O lepszą jakość PDF-ów

Grzegorz Sapijaszko
ul. Trzebnicka 6 m. 5, 50–246 Wrocław
grzegorz@sapijaszko.net

Pracę zgłosił: Tomasz Przechlewski

Streszczenie

Jakość wynikowego pliku PDF (szczególnie na ekranie monitora) w dużej mierze zależy od użytych w nim fontów. Jednocześnie nic nie stoi na przeszkodzie, aby zamiast fontów bitmapowych używać ich wektorowe odpowiedniki. I do tego ma zachęcić poniższy artykuł.

Wstęp

Do napisania tego artykułu skłoniły mnie pojawiające się dość często na obu listach dyskusyjnych pytania dotyczące wyglądu plików PDF tworzonych przy pomocy TeX-a. Na ogół pytania te dotyczą fontów, ich „brzydkiego i postrzępionego” wyglądu przy podglądzie pliku na ekranie monitora – przeto podjęto próbę przedstawienia większości aspektów związanych z konfiguracją poszczególnych programów, które najczęściej wykorzystywane są do tworzenia plików PDF.

Fonty w TeX-u

Tak naprawdę to TeX do poprawnej pracy nie potrzebuje żadnych fontów, a jedynie informację o nich zawartą w plikach metryk (`.tfm`). „Właściwe fonty” (bitmapowe, wektorowe) wykorzystywane są dopiero przy przetwarzaniu plików `DVI` na postać zrozumiałą dla człowieka (wydruk na drukarce, na monitorze). Trochę inaczej rzecz się ma w przypadku pdfTeX-a. Jako, że wynikiem programu pdfTeX jest bezpośrednio plik `PDF`, to w momencie kompilowania dokumentu do utworzenia poprawnego dokumentu `PDF` oprócz metryk potrzebne są jeszcze „fonty właściwe”. Więcej podstawowych informacji na temat fontów zawartych jest w artykule [1].

O tym, jakiego rodzaju font zostanie użyty przy dalszym przetwarzaniu decydują poszczególne programy, a w szczególności ich konfiguracja. Poniżej opisana zostanie konfiguracja programów pdfTeX, `dvipdfm` oraz `dvips`.

pdfTeX

Plik konfiguracyjny programu (`pdftex.cfg`) standardowo znajduje się w katalogu `/texmf/pdftex/config/`. Plik ten zawiera szereg opcji wpływających na zachowanie się pdfTeX-a w trakcie przetwarzania dokumentu, w szczególności zaś podane są pliki zawierające mapowanie fontów TeX-a na ich wektorowe odpowiedniki. Więcej informacji na temat konfiguracji pdfTeX-a w [2].

Część odpowiadająca za mapowanie może wyglądać następująco:

```
map psfonts.map
map +pl.map
```

Do pliku konfiguracyjnego można dodać kilka plików z mapowaniem, każdy z nich musi być podany w osobnej linii. Należy zwrócić uwagę na znak `+` stojący przed plikiem z mapami. Dzięki dodaniu tego znaku mapowania zawarte w pliku dodawane są do wczytanych wcześniej mapowań. Usunięcie znaku `+` z ostatniej linii, np. poniższy zapis:

```
map psfonts.map
map +pl.map
map jakis_plik.map
```

spowoduje, że pdfTeX będzie znał jedynie mapowania zawarte w pliku `jakis\_plik.map`. Przykładowa linia z pliku `pl.map` zawiera:

```
plr10  PLRoman10-Regular  "encplrm
   ReEncodeFont" <plrm.enc  <plr10.pfb
```

gdzie poszczególne pozycje oznaczają:

`plr10` – TEX-owa nazwa fontu,

`PLRoman10-Regular` – wewnętrzna nazwa fontu,

`"encplrm ReEncodeFont"` – przekodowanie fontu,

`<plrm.enc` – polecenie dołączenia pliku kodowania,

`<plr10.pfb` – polecenie dołączenia pliku z fontem.

przy czym pojedynczy znak < przed nazwą pliku z fontem poinstruuje program aby włączył do dokumentu PDF tylko te znaki z fontu, które zostały użyte (dzięki czemu wielkość dokumentu PDF jest mniejsza), zaś zapis `<<plr10.pfb` spowoduje wbudowanie całego zestawu znaków zawartych w foncie.

Instalowanie fontów Dla przykładu opisano zainstalowanie fontu tak, aby widziany był on przez pdfTEX-a. Pisząc ten artykuł na konferencję początkowo użyta została klasa `gust`, który wykorzystuje fonty Computer Concrete. Jednocześnie używając pakietu `polski` i kodowania OT4 można skorzystać ze spolszczonej wersji fontów. Niestety, w używanej dystrybucji zainstalowane były jedynie metryki fontów, nie było natomiast fontów wektorowych. Aby je zainstalować należało skopiować je ze stron GUST-u, a następnie rozpakować zgodnie ze strukturą katalogów: metryki fontów do katalogu `/fonts/tfm/public/cc-pl/`, pliki obwiedniowe `.pfb` do katalogu `/fonts/type1/public/cc-pl` zaś pliki `.map` oraz `.enc` do katalogu `/dvips`. Następnie należy odświerzyć bazy danych TEX-a. Aby fonty wektorowe były widziane przez program (pdfTEX) należy go skonfigurować – całe zagadnienie sprowadza się do dodania pliku z mapowaniem do pliku konfiguracyjnego. W pliku tym wystarczy dopisać:

```
map +pcr.map
```

i przekompilować dokument.

W przypadku braku fontów wektorowych należy zagwarantować, aby oprócz metryk dostępne były metafontowe źródła – na ich podstawie pdfTEX przy pomocy programu METAFONT wygeneruje fonty bitmapowe, które wbuduje do pliku PDF. O ile plik ten dobrze będzie wyglądał na wydruku, o tyle źle (lub bardzo źle) będzie wyglądał na ekranie monitora. O jakości generowanych fontów bitmapowych, a dokładnie o ich rozdzielczości decyduje kolejny parametr występujący w pliku konfiguracyjnym:

```
pk_resolution 600
```

W tym konkretnym przypadku parametr ten informuje on Metafont-a, iż generowane mają być fonty o rozdzielczości 600 punktów na cal.

dvipdfm

Podobnie jak pdfTEX `dvipdfm` również wykorzystuje plik konfiguracyjny, na ogół jest to plik `config` umieszczony w katalogu `/texmf/dvipdfm/config`. Po skompilowaniu dokumentu `.tex` i otrzymaniu pliku `.dvi` należy użyć `dvipdfm` wywołując go z linii poleceń, zaś opcje podać jako parametry wywołania, np.:

```
dvipdfm -p a4 -f pl.map plik.dvi
```

Za fonty odpowiada opcja `-f`

```
% Font map file
f psfonts.map
```

Należy zwrócić szczególną uwagę na tworzenie i używanie tablic kodowań, które różnią się od tablic wykorzystywanych przez `dvips` lub pdfTEX. Format pliku z tablicą jest następujący:

```
font_w_TeX-u  kodowanie
  font_wektorowy -opcje
```

przy czym jako pierwszy argument występuje TEX-owa nazwa fontu, następnie kodowanie (rozszerzenie `.enc` może być dodane ale nie jest wymagane), jako trzeci argument podana jest albo postscriptowa nazwa fontu (o ile jest to jeden ze standardowych fontów postscriptowych) albo też nazwa pliku z fontem Type1 (`.pfb` – rozszerzenie dodawane jest przez program). Jako dodatkowe opcje mogą wystąpić:

opcja	opis
-r	pozwala na rempapowanie fontów
-e *liczba*	pozwala na rozciągnięcie fontu poziomo, mnożąc podstawową szerokość znaku w foncie przez podaną *liczbę*
-s *liczba*	pozwala na pochylenie fontu

Przykładowo, fragment pliku mapowań dla fontów PL:

```
plb10    plrm.enc  plb10
plbx10   plrm.enc  plbx10
...
plbsy10  plms.enc  plbsy10
...
% Antykwa Półtawskiego
antpr  antp.enc antpr
antpri antp.enc antpri
...
```

dvips i Ghostscript

`dvips` jest to program służący do przetwarzania plików `.dvi` na pliki postscriptowe, które następnie można m.in. konwertować do formatu PDF. Główny plik konfiguracyjny `config.ps` umieszczony jest w katalogu `/texmf/dvips/config`. W katalogu tym znajduje się szereg innych plików, a wśród nich m.in. plik `config.pdf`, dzięki któremu `dvips` powinien generować pliki postscriptowe optymalizowane do późniejszego ich destylowania. O tym, który plik konfiguracyjny zostanie użyty decyduje opcja `-P` podana przy wywołaniu programu, np.:

```
dvips -Ppdf artykul.dvi
```

spowoduje użycie pliku `config.pdf`. Domyślnie, bez podawania opcji w linii poleceń, używany jest plik `config.ps`. Interesująca nas zawartość tego pliku, dotycząca fontów, przedstawiać się może następująco:

```
% Default resolution of this device,
% in dots per inch.
D 600
X 600
Y 600

% Metafont mode.
M ljfour

% Partially download Type 1 fonts
% by default.
j

% Configuration of postscript
% type 1 fonts:
p psfonts.map
p +pcr.map
p +pl.map
```

Pierwsza grupa parametrów określa rozdzielczość urządzenia, dla którego przygotowywane będą pliki postscriptowe, druga (`M ljfour`) informuje, iż METAFONT będzie pracował w trybie dla drukarki Laser Jet 4. Opcja `j` nakazuje częściowe wbudowywanie używanych fontów w dokument zaś za kolejność wczytywania plików z mapami fontów odpowiada ostatnia część pliku. Zawartość plików z mapami fontów jest taka sama jak w przypadku pdfTEX-a.

Instalowanie fontów... opisano na przykładzie bardzo ładnych fontów ozdobnych Web–O–Mints. Po skopiowaniu archiwum z serwera CTAN (znajduje się w katalogu `/fonts/webomints` na dysk lokalny należy postępować zgodnie z załączoną instrukcją (plik `Readme`):

- skopiować plik `WebOMintsGD.tfm` do katalogu `texmf/fonts/tfm/public/misc`. Jeżeli takiego katalogu nie ma, należy go utworzyć;
- umieścić `WebOMintsGD.pfb` w katalogu `texmf/fonts/type1/public/misc`;
- skopiować plik `uwebo.fd` do katalogu `texmf/tex/latex/misc`;
- skopiować plik `webo.map` do katalogu `texmf/dvips/config`;
- skonfigurować `dvips` tak, aby „widział" zawartość pliku z mapowaniem dodając do pliku `/texmf/dvips/config/config.ps` linijkę:

  ```
  p +webo.map
  ```

- odświerzyć bazy danych TEX-a (przy pomocy programu `texhash` w dystrybucji teTEXlub `initexmf -u` w dystrybucji MikTEX).

Teraz należy sprawdzić, czy zamieszczone przykłady kompilują się i (co ważniejsze) używają fontu wektorowego. Kompilujemy przykład:

```
latex sample1.tex
```

przetwarzamy `dvipsem` do pliku postscriptowego

```
dvips sample1.dvi
```

Dalsze kroki Po przetworzeniu pliku `.dvi` do formatu PS plik wynikowy można poddać dalszej „obróbce", a mianowicie przedestylować go do formatu PDF. W tym celu można skorzystać z Ghostscripta, a szczególnie przydatnym będzie jeden ze skryptów `ps2pdfxx` gdzie `xx` oznacza wersję formatu PDF w jakiej ma być utworzony plik wynikowy. Kontynuując powyższy przykład:

```
ps2pdf sample1.ps sample1.pdf
```

otrzymujemy dokument wynikowy:

The King and Queen of Hearts were seated on their throne when they arrived, with a great crowd assembled about them — all sorts of little birds and beasts, as well as the whole pack of cards: the Knave was standing before them, in chains, with a soldier on each side to guard him; and near the King was the White Rabbit, with a trumpet in one hand, and a scroll of parchment in the other. In the very middle of the court was a table, with a large dish of tarts upon it: they looked so good, that it made Alice quite hungry to look at them — "I wish they'd get the trial done," she thought, "and hand round the refreshments!". But there seemed to be no chance of this, so she began looking at everything about her to pass away the time.

Alice had never been in a court of justice before, but she had read about them in books, and she was quite pleased to find that she knew the name of nearly everything there. "That's the judge," she said to herself, "because of his great wig.".

The judge, by the way, was the King, and as he wore his crown over the wig, (look at the frontispiece if you want to see how he did it,) he did not look at all comfortable, and it was certainly not becoming.

"And that's the jury-box," thought Alice, "and those twelve creatures," (she was obliged to say "creatures," you see, because some of them were animals, and some were birds) "I suppose they are the jurors.". She said this last word two or three times over to herself being rather proud of it: for she thought, and rightly too, that very few little girls of her age knew the meaning of it at all. However, "jurymen" would have done just as well.

The twelve jurors were all writing very busily on slates. "What are they doing?" Alice whispered to the Gryphon. "They can't have anything to put down yet, before the trial's begun.".

"They're putting down their names," the Gryphon whispered in reply, "for fear they should forget them before the end of the trial.".

"Stupid things!" Alice began in a loud indignant voice, but she stopped herself hastily, for the White Rabbit cried out, "Silence in the court!"; and the King put on his spectacles and looked anxiously round, to make out who was talking.

Ghostscript pozwala na kontrolę wbudowywania fontów do pliku `.pdf` co gwarantuje, że dokładnie takim samym fontem będzie renderowany ten plik w przeglądarce, niezależnie od tego, czy font ten jest zainstalowany w systemie, na którym `.pdf` jest wyświetlany.

Do kontrolowania fontów można wykorzystać następujące opcje:

`AlwaysEmbed` – parametr zawierający nazwę (lub ciąg nazw) fontów, które zawsze powinne być wbudowywane w plik `.pdf`, przy czym każda nazwa musi być postscriptową nazwą fontu.

`CannotEmbedFontPolicy` – parametr pozwalający na obsługę błędów w przypadku braku możliwości wbudowania fontu. Dopuszczalne wartości to: `OK` – powodująca zignorowanie błędu jeśli font nie może zostać znaleziony lub wbudowany, `Warning` – powodująca wyświetlenie ostrzeżenia i kontynuowanie pracy oraz `Error` – powodująca zakończenie pracy w przypadku braku możliwości znalezienia bądź wbudowania fontu.

`EmbedAllFonts` – parametr, który przy wartości `true` powoduje, iż wszystkie fonty (oprócz wymienionych w `NeverEmbed`) są wbudowywane.

`MaxSubsetPct` – parametr określający w procentach maksymalną ilość znaków w foncie, która może być użyta zanim cały font jest wbudowywany (zamiast podzbioru); parametr ten jest używany tylko gdy `SubsetFonts` ma wartość `true`.

`NeverEmbed` – nazwa lub ciąg nazw fontów, które nie mogą być wbudowywane.

`SubsetFonts` – parametr pozwalający (dla wartości `true`) na wbudowywanie tylko używanego w dokumencie podzbioru znaków z fontu (zamiast całego fontu), co pozwala na zmniejszenie wielkości wynikowego pliku `pdf`. Jeżeli parametr ten zostanie włączony, decyzja o wbudowywaniu całego fontu lub tylko jego podzbioru zależna jest od ustawienia opisanego wcześniej parametru `MaxSubsetPct`.

Poszczególne opcje można podawać przy wywoływaniu Ghostscripta, np.:

```
ps2pdf
  -dSubsetFonts=true -dMaxSubsetPct=10
  artykul.ps artykul.pdf
```

Rozprowadzany wraz z Ghostscriptem skrypt `dvipdf` pozwala na „bezpośrednie” przetwarzanie plików `.dvi` do formatu PDF – wykorzystuje on program `dvips` z opcją `-q` a następnie przetwarza wyjście tegoż Ghostscriptem z opcjami:

```
-q -dNOPAUSE -dBATCH -sDEVICE=pdfwrite
```

Literatura

[1] Bogusław Jackowski, Stanisław Wawrykiewicz. Fontowe abc. *Biuletyn GUST*, 9:7–12, 1997. Dokument dostępny w: `http://www.gust.org.pl/fonty/font-abc.html`.

[2] Grzegorz Sapijaszko. Tworzenie dokumentów pdf przy pomocy LaTeX-a. Dokument dostępny w: `http://www.sapijaszko.net/pedeefy.pdf`, 2001.