➡ LATEX

LATEX2ε a sprawa polska

Marcin Woliński

Wybrane problemy polonizacji

Niniejszy tekst omawia problemy związane z używaniem LATEXa 2ε do składu dokumentów w języku polskim.

Zapewne należy zacząć od wyjaśnienia, dlaczego warto używać LATEXa 2ε zamiast LATEXa 2.09. Oto garść zalet nowej wersji programu:

- możliwość uzyskania poprawnego składu po polsku bez konieczności utrzymywania oddzielnego formatu,
- możliwość zmiany za pomocą jednego polecenia zestawu fontów używanego przez dokument, np. na fonty postscriptowe,
- rozszerzalność zestawu dostępnych fontów, bez ingerencji w plik formatu,
- duży zestaw standardowych pakietów, np. do włączania plików graficznych i pracy w kolorze (część z nich dostępna jest również w LATEXu 2.09, ale wersje te nie są już utrzymywane),
- uporządkowany interfejs klas i pakietów – próba wprowadzenia modularyzacji do programowania w TEXu, dająca pewną szansę na zapanowanie nad dużym systemem, jakim niewątpliwie jest LATEX.

Główną ideą LATEXa 2ε jest zintegrowanie w nowy standardowy format różnych rozszerzeń, które wyrosły wokół LATEXa 2.09. Wydaje się, że w ogólnym zarysie to się udało i zarówno interfejs użytkownika, jak i interfejs twórcy pakietów jest lepszy w nowej wersji.

Ale uwaga: LATEX2ε podlega ciągłym zmianom! Nie można zakładać, że wszystko co działa w obecnej wersji, będzie działało również w następnych. W szczególności The LATEX Team zapowiada wprowadzenie do formatu jakiejś formy wsparcia dla wielojęzyczności, a pakiet inputenc ciągle przedstawia się jako „beta version".

Dalsza część tego tekstu dotyczy LATEXa 2ε w jego obecnej postaci.

Narzędzia polonizacji LATEXa. Dwa pakiety są postrzegane jako mające coś wspólnego z problemem. Są to Babel i PLATEX.

PLATEX jest produktem rodzimym – jego autorem jest Mariusz Olko (małe fragmenty wersji 1.0 zostały napisane przeze mnie). Z założenia służy do składu dokumentów tylko po polsku. Instrukcja użytkowania PLATEXa stanowi drugi punkt tego artykułu.

Pakiet Babel Johannesa Braamsa ma ambicje bycia pakietem wielojęzycznym i przeznaczonym do tworzenia wielojęzycznych dokumentów.

Osoby faktycznie tworzące dokumenty wielojęzyczne są chwilowo skazane na jego użycie. Babel jest jednak miernym narzędziem do składu dokumentów po polsku ze względu na liczne błędy, zwłaszcza we fragmentach poświęconych polszczyźnie. Do składu dokumentów polskojęzycznych lepszym narzędziem jest PLATEX.

W dalszym ciągu tekstu omówię wybrane problemy polonizacji, wskazując jak ma się do nich Babel i PLATEX.

Na wstępie jeszcze jedna obserwacja: nie istnieje LAMEX2e. Nie istnieje, ponieważ nie ma takiej konieczności. LAMEX był formatem przypominającym LATEXa 2.09 z dwiema zasadniczymi różnicami: miał wbudowane polskie wzorce dzielenia wyrazów i posługiwał się polskimi fontami zamiast oryginalnych – wprowadzenie tych zmian wymagało sporządzenia oddzielnego pliku formatu. LATEX2ε może mieć wbudowany dowolny zestaw wzorców dzielenia i nadal pozostaje Standardowym LATEXem. Natomiast zmiana zestawu używanych fontów może się dokonać w dowolnym momencie pracy programu, nawet w środku przetwarzania dokumentu. Dlatego w przypadku LATEXa 2ε stało się zbędne utrzymywanie oddzielnego formatu do składu po polsku.

I jeszcze jedna uwaga natury ogólnej: większość trudności związanych z polonizacją została pokonana przez Bogusława Jackowskiego i Marka Ryćko przy tworzeniu MEXa. W przypadku LATEXa 2ε wystarczyło dostosować już istniejące rozwiązania do nowych warunków.

Wzorce dzielenia wyrazów. Pierwszym problemem, który każdemu kojarzy się z polonizacją, jest nauczenie TEXa właściwego dzielenia przy przenoszeniu do nowej linii słów polskich. Szczęśliwie odpowiednie wzorce dzielenia zostały przygotowane przez Hannę Kołodziejską, Bogusława Jackowskiego i Marka Ryćko i są one zawarte zarówno w dystrybucji Babela jak i PLATEXa w pliku `plhyph.tex`. (Babel przez długi czas zawierał przestarzały plik `pohyph.tex`, na szczęście należy to już do przeszłości).

Aby TEX mógł posługiwać się wzorcami dzielenia wyrazów, muszą one zostać odpowiednio przetworzone przez program iniTEX. Oznacza to, że zestaw wzorców dzielenia wyrazów wybiera się na etapie generowania pliku formatu i w trakcie pracy TEXa będą dostępne tylko te wzorce, które przewidziano w formacie.

Zestaw wzorców wbudowany w plik formatu LATEXa 2_ε można określić tworząc plik konfiguracyjny `hyphen.cfg`. Zarówno w Babelu, jak i PLATEXu zastosowano następujące rozwiązanie: lista języków i odpowiadających im nazw plików z wzorcami znajduje się w pliku `language.dat`, natomiast plik `hyphen.cfg` zawiera makrodefinicje przetwarzające tę listę. W `language.dat` należy zainterweniować przed przystąpieniem do generowania formatu, aby wybrać właściwy zestaw języków.

Z wersją `hyphen.cfg` zawartą w Babelu wiąże się jednak przykry problem. Babelowy `hyphen.cfg` zawiera globalne definicje nowych poleceń (np. `\selectlanguage`), które są wbudowywane w format. Zgodnie z licencją LATEXa 2_ε tak wygenerowany format nie ma prawa używać nazwy `latex`, aby nie wprowadzać w błąd użytkowników – musi istnieć tylko jeden Standardowy LATEX i zestaw poleceń w nim dostępnych musi być ściśle określony. Ale „zbabelizowane" dystrybucje TEXa nazywają format zawierający `\selectlanguage` `latex.fmt`. Autorzy nagabywani w tej sprawie odpowiadają, że Babel stanowi próbną wersję interfejsu, który będzie włączony do Standardowego LATEXa 2_ε. Ale w takim razie czy obecna wersja LATEXa 2_ε zawiera polecenie `\selectlanguage`?

Można z tego wyciągnąć jeden dosyć smutny wniosek: wszyscy autorzy pakietów są równi, ale niektórzy autorzy są równiejsi niż inni.

Drugim nadużyciem, którego biorąc rzecz formalnie autor Babela nie popełnia, ale do którego podjudza, jest podmiana uświęconego tradycją pliku `hyphen.tex`. Ten plik powinien zawierać oryginalne amerykańskie wzorce dzielenia dystrybuowane przez Knutha. Tymczasem w instalacjach „zbabelizowanych" zawiera on kawałki kodu Babela. Żeby było ciekawiej, w dokumentacji LATEXa (`cfgguide.tex`) jest napisane, że plik `hyphen.tex` zawiera amerykańskie wzorce dzielenia wyrazów. Szczęśliwie autor Babela zapewnia, że w wersji 3.7 problem zniknie.

Dokumenty używające pakietu Babel mogą być przetwarzane przy pomocy formatu wygenerowanego z użyciem plików konfiguracyjnych pochodzących z PLATEXa (Babel jest w stanie załadować w trakcie przetwarzania dokumentu definicje, które zwykł wbudowywać w format – nie ma tu umniejszenia funkcjonalności).

Również dokumenty używające pakietu PLATEX mogą być przetwarzane formatem wygenerowanym z użyciem plików konfiguracyjnych z Babela.

Z powodu kłopotów wymienionych powyżej, proponuję raczej używanie formatu wygenerowanego przy pomocy PLATEXa – jest on zgodny z licencją LATEXa 2_ε, a dzięki możliwości wybrania dowolnego zbioru wzorców dzielenia wyrazów nadaje się również do składu wielojęzycznego.

Polskie kroje pisma. W chwili obecnej w LATEXu 2_ε można łatwo korzystać z następujących fontów zawierających dobrej jakości polskie znaki:

- fonty PL – polska adaptacja fontów Computer Modern dokonana przez Bogusława Jackowskiego i Marka Ryćko,
- fonty EC – Extended Computer Modern – „europejskie" rozszerzenie fontów Computer Modern,
- fonty PC – polska adaptacja fontów Computer Concrete (Jackowski, Ryćko),
- Quasi Palladio i Quasi Times (Jackowski),
- antykwa toruńska (Janusz M. Nowacki).

Fonty EC i LATEXowe wsparcie dla nich jest obecnie dostępne we wszystkich dystrybucjach TEXa. Wsparcie dla fontów PL i PC jest zawarte w pakiecie PLATEX. Aby umożliwić LATEXowi dostęp do fontów Quasi Palladio, Quasi Times i antykwy toruńskiej, należy zainstalować pakiet pl-qx Piotra Kłosowskiego.

Istnieje również możliwość użycia fontów wirtualnych opartych na krojach wbudowanych w typowe urządzenia postscriptowe. Niestety kształty tak uzyskanych polskich znaków pozostawiają wiele do życzenia.

Wprowadzanie znaków narodowych. Istnieją trzy zasadnicze sposoby wprowadzania polskich znaków w LATEXu: standardowe polecenia, notacja prefiksowa i notacja „naturalna" (8-bitowa).

Wśród standardowych poleceń LATEXa 2_ε oprócz `\'`, `\.`, `\l` i `\L` znalazło się polecenie `\k` oznaczające „akcent ogonek". Znaczy to, że w LATEXu 2_ε jest wreszcie dostępny standardowy i jednoznaczny zapis dla wszystkich polskich znaków narodowych. Co więcej w LATEXu 2_ε, zapis np. `\'o` oznacza „użyj najlepszej metody uzyskania znaku oacute dostępnej w bieżącym układzie fontu". Jeżeli w bieżącym foncie jest dostępny znak oacute, zostanie on użyty bezpośrednio. Jeżeli go nie ma, zostanie użyte polecenie pierwotne \accent do położenia akcentu acute nad literą o. Jeżeli w użyciu byłby układ nie zawierający któregoś z tych znaków, LaTeX powiadomiłby o niemożności uzyskania znaku oacute.

Z notacją tą wiąże się jednak dobrze znany problem: środowisko tabbing zmienia lokalnie definicję kilku makr, w tym \'. Dlatego w jego obrębie do uzyskiwania znaków z akcentem acute trzeba używać notacji typu \a'o. Konsekwencją tego faktu jest, że zarówno w implementacji notacji prefiksowej, jak i „stron kodowych" pakietu inputenc trzeba dostawać się do akcentu acute nieco naokoło.

Posługiwanie się powyższymi poleceniami do pisania tekstu po polsku jest oczywiście bardzo uciążliwe. Stąd pomysł notacji prefiksowej – polega ona na zapisywaniu polskich znaków diakrytycznych w postaci dwóch znaków: ustalonego znaku specjalnego (prefiksu) i litery łacińskiej podobnej graficznie do odpowiedniej polskiej.

Notacja prefiksowa doczekała się wielu wersji i wcieleń: stosowano różne znaki jako prefiks (co najmniej ", @ i /) oraz różne konwencje zapisywania znaków ź oraz ż (któryś z nich musi być zapisywany przy pomocy znaku różnego od z).

Babel stosuje jako prefiks cudzysłów, oto zapis minuskuł „polskich znaków" w notacji Babela:

```
"a "c "e "l "n "o "s "z "r
```

Implementacja notacji prefiksowej w Babelu nie działa z fontami PL.

PLaTeX oferuje implementację notacji prefiksowej z użyciem ukośnika („ciacha"), wprowadzoną po raz pierwszy w formacie MeX (Rycko, Jackowski):

```
/a /c /e /l /n /o /s /x /z
```

Podwojony ukośnik w pliku źródłowym daje ukośnik w składzie. PLaTeXowa implementacja wykorzystuje standardowe polecenia akcentowe i mechanizm układów fontów.

Niektórzy użytkownicy TeXa są bardzo silnie związani uczuciowo z notacją, którą stosowali od dziesięciu lat, i odmawiają kategorycznie zmiany swoich przyzwyczajeń. Ze względu na nich, w dystrybucji PLaTeXa został umieszczony pakiet plprefix pozwalający na dosyć dowolny wybór stosowanej notacji polskich znaków.

Ośmiobitowe wprowadzanie polskich znaków. Niektóre implementacje TeXa mają konfigurowalną tabelę kodów wejściowych. Oznacza to, że znaki czytane z wejścia podlegają konwersji, zanim zobaczy je TeX. Sposób przekodowania zadaje się w trakcie generowania formatu (emTeX) albo w wywołaniu TeXa (web2c). Tabele konwersji są zależne zarówno od kodu wejściowego, jak i od układu używanych fontów. Na przykład tabela dla kodu ISO 8859-2 i fontów PL przekodowuje znak o numerze 177 (litera ą w ISO 8859-2) na znak 161 (pozycja znaku aogonek w fontach PL).

Największą zaletą przekodowywania w emTeXu jest to, że przy wypisywaniu tekstów na ekranie i do plików emTeX wykonuje transformację odwrotną, co oznacza, że na ekranie pojawia się tekst czytelny, zamiast napisów typu ^^bb^^f3^^aa^^a2 (co ma oznaczać żółć). Ułatwia to współpracę z programami zewnętrznymi (Makeindex, bibTeX).

Podstawową wadą jest to, że ten sposób praktycznie uniemożliwia wykorzystywanie w jednym dokumencie fontów w różnych układach. Pozycje „polskich znaków" są takie same praktycznie we wszystkich powszechnie używanych w TeXu fontach zawierających polskie znaki. Można by więc „zapomnieć" o problemie. Jeśli jednak próbować zrobić tabelę konwersji dla pełnej strony kodowej, np. cp852, cp1250 albo ISO 8859-2, okaże się, że jest ona zależna od układu fontu, przede wszystkim dlatego, że zestawy dostępnych w fontach znaków są różne.

Druga metoda polega na wykorzystaniu znaków aktywnych. Pakiet inputenc „aktywizuje" znaki o kodach 128–255. To znaczy, że górna połówka kodu jest uznawana za wymienną, a dolna powinna być zgodna z ASCII (jest to prawdą dla większości stosowanych kodów). Definicja kodu ISO 8859-2 w pakiecie inputenc definiuje znak o kodzie 177 (litera ą) jako ciąg \k{a}. Taka definicja jest niezależna od układu fontów – to LaTeXowy mechanizm znaków akcentowanych zapewnia dostęp do odpowiedniego znaku z fontu.

Zaletą jest tu fakt, że dokumenty używające pakietu inputenc zawierają w sobie informację o użytej stronie kodowej. Jeżeli otrzymamy od kogoś plik zapisany w taki sposób, to nawet jeśli używane przez nas oprogramowanie nie pozwala wyświetlić poprawnie tekstu źródłowego, jesteśmy w stanie przetworzyć dokument i wynik składu będzie poprawny. Nie można tego powiedzieć o dokumentach korzystających z przekodowania na poziomie implementacji.

Podstawowym problemem jest, że w plikach pisanych przez LaTeXa 2_ε znaki akcentowane są zapisywane w postaci \k a, \'c itd., co utrudnia ich przetwarzanie przez programy zewnętrzne.

Pakiet PLATEX

Jak powiedziano wyżej, PLATEX jest pakietem umożliwiającym skład w LATEXu w języku polskim. Zadanie to obejmuje:

- dostęp do fontów zawierających polskie znaki akcentowane,
- poprawne przenoszenie słów polskich,
- obsługę zamiany minuskuł polskich liter na majuskuły i odwrotnie,
- w implementacji TEXa z konfigurowalną tabelą kodów wejściowych definiowanie makr o nazwach zawierających „polskie znaki",
- notację prefiksową polskich znaków,
- polskie napisy generowane automatycznie (spis treści, rozdział, ...) i datę generowaną przez `\date`,
- poprawne przenoszenie wyrazów złożonych typu biało-czerwony (makro `\dywiz`),
- polskie symbole funkcji trygonometrycznych `\tg`, `\ctg`, `\tgh`, `\ctgh` i `\arc` i polskie znaki większe-równe i mniejsze-równe,
- zmianę znaczenia standardowych nazw `\tan`, `\ctan` itd. na polskie odpowiedniki (sterowaną opcją pakietu).

Sposób użycia pakietu. Każdy dokument w języku polskim (używający PLATEXa) musi zawierać w preambule deklarację

```
\usepackage{polski}
```

Deklaracja ta może również zadawać dodatkowe opcje. Oto lista dostępnych możliwości.

`OT4` Ładuje definicję układu fontów `OT4` i wybiera go jako domyślny dla dokumentu. W praktyce oznacza to skład czcionkami PL. (Pojęcie układu fontów jest omówione w trzeciej części tekstu.)

`T1` Ładuje definicję układu `T1` i wybiera go jako domyślny dla dokumentu. W praktyce oznacza to skład „europejskimi" czcionkami EC.

`OT1` Zaznacza świadomą decyzję użycia „starego" układu tekstowego `OT1`. Układ ten nie zawiera kompletu znaków potrzebnego do składu w języku polskim (!).

`plmath` Nakazuje użycie czcionek PL w trybie matematycznym. Dodatkowo zmienia LATEXową definicję symboli większe-równe oraz mniejsze-równe.

`nomathsymbols` Blokuje zmianę przez styl znaczenia standardowych symboli funkcji trygonometrycznych oraz relacji większe-równe i mniejsze-równe (zob. niżej),

`MeX` Tryb zgodności z MEXem. Ta opcja definiuje wszystkie makra, które są normalnie dostępne dla użytkownika w MEXu.

Jeżeli nie użyto żadnej z opcji wyboru układu fontów, `polski.sty` próbuje włączyć fonty PL, jeżeli są one zainstalowane. Dotyczy to zarówno fontów tekstowych, jak i matematycznych. W instalacji zawierającej fonty PL wywołanie pakietu bez opcji jest równoważne wywołaniu

```
\usepackage[OT4,plmath]{polski}
```

Opcja `OT1` służy do powiedzenia pakietowi, że użytkownik świadomie używa układu nie zawierającego kompletu znaków potrzebnych do składu po polsku.

Pakiet `polski.sty` definiuje polskie symbole funkcji trygonometrycznych `\tg`, `\ctg`, `\tgh`, `\ctgh`, a także `\arc`. Jeżeli użyto opcji `plmath` dostępne będą również polskie symbole relacji: `\xleq` $\leqslant$ i `\xgeq` $\geqslant$.

Nadto `polski` zmienia znaczenie makr `\tan`, `\ctan`, `\tanh`, `\ctanh`, `\arcsin`, `\arccos`, `\arctan`, `\arcctan`, `\leq` i `\geq`, tak aby produkowały polskie odpowiedniki. Ponieważ zamiana taka nie zawsze jest pożądana, pakiet udostępnia blokującą ją opcję `nomathsymbols`.

Polskie reguły ortograficzne wymagają, aby przy przeniesieniu wyrazu pisanego z dywizem (np. biało-czerwony) dywiz został powtórzony na początku nowego wiersza. Na tę okoliczność PLATEX oferuje makro `\dywiz`. Poprawnym zapisem słowa biało-czerwony jest `biało\dywiz czerwony`.

Makro `\selecthyphenation` (dostępne od wersji 1.01 pakietu) pozwala na przełączanie używanego zestawu wzorców dzielenia wyrazów. Argumentem tego makra jest nazwa języka, taka jak w pliku `language.dat`, np.:

```
\selecthyphenation{english}
\selecthyphenation{polish}
```

Jest to makro niskopoziomowe, które nie zmienia nic poza samymi wzorcami dzielenia wyrazów. Jak zaznaczyłem wcześniej PLATEX nie jest pakietem wielojęzycznym.

Notacja prefiksowa jest domyślnie wyłączona, można ją uaktywnić wydając polecenie \prefixing. Zaleca się stosowanie tego polecenia po załadowaniu wszystkich pakietów, ponieważ uczynienie makra z ukośnika może powodować problemy przy ich ładowaniu. Polecenie \prefixing jest lokalne, obowiązuje do końca bieżącej grupy lub do wystąpienia polecenia \nonprefixing.

Ośmiobitowe wprowadzanie „polskich znaków" wymaga użycia pakietu inputenc. W poprzednich wersjach istniała możliwość wywoływania pakietu inputenc przez polski poprzez podanie temu ostatniemu opcji nazywającej stronę kodową. Możliwość ta została usunięta, ponieważ polski nie dodawał nic do funkcjonalności pakietu inputenc, a przekazywanie opcji tylko zaciemniało obraz sytuacji.

Instalowanie PLATEXa. Kilka razy spotkałem się z zarzutem, że PLATEX wymaga skomplikowanego instalowania w odróżnieniu od Babela, z którego wystarczy „skopiować jeden styl". Główną część funkcjonalną PLATEXa stanowi jeden plik polski.sty. Oprócz tego pakiet zawiera pliki instalujące fonty PL i pliki konfigurujące format, o których wspominałem wyżej. „Kopiując jeden plik" można osiągnąć podobny zakres funkcjonalności jak w Babelu. Pozostałe elementy pozwalają natomiast osiągnąć więcej.

Dla PLATEXa najlepiej przeznaczyć oddzielny katalog, gdzieś w strukturze przeszukiwanej przez TEXa (stosownie do implementacji). Po rozpakowaniu pakietu należy wykonać polecenie

```
latex platex.ins
```

Spowoduje to wygenerowanie kilkunastu plików, wśród nich polski.sty.

Dla uzyskania poprawnego przenoszenia słów polskich konieczne jest zapewnienie obecności polskich wzorców dzielenia wyrazów w formacie. W tym celu proponuję usunąć pliki hyphen.cfg, language.dat i hyphen.tex w wersji z Babela (jeśli ten jest w systemie) i zaopatrzyć się w prawdziwy plik hyphen.tex z wzorcami Knutha (w niektórych systemach jest on dostępny jako ushyph1.tex). Następnie trzeba uruchomić iniTEXa na pliku latex.ltx. Dokładna postać polecenia jest zależna od systemu, np. w emTEXu:

```
tex386 /i /mt13000 latex.ltx
```

(można jeszcze dodać opcję sterującą konwersją znaków), a pod UNIXem:

```
initex latex.ltx
```

Powstały w wyniku plik latex.fmt trzeba umieścić w katalogu właściwym dla formatów.

Przykładowy dokument w notacji prefiksowej, fonty PL:

```
\documentclass{article}
\usepackage{polski}
\prefixing
\begin{document}
  Za/z/o/l/c g/e/sl/a ja/x/n.
\end{document}
```

oraz w notacji naturalnej, zakładając kod ISO 8859-2 (Latin-2), fonty EC (wybór fontów jest oczywiście niezależny od wyboru notacji wejściowej):

```
\documentclass{article}
\usepackage[T1]{polski}
\usepackage[latin2]{inputenc}
\begin{document}
  Pójdź kińże tę chmurność w~głąb flaszy.
\end{document}
```

Jeżeli stosowane jest przekodowanie na poziomie implementacji, w powyższym przykładzie trzeba usunąć wywołanie pakietu inputenc.

Czego PLATEX nie robi. Dokumenty składane przy pomocy PLATEXa i standardowych klas są spolszczone w sensie opisanym powyżej. Są jednak oparte na klasach o amerykańskim układzie, co powoduje, że w większości złożonych dokumentów widać, że wyszły z amerykańskiego programu składu.

Umieszczenie w pakiecie polski.sty fragmentów zmieniających projekt typograficzny standardowych klas nie wydaje się dobrym pomysłem. Spowodowałoby to, że polski stałby się zależny od tych konkretnych klas. Obecna postać pakietu jest w stanie spolonizować każdy dokument LATEXa.

Należałoby zatem opracować zestaw nowych klas dokumentów o polskim projekcie typograficznym, zgodnych z oczekiwaniami redaktora technicznego ze „starych dobrych czasów". Klasy takie mogłyby również być lepiej sparametryzowane niż oryginały (aby ułatwiać modyfikację wyglądu dokumentów) oraz unikać ich znanych błędów (np. dzielenia wyrazów w tytułach).

Fonty w LATEXu 2ε: NFSS

W niniejszym wprowadzeniu w temat fontów zostanie przedstawiony wyłącznie sposób zarządzania fontami tekstowymi. Do prezentacji zagadnień związanych ze składem formuł matematycznych autor nie czuje się jeszcze odpowiednio przygotowany.

TEX ma wbudowany jedynie bardzo prosty sposób dostępu do fontów. Poleceniem `\font` można związać nazwę symboliczną z operacją wybrania fontu załadowanego z konkretnego pliku metrycznego i o ustalonej wielkości[1]. Posługiwanie się tak zdefiniowanymi poleceniami na dłuższą metę jest uciążliwe, ponieważ wywołują one bezpośrednio określoną odmianę i stopień określonego kroju pisma. Wyobraźmy sobie, że wykonano

```
\font\it=plti10
```

Jeśli teraz fragment tekstu zawierający użycie tego polecenia zostanie przeniesiony z głównego tekstu dokumentu do przypisu, wydarzy się katastrofa: przypisy są tradycyjnie składane pismem mniejszego stopnia, ale fragment kursywą będzie większy!

Dlatego każdy TEXowy format ma swoją metodę zdefiniowania polecenia `\it` tak, aby znaczyło coś w rodzaju „przełącz font na kursywę w bieżącym stopniu". LATEX 2.09 zapewniał dostęp do ustalonego zbioru fontów – mianowicie krojów Computer Modern – tak aby dawało się manipulować niezależnie stopniem i odmianą pisma. Dodanie nowych fontów do zestawu dostępnego w LATEXu w praktyce wymagało ingerencji w plik `lfonts.tex`, co było zadaniem wyjątkowo niewdzięcznym, a ponadto prowadziło do powstawania niezgodnych ze sobą lokalnych wersji LATEXa.

W LATEXa 2$_\varepsilon$ wbudowano skomplikowany, ale za to bardzo elastyczny, mechanizm nazywający się NFSS (Nowy sposób wyboru fontów). System ten jest stosunkowo łatwy w użyciu, zarejestrowanie fontów w systemie jest elementarne (jeżeli dostępne są pliki opisu fontów – *font descripton files*) i nie wymaga ingerencji w format LATEXa, natomiast siłą rzeczy stworzenie plików opisu wymaga odrobiny wiedzy o systemie.

Opis fontu w nowym sposobie wyboru fontów. W LATEXu 2$_\varepsilon$ bieżący font jest charakteryzowany przez pięć elementów: układ (zestaw znaków), krój (rodzinę), grubość i szerokość, odmianę, stopień i interlinię. Każdy z tych elementów można wybrać oddzielnym poleceniem.

układ fontu	`\fontencoding{OT1}`
rodzina (krój)	`\fontfamily{cmr}`
grubość i szerokość	`\fontseries{m}`
odmiana	`\fontshape{n}`
stopień i interlinia	`\fontsize{10}{12pt}`
	`\selectfont`

Argumenty poleceń wymienione w tabelce odpowiadają domyślnym wartościom dla dokumentu składanego przy pomocy standardowych klas w stopniu 10pt. Użycie poleceń `\font...` nie powoduje od razu włączenia fontu, ponieważ użytkownik może chcieć zmienić kilka elementów charakterystyki bieżącego fontu jednocześnie. Dlatego ostateczny wybór fontu zatwierdza się poleceniem `\selectfont`.

Wszystkie elementy charakterystyki fontu, za wyjątkiem stopnia pisma, można zmienić jednocześnie poleceniem `\usefont`. Nie wymaga ono również zatwierdzenia przez `\selectfont`.

```
\usefont{OT1}{cmr}{m}{n}
```

Poszczególne elementy opisu fontu są od siebie niezależne, co daje LATEXowi przewagę nad procesorami tekstu. Wyobraźmy sobie, że w „zwykłym procesorze tekstu" w tekście po polsku używamy wyrażenia «à propos». Ponieważ w „polskim" foncie nie ma znaku à, musimy pobrać symbol z fontu „zachodniego". Jeżeli w późniejszym etapie opracowania redakcyjnego postanowimy zmienić krój pisma np. z Timesa na Helveticę, obca litera pozostanie w starym kroju, bo uzyskując ją nie mogliśmy powiedzieć „chcę bieżący krój i odmianę, ale w układzie zachodnioeuropejskim", zamiast tego w dokumencie stoi „tu ma być zachodnioeuropejski Times jasny prosty stopnia 10pt".

Gdy NFSS stwierdzi, że dana kombinacja parametrów nie odpowiada żadnemu istniejącemu fontowi, wypisuje odpowiednią informację do pliku `.log` i stara się użyć fontu o możliwie bliskiej charakterystyce. Niektórzy bardzo nie lubią tej własności NFSS, uważając że jest to jedno z miejsc, w których LATEX usiłuje coś (źle) zrobić za użytkownika. To zachowanie można zmienić. Użycie w preambule deklaracji

```
\usepackage[pausing]{tracefnt}
```

powoduje zamianę wszystkich komunikatów o brakujących fontach w komunikaty o błędach, wymagające interwencji użytkownika.

1: W całym niniejszym tekście słowo font będzie używane na określenie takiej właśnie istności. W sensie TEXa – coś załadowanego z konkretnego pliku metrycznego. W sensie typograficznym – pismo o ustalonym kroju, odmianie i stopniu.

OT1	*Old Text 1*	układ fontów zastosowany przez Knutha
OT2	*Old Text 2*	układ cyrylicy z Washington University
OT3	*Old Text 3*	układ fontów wsuipa (?)
OT4	*Old Text 4*	układ fontów PL
T1	*Text 1*	„europejski" układ fontów (Cork)
T3	*Text 3*	IPA – znaki fonetyczne
T4	*Text 4*	układ dla języków afrykańskich
TS1	*Text Symbol 1*	dodatkowe symbole tekstowe
OML	*Old Math Letters*	układ fontu matematycznego wg. Knutha
OMS	*Old Math Symbols*	układ symboli wg. Knutha
OMX	*Old Math Extensible*	układ fontu ze znakami „rozciągliwymi"

Układy fontów w LATEXu 2_ε

Omówię teraz kolejne parametry opisujące font w NFSS.

UKŁAD FONTU jest najciekawszym elementem, ponieważ wydaje się, że LATEX jest jedynym systemem, w którym pojęcie to występuje jawnie. Układ fontu opisuje, jakie znaki są dostępne w foncie i w jakich pozycjach. Polecenia akcentowe w LATEXu korzystają z informacji o układzie bieżącego fontu, aby użyć najlepszej dostępnej metody uzyskania danego znaku.

Ponieważ mechanizm sterowania tłumaczeniem znaków na pozycje w foncie nie występuje w samym TEXu, LATEXowy mechanizm układów działa na poziomie makr. W związku z tym dotyczy on tylko znaków dostępnych poprzez makra. Tabele konwersji kodu wejściowego na poziomie implementacji dokonują tłumaczenia od razu w pozycje znaków w foncie, ignorując ten mechanizm. Z drugiej strony pakiet inputenc posługuje się znakami aktywnymi, które zachowują się jak makra, wykorzystując mechanizm układów fontów.

Nazwy układów są tradycyjnie pisane wielkimi literami i cyframi. Tabela na górze strony zestawia układy fontów należące do standardu LATEXa.

Układy o nazwach, zaczynających się literą O, zawierają po 128 pozycji. Pozostałe są „nowymi" układami 256-znakowymi. OT4 wyłamuje się z tego schematu, będąc rozszerzeniem OT1, ale nie wypełniającym wszystkich 256 pozycji.

Układ T1 jest „nowym" układem opracowanym na konferencji w Cork. Zawiera on znaki umożliwiające skład w większości języków europejskich. TS1 jest odpowiadającym mu układem uzupełniającym zawierającym latające akcenty dla majuskuł (które nie zmieściły się w T1), znaki „copyright", „registered" i „trade mark", liczne symbole walut oraz znaki pauzy o długościach pośrednich między „endash" i „emdash", które powinny uszczęśliwić polskich redaktorów technicznych.

Jak widać, brak „nowych" układów matematycznych. Wydaje się, że nikt nie czuje się kompetentny, by takie opracować, lub też układ zaproponowany przez Knutha funkcjonuje wystarczająco dobrze, a wielojęzyczność nie ma wielkiego zastosowania do formuł matematycznych.

Układy fontów muszą być jawnie deklarowane w dokumencie. Wszystkie układy, które będą używane w dokumencie powinny zostać wymienione jako opcje wywołania pakietu fontenc. Na przykład żeby załadować definicje układu OT2 (cyrylica) i OT4 (polski) wykonujemy:

```
\usepackage[OT2,OT4]{fontenc}
```

Ostatni układ wymieniony w tej klauzuli będzie domyślnym układem dokumentu.

KRÓJ PISMA (RODZINA). Krój pisma jest wyróżniony przez jednolitość projektu graficznego. Jest to wszakże dość nieprecyzyjne określenie, jako że np. antykwa i kursywa tego samego kroju różnią się dość znacznie kształtem znaków. Kryterium praktyczne jest więc takie, że krój stanowi taki zbiór fontów, któremu jego projektant nadał wspólną nazwę, np. Quasi Palladio.

W NFSS stosuje się kilkuznakowe skróty nazw krojów. Dla Computer Modern są to np. cmr, cmss, cmtt. Dla krojów postscriptowych stosuje się zwykle trzy pierwsze znaki nazwy plikowej fontu wg. konwencji Karla Berry. Na przykład ptm to Adobe Times, phv to Adobe Helvetica, a bch to Bitstream Charter. Tych nazw używa się jako argumentu `\fonfamily`.

Dla osób po raz pierwszy stykających się z NFSS trudność stanowi zwykle zrozumienie zależności między rodziną a układem fontu. I tak na przykład krój

Computer Modern Roman (cmr) występuje co najmniej w trzech układach. Krój cmr w starym układzie tekstowym OT1 jest dostępny w oryginalnych fontach Knutha o nazwach plikowych np. cmr10, cmbx12, cmti9, itd. Krój cmr w układzie polskim OT4 jest realizowany przez adaptację fontów Knutha, dokonaną przez B. Jackowskiego i M. Ryćko. Nazwy plikowe tych fontów to np. plr10, plbx12, plti9. Trzecim układem, w którym dostępny jest krój cmr, jest nowy „europejski" układ tekstowy T1. Ta kombinacja parametrów oznacza użycie fontów EC (Extended Computer Modern), czyli plików metrycznych np. ecr1000, ecbx1200, ecti0900.

Fonty Knutha oprócz rodziny Computer Modern Roman obejmują jeszcze rodzinę Computer Modern Sans Serif (cmss), Computer Modern Typewriter (cmtt) i kilka drobnych eksperymentalno-dekoracyjno-zabawowych, jak Computer Modern Funny Roman (cmfr), Computer Modern Dunhill Roman (cmdh), Computer Modern Fibonacci Roman (cmfib).

Zatem wybór spośród fontów CM, PL czy EC, odbywa się poprzez zmianę używanego układu, a nie rodziny, bo wszystkie one zawierają rodziny cmr, cmss i cmtt – znaki występujące w każdym z tych fontów mają w zasadzie ten sam projekt graficzny, jedynie zestaw dostępnych znaków jest różny.

Przełączenie standardowego LaTeXa na fonty zawierające polskie znaki diakrytyczne, polega na zmianie używanego układu fontów na OT4 lub T1. Od tego momentu odwołania do standardowych rodzin fontów będą dotyczyły fontów zawierających polskie znaki.

Krojów pisma nie trzeba deklarować. LaTeX załaduje odpowiednie informacje przy pierwszym odwołaniu do danego kroju.

Grubość i szerokość. Oto najczęściej spotykane wartości tego parametru:

m	*medium*	zwykły
b	*bold*	gruby
bx	*bold extended*	gruby szeroki
sb	*semi-bold*	półgruby
c	*condensed*	wąski

W ogólności grubość i szerokość mogą przebiegać kilkustopniową skalę, dając kilkanaście kombinacji opisanych dwuliterowymi kodami. Większość krojów występuje jednak tylko w dwóch lub trzech wersjach grubości/szerokości.

Odmiana. Najczęściej dostępne odmiany to:

n	*normal*	prosta
it	*italic*	kursywa
sl	*slanted*	pochyła
sc	*small caps*	kapitaliki
ui	*upright italic*	„wyprostowana" kursywa

Stopień pisma i interlinia. Argumentami polecenia \fontsize są stopień pisma i interlinia. Domyślną jednostką są punkty (amerykańskie), ale w każdym z argumentów można też użyć dowolnej jednostki akceptowanej przez TeXa.

W TeXu interlinia jest stała dla całego akapitu. Mówiąc ściślej, w całym akapicie zostanie użyta jej wartość bieżąca na jego końcu.

Obsługa typowych sytuacji. Opisany powyżej sposób manipulowania fontami pozwala na dostęp do wszystkich zainstalowanych fontów rodziny, ale jest uciążliwy. Dlatego LaTeX ma serię poleceń (definiowanych po części przez klasy) obsługujących typowe sytuacje.

Po pierwsze, mamy serię poleceń wybierających stopień pisma i interlinię, które występowały również w LaTeXu 2.09. Poniższa tabela wymienia stopnie odpowiadające poszczególnym poleceniom, w zależności od stopnia pisma tekstu podstawowego wybranego w \documentclass.

stopień pisma tekstu głównego	10pt	11pt	12pt
\tiny	5pt	6pt	6pt
\scriptsize	7pt	8pt	8pt
\footnotesize	8pt	9pt	10pt
\small	9pt	10pt	11pt
\normalsize	10pt	11pt	12pt
\large	12pt	12pt	14pt
\Large	14pt	14pt	17pt
\LARGE	17pt	17pt	20pt
\huge	20pt	20pt	25pt
\Huge	25pt	25pt	25pt

Zostały również zachowane polecenia LaTeXa 2.09 \rm, \bf, \it, \sl, \sc, \ss i \tt. Ale uwaga: zachowują się one dokładnie tak jak w LaTeXu 2.09 – nie da się na przykład przy ich pomocy uzyskać grubej kursywy.

Nowy zestaw poleceń posługuje się następującym modelem użycia fontów w dokumencie. Ponieważ mieszanie różnych krojów szeryfowych w jednym dokumencie w zasadzie jest błędem (a przynajmniej powinno być wynikiem przemyślanej decyzji), w schemacie przyjętym przez standardowe klasy dokumentów dostępny jest jeden krój szeryfowy, jeden bezszeryfowy i jeden grotesk. Oczywiście domyślnie są to Computer Modern Roman, Computer Modern Sansserif i Computer Modern Typewriter.

Oto podstawowe polecenia wyboru krojów i odmian w tym schemacie:

Krój pisma (rodzina)	
`\textrm{...}` lub `\rmfamily`	krój szeryfowy
`\textsf{...}` lub `\sffamily`	krój bezszeryfowy
`\texttt{...}` lub `\ttfamily`	`grotesk`
Grubość i szerokość pisma	
`\textmd{...}` lub `\mdseries`	pismo jasne
`\textbf{...}` lub `\bfseries`	**pismo grube**
Odmiana pisma	
`\textup{...}` lub `\upshape`	odmiana prosta
`\textit{...}` lub `\itshape`	*kursywa*
`\textsl{...}` lub `\slshape`	*odmiana pochyła*
`\textsc{...}` lub `\scshape`	Kapitaliki

Jak widać, tworzą one pary poleceń o podobnej funkcjonalności. Wersje z argumentem składają ten argument pismem wybranym przez swój odpowiednik bez argumentu. Ponadto polecenia te na granicy między pismem pochyłym a prostym dodają kompensatę wychylenia ostatniego znaku (*italic correction*), chyba że następnym znakiem jest kropka lub przecinek i kompensata nie jest potrzebna.

Polecenia bez argumentów można sobie wyobrażać jako zdefiniowane wg schematu:

```
\newcommand\rmfamily{
  \fontfamily{\rmdefault}\selectfont}
```

One również nie mają zaszytej w sobie nazwy rodziny, odmiany, grubości, ale odwołują się do makr przechowujących wartości domyślne. Oczywiście domyślną wartością makra `\rmdefault` jest `cmr`, czyli polecenie `\rmfamily` domyślnie włącza krój Computer Modern Roman.

Stąd płynie wniosek, że aby w LaTeXu 2_ε złożyć dokument, używając jako kroju szeryfowego Timesa, wystarczy:

```
\renewcommand\rmdefault{ptm}
```

(Polecenia zapożyczone z LaTeXa 2.09 działają w podobny, pośredni sposób, więc również konsekwentnie poddają się takiemu przedefiniowaniu.)

Wartości domyślne wkraczają jeszcze w kilku innych miejscach.

Zasadnicza treść dokumentu jest składana fontem uaktywnianym poleceniem `\normalfont`. Jest ono mniej więcej równoważne

```
\usefont{\encodingdefault}
  {\familydefault}
  {\seriesdefault}{\shapedefault}
```

Płyną z tego ciekawe wnioski:

1) Zmieniając wartość `\encodingdefault` na `OT4` możemy sprawić, że dokument będzie składany polskimi fontami. Pakiet `polski` tak właśnie postępuje, przełączając układ na `OT4` lub `T1` w zależności od opcji.

2) Standardową wartością `\familydefault` jest `\rmdefault`. Skład całego dokumentu czcionką bezszeryfową można uzyskać zmieniając tę wartość na `\sfdefault`:

```
\renewcommand\familydefault{\sfdefault}
```

Tego rodzaju zmian należy dokonywać w preambule dokumentu (to jest przed `\begin{document}`).

Konfigurowanie NFSS. Pozostaje jeszcze odpowiedzieć na pytanie, skąd NFSS „wie", jakie kroje, odmiany i stopnie pisma są dostępne i w jakich układach.

Deklaracja układów fontów powoduje, że LaTeX ładuje pliki opisujące dane układy. Np. dla układu `OT4` będzie to plik `ot4enc.def`[2]. W tym pliku zawarte są informacje, jakie akcenty są dostępne w danym układzie, w jakich pozycjach się one znajdują i jakie są dostępne znaki akcentowane (kombinacje akcent-litera).

Pakiet `polski.sty` wewnętrznie ładuje odpowiednią definicję układu.

Opisy krojów są ładowane automatycznie z plików `.fd` przy pierwszej próbie użycia fontu z danej rodziny w danym układzie. Na przykład po zadeklarowaniu jako domyślnego układu `OT4`, pierwsza próba włączenia fontu z rodziny `cmr` spowoduje, że LaTeX załaduje plik `ot4cmr.fd`. Oto fragment tego pliku:

```
\DeclareFontFamily{OT4}{cmr}{}
\DeclareFontShape{OT4}{cmr}{m}{n}
   {  <5> <6> <7> <8>
          <9> <10> <12> gen * plr
      <10.95> plr10
      <14.4>  plr12
      <17.28><20.74><24.88>plr17}{}
 ...
```

2: Od grudnia 1996 nazwy wszystkich plików w dystrybucji LaTeXa 2_ε są pisane tylko małymi literami. Fakt ten wywołał niejakie zamieszanie, ale wydaje się, że na dłuższą metę jest to decyzja rozsądna.

```
\DeclareFontShape{OT4}{cmr}{bx}{it}
   { <-10>sub * cmr/bx/n
     <10> <10.95> <12> <14.4> <17.28>
          <20.74> <24.88> plbxti10
     }{}
...
```

Pierwsza deklaracja powiadamia, że jest to opis rodziny cmr w układzie OT4. Dalej następuje ciąg deklaracji `\DeclareFontShape` opisujących poszczególne kombinacje rodzina/grubość/odmiana. Pierwsza opisuje odmianę jasną (m *medium*), prostą (n *normal*) – dla stopni 5, 6, 7, 8, 9, 10 i 12 należy użyć fontu o nazwie plikowej powstającej z ciągu `plr` po dołączeniu doń liczby stopnia pisma. Dla stopnia 10.95pt należy użyć fontu `plr10 at 10.95pt`, podobnie dla stopni 14.4, 17.28, 20.74, 24.88 użyty zostanie odpowiednio zeskalowany font `plr12` i `plr17`. Druga klauzula `\DeclareFontShape` opisuje grubą (bx *bold extended*) kursywę (it *italic*) – dla stopni mniejszych od 10pt należy użyć w zastępstwie grubej prostej odmiany rodziny `cmr`. Dla wymienionych dalej stopni użyty zostanie odpowiednio zeskalowany font `plbxti10`.

Deklaracje układów OT1 i T1 oraz opis rodziny cmr w tych układach są wbudowywane w format razem z plikami metrycznymi podstawowych fontów. Służy to przyspieszeniu ładowania LaTeXa na początku przetwarzania dokumentu. Zestaw wbudowanych układów, rodzin i plików metrycznych można zmienić przy pomocy plików konfiguracyjnych `fonttext.cfg` i `preload.cfg` – szczegóły opisano w dokumencie `cfgguide.tex`.

Aby można było powiedzieć, że font jest zainstalowany w LaTeXu 2$_\varepsilon$ musi istnieć dla niego definicja układu i definicja kroju (rodziny) w tym układzie. Układów w zasadzie jest skończona niewielka liczba i definicje większości z nich są zawarte w dystrybucji LaTeXa (w tym OT4). Natomiast dla każdego kroju potrzebny jest jego opis w pliku o nazwie zbudowanej z nazwy układu i nazwy rodziny z rozszerzeniem `.fd` (wszystko małymi literami). Pliki te muszą znajdować się w katalogu widocznym dla LaTeXa.

Pliki opisu dla fontów PL i PC są zawarte w pakiecie PLaTeX. Pliki opisu dla kroju Quasi Palladio (qpl), Quasi Times (qtm) i antykwy toruńskiej (antt) są częścią pakietu `pl-qx` Piotra Kłosowskiego.

◇ Marcin Woliński
wolinski@melkor.mimuw.edu.pl

Wykorzystanie polskich fontów postscriptowych w LaTeX-u

Piotr Kłosowski

Wstęp

W artykule będę się zajmował polskimi fontami postscriptowymi, tzn. fontami które posiadają „polskie litery” i polski cudzysłów otwierający jako osobne znaki. Wykluczam z tego towarzystwa fonty, w których te specyficzne znaki konstruuje się przez nakładanie osobnych elementów na poziomie fontów wirtualnych lub makr TeX-owych.

Pisząc LaTeX myślę o LaTeX2ε. Poprzednik tego formatu, długo panujący LaTeX 2.09 chyba definitywnie odchodzi w przeszłość. To dobrze; jego mechanizmy obsługi fontów były nieprzyjazne[1].

Zakładam więc, że adresat moich dywagacji ma dostęp do TeX-a z formatem LaTeX2ε i sterownikiem `dvips` i że to oprogramowanie jest prawidłowo zainstalowane i skonfigurowane. Zakładam też, że LaTeX jest spolszczony przy pomocy pakietu PLaTeX oraz niepostscriptowych fontów PL (autorzy: Bogusław Jackowski i Marek Ryćko). a efekty pracy można oglądać na dostępnym urządzeniu postscriptowym. (Nie musi to być naświetlarka czy drukarka z wbudowanym PostScript-em. Świetnie spisze się interpreter Ghostscript.)

Po co, mając to wszystko, miałby leniwy użytkownik TeX-a wchodzić na grząski grunt eksperymentów z nowymi rodzajami fontów? Może jego artykuł będzie częścią zbiorówki składanej czcionką Times, może potrzebuje liter z czarną obwódką i białym wypełnieniem, a może znudził się krojem Computer Modern i pozazdrościł produkcjom kolegi uzbrojonego w MS Worda? Nie do końca wierzę w jego lenistwo. Jednak by go nie drażnić, przenoszę bardziej szczegółowe dygresje (nie przed wszystkimi udało mi się powstrzymać) do przypisów.

1: Obawiam się jednak, że w gronie adresatów mojej pisaniny może być znacząca grupa, która siłą inercji czy z przywiązania do narosłych latami kolekcji makr zostaje przy „starym LaTeX-u”. Żeby całkiem ich nie zignorować, mogę im polecić pakiecik `pslatex`. Może znajdzie się gdzieś archiwach w okolicach katalogu `macros/latex209/contrib`.

Jakie fonty są dostępne?

Jak dotąd, zostały stworzone i publicznie udostępnione cztery polskie fonty postscriptowe:

- Fonty PL – obwiedniowa wersja tradycyjnego spolszczenia fontów Computer Modern przez B. Jackowskiego i M. Ryćko. Autor: Janusz Marian Nowacki [9].
- Antykwa Toruńska – adaptacja komputerowa oryginalnego polskiego kroju akcydensowego autorstwa Zygfryda Gardzielewskiego. Autor: J. M. Nowacki [10].
- QuasiPalladio – spolszczenie kroju Palladio firmy URW, krój bardzo podobny do popularnego kroju Palatino (QuasiPalladio zastępuje cenionego Platana). Autor B. Jackowski.
- QuasiTimes – spolszczenie kroju Nimbus Roman No 9 firmy URW, krój typu Times New Roman. Autor B. Jackowski.

Wszystkie te fonty są przygotowane do użycia z systemem TEX (gotowe pliki metryczne `*.tfm`) oraz ze sterownikiem dvips (dołączone pliki przekodowania `*.enc` i `psfonts.map`)[2]. Można je znaleźć w archiwach CTAN oraz GUST w katalogu `fonts/psfonts/polish`

Mam nadzieję, że wysiłki obu zasłużonych fontologów nie ustaną. Chodzą słuchy o trwających w Grudziądzu pracach nad komputerową Antykwą Półtawskiego. Ten oryginalny polski krój zaprojektowany w 1928 r. był bardzo popularny w epoce przedkomputerowej. Nie chce mi się też wierzyć, żeby rodzina quasi-fontów skończyła się na dwóch krojach. Może teraz coś bezszeryfowego... A może ktoś trzeci dołączy do silnej grupy? Kto wie?

Oprócz wymienionych, w sprzedaży znajdują się różne polskie kroje postscriptowe. Niestety poziom cenowy odpowiada tu mniej więcej jakości technicznej i wizualnej. Popularnością cieszą się bardzo tanie fonty dołączane do programu graficznego Corel Draw. Droższe i mniej krytykowane są fonty sprzedawane przez warszawską firmę Theta[3]. Wyżej wymienione fonty są przeznaczone do użytku w systemie MS Windows i ich wykorzystanie w TEX-u wymaga wysiłku. (W przypadku braku plików `*.afm` gra w ogóle wydaje się być nie warta świeczki.) Pomimo, że istnieje pakiet TOIL B. Jackowskiego [4], który ułatwia zadanie, moim zdaniem nie jest to droga dla leniwych.

Jak dołączyć do instalacji TEX-owej nowy font postscriptowy?

Niestety lenistwo będzie tu wystawione na próbę. Nie da się wykonać tej operacji bez choćby minimalnej wiedzy o używanej instalacji TEX-owej. Jeżeli masz zainstalowanego emTEX-a lub teTEX-a, to jesteś w lepszej sytuacji. To popularne dystrybucje i znane mi z praktyki, stąd przykłady następujące w dalszym ciągu dotyczą ich obu[4].

Fonty dodaje się do systemu TEX według jednolitego schematu różniącego się jednak w szczegółach zależnych od dystrybucji.

I. Program TEX używa wyłącznie plików metrycznych `*.tfm`. Niezbyt precyzyjnie mówiąc, nie zawierają one informacji o kształcie znaków tylko o ich wymiarach[5]. Jest tam wszystko czego TEX potrzebuje i w gruncie rzeczy TEX-owi wszystko jedno czy kształty liter są opisane mapą bitową (pliki `*.pk`) czy obwiednią (pliki `*.pfb`).

Jeżeli chodzi o pliki `*.tfm` dla postscriptowych fontów PL, to w ogóle nie ma kłopotu. Zgodnie z założeniami sformułowanymi we Wstępie, mamy je już w naszym systemie dobrze umieszczone (to te same, które obsługują bitmapowe fonty PL wygenerowane z metafontowych źródeł).

2: Warto zauważyć, że trzy ostatnie fonty są również przygotowane do współpracy z programem ATM (a tym samym z systemami MS Windows i OS/2), to znaczy są do nich dołączone pliki metryczne `*.pfm`. Co prawda priorytety TEX-owe dominują i rozmieszczenie znaków w foncie, tzw. układ QX, różni się od Windowsowego, ale jest w miarę możliwości do niego zbliżone [13].

3: Fonty, które nie byłyby krytykowane znajdują się raczej poza zasięgiem finansowym nieprofesjonalisty. Zainteresowani mogą zapoznać się z opiniami i informacjami Adama Twardocha [12].

4: Od pewnego czasu propagowany jest standardowy układ katalogów dystrybucji TEX-owych – TDS. Ponieważ teTEX jest zgodny z tym standardem sądzę, że przykłady dotyczące go można próbować uogólniać.

5: Więcej o plikach metrycznych (i o organizacji fontów w TEX-u) można się dowiedzieć z następujących pozycji, które podaję w kolejności od najprzystępniejszej do najbardziej wyczerpującej: [3], [5] oraz [7]

Dla pozostałych trzech fontów dobrze jest utworzyć dla porządku oddzielne katalogi i umieścić tam odpowiednie pliki `*.tfm`, np. w emTEX-u:

```
emtex\tfm\qfonts
emtex\tfm\antt
```

a w teTEX-u:

```
texmf/fonts/tfm/public/qfonts
texmf/fonts/tfm/public/antt
```

To powinno wystarczyć[6]. Uwaga: po każdej zmianie w drzewie katalogów teTEX-a należy pamiętać o uruchomieniu z odpowiednimi uprawnieniami skryptu `texhash` by uaktualnić indeks `ls-R`.

Sprawdźmy teraz czy TEX widzi świeżo doinstalowaną Antykwę Toruńską. Można w tym celu skompilować np. trzylinijkowy plik o treści:

```
\font\antykwaxii = anttr at 12dd
\antykwaxii Ala ma kota
\bye
```

Ja lubię w takich przypadkach posługiwać się plikiem `testfont.tex`, będącym na standardowym wyposażeniu METAFONT-a.

II. Jeżeli nasz plik testowy składa się gładko to teraz trzeba jeszcze skonfigurować sterownik `dvips`.

1. Najpierw umieśćmy pliki `*.pfb` i `*.enc` w katalogu, w którym `dvips` ich oczekuje. Może to być `c:\psfonts` jeżeli współdzielimy katalog fontów postscriptowych między TEX-a i windowsowego ATM, albo `texmf/fonts/type1/public/antt` jeżeli nasza instalacja jest zgodna z TDS, albo inne logiczne miejsce. Czy `dvips` znajdzie tam nowe fonty? Jeżeli w `c:\psfonts` były już wcześniej inne fonty postscriptowe, a w `texmf/fonts/type1` były już wcześniej inne podkatalogi z nimi i były widziane przez `dvips`-a, to mamy duże szanse, że tak.

Program `dvips` szuka plików potrzebnych do pracy według wpisów do plików konfiguracyjnych albo zgodnie z zawartością zmiennych systemowych[7].

Przejdźmy do konkretów zależnych od dystrybucji. W emTEX-u zacząłbym od skontrolowania zawartości zmiennej `DVIPSHEADERS` wskazującej na położenie tak zwanych plików nagłówkowych – pliki fontowe `*.pfb` i pliki przekodowania `*.enc` są tak traktowane. Jeżeli zmienna zawiera ścieżki, to są one obowiązujące. Jeżeli nie, to musimy poszukać głównego pliku konfiguracyjnego `config.ps`[8]. Typowe położenie jest w katalogu `emtex\data\dvips` (może być inne, bo `dvips` nie należy ściśle rzecz biorąc do dystybucji emTEX). Wewnątrz pliku powinna się znaleźć linijka zaczynająca się literą „H”, zawierająca listę ścieżek do plików nagłówkowych. Może to być coś podobnego do:

```
* PostScript headers and fonts
H c:\psfonts;$(EMTEXDIR)\data\dvips;.\lofo;.
```

Notacja jest chyba intuicyjnie zrozumiała. Gwiazdka jest znakiem komentarza, a `$(EMTEXDIR)` rozwinie się w położenie szczytu drzewa TEX-owego. Warto mieć możliwość korzystania ze specjalnych fontów lokalnie, tylko na potrzeby konkretnej publikacji. Dla porządku, zamiast wrzucać je do katalogu bieżącego razem z tekstem zwykle tworzę specjalny podkatalog `lofo`.

2. Przy okazji sprawdźmy, czy `dvips` wie gdzie są nasze pliki metryczne. Odpowiednia ścieżka powinna być w `config.ps` w linijce zaczynającej się od „T”, na przykład jakoś tak:

```
* Paths
T $(EMTEXDIR)\tfm;.\localtfm
```

6: W emTEX-u pliki `*.tfm` poszukiwane są domyślnie w podkatalogach katalogu `tfm` (tylko pierwszy poziom zagłębienia!). Dokładniej, o początkowym katalogu drzewa tfm-ów decyduje zmienna `TEXTFM`, a jeżeli nie ma ona nadanej wartości, to domyślnie jest przyjmowana lokalizacja `$EMTEX:tfm`. W teTEX-u przeszukiwane są domyślnie dowolnie zagłębione podkatalogi `fonts/tfm`. Dokładniej, decyduje o tym linijka zaczynająca się od `TFMFONTS` w pliku `texmf.cnf`.

7: Jest jeszcze trzecia możliwość: ścieżki mogą być wkompilowane w program jako domyślne. Jest także czwarta: w linii wywołania komendy można przez podanie opcji decydować, które pliki konfiguracyjne będą wczytywane. Pełna informacja o opcjach i sposobach konfiguracji znajduje się w dokumentacji [11].

8: Obraz komplikuje fakt, że mogą być jeszcze inne zbiory konfiguracyjne (celowe przy liczbie drukarek lub użytkowników większej niż jeden, np. instalacje sieciowe) wczytywane po głównym i zmieniające jego ustawienia [11]. Jeżeli są kłopoty z rozpoznaniem sytuacji, to rozwiązaniem może się okazać ustawienie prywatnej zmiennej `DVIPSHEADERS` co przykryje działanie zbiorów konfiguracyjnych.

3. Na koniec musimy uzupełnić istniejący w naszej instalacji plik `psfonts.map` o zawartość pliku o tej samej nazwie znajdującego się w dystrybucji spolszczonych fontów postscriptowych. Plik `psfonts.map` jest bardzo ważny. Dzięki niemu dvips wie jakimi fontami dysponuje, czy są to fonty drukarkowe czy dyskowe i czy ma je poddać przekodowaniu lub transformacji. Więcej szczegółów w [5] i [11]. Przy jego modyfikacji warto zachować ostrożność: mimo, że czytelny, jego struktura jest dość ściśle określona i dla początkującego niekoniecznie intuicyjna. Przed jego edycją warto zrobić kopię (to ogólnie pożyteczy zwyczaj – leniwi, ubezpieczajcie się przed koszmarem wielu godzin spędzonych nad doprowadzeniem waszego komputera do stanu używalności po, wydawało by się, niewinnym eksperymencie).

W teTEX-u informacje o podkatalogach zostały usunięte z pliku `config.ps` i scentralizowane w pliku `texmf.cnf`. Ma on nieco zawiłą strukturę, ale linijka[9]:

```
DVIPSHEADERS=$KPSE_DOT:\
 $TEXMF/{dvips,fonts/type1}//
```

mówi, z grubsza biorąc, że gdy umieścimy nasze fonty w katalogu bieżącym (bałaganiarstwo – nie polecam) albo jakimś podkatalogu `texmf/dvips` czy `texmf/fonts/type1`, to powinny być znalezione (oczywiście pod warunkiem uruchomienia wcześniej wspomnianego skryptu rejestrującego zmiany w instalacji `texhash`)[10]. Plik `psfonts.map` można znaleźć w `/dvips/base`.

Jak składać nowym fontem?

Nowy Sposób Wyboru Fontów (NFSS). Zanim zacznę mówić o LATEX-owym mechanizmie obsługi fontów, NFSS, muszę się zastrzec, że powiem o nim tak mało, jak to tylko możliwe. Ma dość skomplikowane wnętrze, choć używa się go prosto i naturalnie[11].

W LATEX-u bieżący font charakteryzuje pięć elementów, z których trzy (grubość i szerokość, odmiana, stopień i interlinia) nie będą nas interesować, bo ich użycie w przypadku fontów postscriptowych jest standardowe. Dwa pozostałe elementy, układ fontu i rodzina (krój), są ważne i zatrzymamy się chwilę przy nich.

Układ mówi o tym jakie znaki, z tych wprowadzanych za pomocą makr, są dostępne i jak ten dostęp jest realizowany. Na przykład litera „ł” wprowadzona do tekstu jako `\l` jest osiągalna w układzie OT1 jako nałożenie znaku o numerze 32 i litery l, w układzie OT4 i T1 jako znak o numerze 170. W układzie OT2 takiej litery nie ma, a odpowiednie makro nie jest zdefiniowane. Plik zawierający informacje tego rodzaju, tzn. defnicje odpowiednich makr, ma nazwę xxxenc.def, gdzie zamiast xxx należy wpisać nazwę właściwego układu. Jest on, w gruncie rzeczy, definicją układu. Tu uwaga: w nazwach układów zwyczajowo używamy dużych liter, jednak jeżeli występują w nazwach plików, to piszemy je małymi literami.

LATEX jest poinformowany bez naszego udziału o układach OT1 (standardowy układ fontów CM) i T1 („europejski” układ fontów EC). Inne układy używane w składanym dokumencie powinny być zadeklarowane w jego preambule, jako opcje wywołania pakietu `fontenc`, np.:

```
\usepackage[OT2, OT4]{fontenc}
```

Deklaracja układu może być też ukryta wewnątrz innego wywoływanego w preambule pakietu, który zarządza językami lub fontami. Tak deklaruje układ OT4 („polski” układ fontów PL) `polski.sty` z pakietu PLATEX. Domyślnym układem jest ostatni zadeklarowany. Przełączanie pomiędzy układami realizuje sekwencja komend:

```
\fontencoding{OT2}\selectfont
```

Krój pisma (rodzina) to grupa fontów o wspólnej nazwie. To nieprecyzyjne określenie jest adekwatne do nieostrej sytuacji. Jedna rodzina fontów może występować w kilku układach, może zawierać różną liczbę fontów różniących się pozostałymi

9: Kreskę „\” na końcu pierwszego wiersza trzeba usunąć jeżeli całość zostanie zapisana w jednej linijce. Tutaj wpis jest dwuwierszowy z uwagi na wąską szpaltę

10: Do zdiagnozowania sytuacji możemy użyć komend, które rozwijają tajemnicze zmienne z pliku `texmf.cnf`, tzn.:
`kpsepath dvips_header`
`kpsewhich dvips_header anttr.ps`

11: Zainteresowanych szczegółami polecam artykuł Marcina Wolińskiego w tym numerze i dokumentację z dystrybucji LATEX-a [8].

cechami[12]. LaTeX poszukuje informacji o fontach tworzących rodzinę w danym układzie w plikach `xxxyyy.fd`, gdzie `xxx` jest nazwą układu a `yyy` nazwą rodziny. Są w nich zawarte przyporządkowania pomiędzy kombinacjami pięciu cech fontów, którymi operuje LaTeX a nazwami plików dyskowych zawierającymi fonty o tych cechach. Informacje o rodzinie `cm` są już umieszczone w formacie. Inne są czytane z dysku przy pierwszym przełączeniu na font nieznanej rodziny. Takie przełączenie realizuje sekwencja `\fontfamily{yyy}\selectfont`.

Reasumując, tak jak plik `xxxenc.def` jest definicją układu `xxx`, tak plik `xxxyyy.fd` jest definicją rodziny w tym układzie. Żeby doinstalować nowy font do LaTeX-a trzeba:

- wymyślić dla niego nazwę rodziny;
- jeżeli układ fontu jest nietypowy, wymyślić dla niego nazwę;
- stworzyć i umieścić w ścieżce przeszukiwań LaTeX-a plik definicji nowej rodziny `*.fd`, który powiąże pliki dyskowe fontów z instrukcjami LaTeX-a umożliwiającymi przełączenie bieżącego fontu składu;
- w razie potrzeby, stworzyć i odpowiednio umieścić plik definicji nowego układu `*.def`, który powiąże znaki nie dające się wprowadzić bezpośrednio z klawiatury z odpowiednimi instrukcjami LaTeX-a;
- wyprodukować ściągawkę z nazwami niestandardowych instrukcji i informacją, które ze standardowych nie działają w naszym foncie[13].

Jeżeli chodzi o Antykwę Toruńską, QuasiPalladio i QuasiTimesa, to potrzebne pliki można znaleźć w pakiecie pl-qx w archiwum GUST w podkatalogu `fonts/psfonts/polish`. Pozostaje je umieścić w jakimś widzianym przez LaTeX-a miejscu. W emTeX-u może to być katalog np. `emtex/texinputs/pl-qx`, w teTeX-u np. `texmf/tex/latex/pl-qx`. Katalog bieżący jest miejscem dobrym do prób, ale na dłuższą metę takie umiejscowienie plików systemowych może prowadzić do zamieszania.

12: Być może dla niektórych czytelnikow zaskoczeniem będzie informacja, że pliki o nazwach `cmbx12`, `plbx12` i `ecbx1200` zawierają fonty należące do tej samej rodziny `cm` a różniące się jedynie układem (odpowiednio `OT1`, `OT4` i `T1`).

13: Zachęcam do tego. Nasza pamięć jest na ogół słabsza niż sądzimy.

W zasadzie fonty możemy teraz uważać za zainstalowane i gotowe do użycia.

Polskie fonty postscriptowe mają znaki rozmieszczone według nowego schematu nazwanego przez autorów QX[14]. Zawiera on prawie wszystkie znaki fontów PL na tych samych pozycjach i jeszcze o wiele więcej[15]. Ma to ważne konsekwencje. Dopóki nie potrzebujemy tych dodatkowych znaków, możemy o nich zapomnieć i stosować LaTeX-owy układ OT4 włączony na ogół w preambule w wierszu:

```
\usepackage{polski}
```

Jeżeli zechcemy je wykorzystać, musimy zadeklarować układ LQX zdefiniowany w pliku `lqxenc.def` zawierający defnicje makr z pliku `ot4enc.def` oraz inne, pozwalające wprowadzać do tekstu znaki układu QX nieobecne w fontach PL[16].

```
\usepackage[LQX]{fontenc}
\usepackage{polski}
```

Rodzinom polskich fontów postscriptowych zostały nadane nazwy LaTeX-owe `antt`, `qpal` i `qtim`. Chcąc zrobić wstawkę w tekście kursywą kroju Times można na przykład napisać:

```
Ala ma {%
\fontencoding{LQX}%
\fontfamily{qtm}%
\fontshape{it}%
\selectfont
kota}
```

Jeżeli mamy do złożenia Timesem cały artykuł rozsądnie jest przedefiniować różne parametry domyślne. Żeby nie wnikać w dość pogmatwane na pierwszy rzut oka zależności, proponuję dołączyć w preambule odpowiedni styl z pakietu pl-qx. Dla Timesa będzie to po prostu:

14: QX jest propozycją układu znaków w spolszczonym foncie postscriptowym wygodnym dla zastosowań TeX-owych a do przyjęcia dla zastosowań windowsowych. Powstał jako efekt długich dyskusji i przemyśleń i miejmy nadzieję, że stanie się standardem [13].

15: Co znaczy to prawie? Jest kilka różnic, które nie powinny się ujawnić w typowych zastosowaniach.

16: Można pójść dalej i wykorzystać fakt, że o ile układ LQX jest rozszerzeniem OT4, tak OT4 jest rozszerzeniem OT1. Jeżeli chcemy złożyć Antykwą Toruńską sonet Szekspira w oryginale, to nie musimy dołączać w preambule stylu `polski`.

```
\usepackage{polski}
\usepackage{qtimes}
\usepackage[LQX]{fontenc}
```

Uwaga, kolejność wywoływania pakietów nie jest dowolna, ale podobnie jak zagadnienie parametrów domyślnych nadaje się raczej na forum Dociekliwych Użytkowników LaTeX-a. I druga uwaga: przy powyższej preambule nie zadziałają standardowe komendy zmiany rodziny LaTeX-a `\textsf` i `\ttfamily`. Po prosty nie istnieją, póki co, fonty bezszeryfowy i maszynopodobny w układzie LQX. W razie potrzeby trzeba najpierw zmienić układ:

```
Ala ma {\fontencoding{OT4}%
\fontfamily{cmss}%
\selectfont
kota}
```

Co jeszcze by można, a czego nie należy

Teraz, kiedy już wszystko mniej więcej działa, możemy pomarudzić. Chociażby, że nasze polskie fonty postscriptowe nie mają odmiany pochylonej i kapitalików. W końcu w LaTeX-u to standard. Szczególnie zdziwią się ci, którzy wiedzą, że mimo braku zaprojektowanych odmian fontu można je dorobić np. programem afm2tfm [11]. Jednak jakość typograficzna tak sztukowanych fontów jest dyskusyjna. Nie trzeba mieć specjalnie wyczulonego oka, żeby rozpoznać komputerowo generowane kapitaliki po zbyt grubych elementach dużych liter. Moim zdaniem, można używać tego rodzaju protez w uzasadnionych przypadkach, zachowując ostrożność i krytycyzm.

Używanie polskich fontów postscriptowych różnych niż PL/EC czasami prowadzi do problemów. Chodzi o sytuacje, kiedy łączymy podstawowy font tekstowy z wstawkami po rosyjsku, grecku lub wzorami matematycznymi. Potrzebne są wtedy optycznie zgodne z fontem podstawowym fonty zawierające znaki z tych alfabetów czy symbole matematyczne. Specjalistyczne czcionki fonetyczne z pakietu TIPA wyjątkowo posiadają odmianę, którą można łączyć z czcionką Times. W przypadku składania fontem QuasiPalladio, jako fontu matematycznego można wykorzystać rodzinę fontów Euler. Są to jednak wyjątki, generalnie będziemy mieli problemy i zapewne wrócimy do fontów PL/EC lub będziemy zmuszeni kupić fonty komercyjne.

Literatura

[1] Archiwum listy dyskusyjnej GUST marzec (?) 1998 r.

[2] Hoenig Alan: *Fonty wirtualne, fonty wirtuoza*, „GUST" nr 9 (1997).

[3] Jackowski Bogusław, Wawrykiewicz Stanisław: *Fontowe ABC* [w:] „GUST" nr 9 (1998);

[4] Jackowski Bogusław: *Pakiet TOIL czyli oswajanie fontów PostScript-owych*, [w:] „GUST" nr 9 (1997).

[5] Jackowski Bogusław: *Korzystanie z fontów PostScriptowych w dokumentach TeX-owych*, Bachotek 1997.

[6] Jackowski Bogusław i Marek Ryćko, *Polerowanie TeX-a: od systemu gotowego do użycia do systemu wygodnego w użyciu*, [w:] „GUST" nr 1 (1993).

[7] Knuth Donald E.: *The* METAFONT*book*, Addison--Wesley, 1986

[8] *LaTeX2ε font selection*, jako fntguide.tex, jeden z plików dokumentacji wchodzącej w skład dystrybucji LaTeX2ε.

[9] Nowacki Janusz M.: *Fonty PL w nowej PostScriptowej Postaci*, [w:] „GUST" nr 9 (1997).

[10] Nowacki Janusz M.: *Antykwa Toruńska – od początku do końca polska czcionka*, [w:] „GUST" nr 9 (1997).

[11] Rokicki Tomas: dvips*: A DVI-to-PostScript Translator.*

[12] Twardoch Adam: *Re: Kto robi polskie fonty ATM?*, Archiwum Listy Dyskusyjnej GUST, 19 maj 1998 r.

[13] *Układ QX – kto zacz?*, [w:] „GUST" nr 9 (1997).

◇ Piotr Kłosowski
pikl@dns.lbrsa.com.pl