg def
/sds .707 s mul def /msds sds neg def
/element{gsave r 0 translate
sds 0 moveto 0 sds lineto
msds 0 lineto 0 msds lineto
closepath fill
grestore
}def
/f r sds sub r sds add div def
/gl 1 def
%
3{8{element 45 rotate}repeat
  f f scale /gl f gl mul def
  gl setgray
 }repeat %showpage
```