

Przewodnik po T_EX Live, wydanie 7

Sebastian Rahtz

sebastian.rahtz@oucs.ox.ac.uk

Czerwiec 2002

Spis treści

1. Wstęp	2
1.1. Rozszerzenia T _E X-a	3
2. Struktura i zawartość CD-ROM-u	3
2.1. Pakiety i ich zestawy	4
3. Instalacja i użytkowanie pod Unixem	7
3.1. Wstępne kroki przed instalacją dla systemu MacOSX	7
3.2. Uruchamianie T _E X Live z CD-ROM-u	9
3.3. Instalacja T _E X Live na twardym dysku	10
3.4. Instalacja pojedynczych pakietów z T _E X Live na twardym dysku	13
3.5. Program texconfig	14
4. Instalacja i użytkowanie pod Windows	15
4.1. Program TeXLive.exe	15
4.2. Uruchamianie T _E X Live z CD-ROM-u	16
4.3. Instalacja edytorów i innych programów pomocniczych	18
4.4. Instalacja na twardym dysku	18
5. Zarządzanie systemem T_EX Live w Windows	23
5.1. Różnice w stosunku do standardowego Web2c w systemach Win32	23
5.2. Dodawanie pakietów do instalacji	24
5.3. Usuwanie T _E X Live z dysku twardego	24
5.4. Uruchamianie TeXSetup z linii poleceń	24
5.5. Instalacja sieciowa	26
5.6. Konfiguracja indywidualna	26
5.7. Testowanie	28
5.8. Drukowanie	29
5.9. Więcej na temat WinShell	29
5.10. Wskazówki dla użytkowników Win32	31
5.11. W razie problemu	34
5.12. Kompilacja plików źródłowych	34

6. Budowa systemu na nowej platformie Unixowej	35
6.1. Warunki wstępne	35
6.2. Konfiguracja	35
6.3. Uruchamianie make	35
6.4. Końcowe kroki konfiguracyjne	36
7. Instrukcja obsługi systemu Web2c	36
7.1. Przeszukiwanie ścieżek przez Kpathsea	38
7.2. Bazy nazw plików	42
7.3. Parametry kontrolujące działanie programów	50
8. Podziękowania	50
9. Historia	51
10. Przyszłe wersje	52
11. Plik konfiguracyjny texmf.cnf	53

Spis tabel

1. Typy plików Kpathsea	43
-------------------------	----

1. Wstęp

Niniejsza dokumentacja opisuje CD-ROM **T_EX Live 7** – dystrybucję T_EX-a/L^AT_EX-a dla systemów Unix, Linux, MacOSX oraz Windows32, zawierającą programy T_EX, L^AT_EX 2_ε, METAFONT, METAPOST, Makeindex, i BibT_EX oraz bogaty zestaw pakietów makr o wielorakim zastosowaniu, fonty i dokumentację. Całość jest zgodna ze *Standardem Katalogów T_EX-owych* (ang. *T_EX Directory Standard* – TDS), wobec tego może być używana prawie z każdą instalacją T_EX-a. W tej edycji na jednym CD-ROM-ie zmieściły się wszystkie przygotowane pliki, wspólne dla wielu platform, zaś same programy jedynie dla trzech systemów: Windows, Linux i MacOSX. Zestawy programów dla wielu innych systemów uniksowych będą zawarte na drugim krążku lub będą one dostępne w sieci.

Pakiet T_EX-owy zawiera programy w implementacji Web2c (wersji 7.3.7), ułatwiającej ich elastyczne uruchamianie; wykorzystuje ponadto zalety wydajnej i konfigurowalnej biblioteki Kpathsea, autorstwa Karla Berry’ego i Olafa Webera. Programy można uruchamiać zarówno bezpośrednio z CD-ROM-u, jak i po zainstalowaniu na twardym dysku.

Większość systemów uruchamialnych z CD-ROM-u zawiera obszerny zestaw sterowników i programów pomocniczych dla T_EX-a, takich jak dvips (sterownik PostScriptowy), dvi2pdf (konwerter DVI do PDF), xdvi (przeglądarka dla X Window), dvilj (sterownik dla drukarek HP LaserJet), lacheck (sprawdzanie składni do L^AT_EX-a), tex4ht (konwerter T_EX do HTML), dviconcat i dviselect, dv2dt i dt2dv (konwerter dvi do ASCII i z powrotem), oraz programy użytkowe dla PostScript-u psutils, autorstwa Angus Duggana.

1.1. Rozszerzenia T_EX-a

Systemy uruchamialne z płyty **T_EX Live** zawierają trzy rozszerzenia standardowego T_EX-a:

1. ε -T_EX, który dodaje niewielki, lecz silny zestaw nowych poleceń wbudowanych, oraz rozszerzenie T_EX--X_ET dla składu od prawej do lewej. W trybie domyślnym ε -T_EX jest w 100% zgodny ze zwykłym T_EX-em. Więcej szczegółów znaleźć można na CD-ROM-ie, w [texmf/doc/etex/base/etex_man.pdf](#).
2. pdfT_EX, który opcjonalnie może zapisywać w formacie Acrobat PDF, zamiast DVI. Podręcznik użytkownika znajduje się w pliku [texmf/doc/pdftex/pdftex-1.pdf](#). Plik [texmf/doc/pdftex/samplepdf/samplepdf.tex](#) pokazuje przykład wykorzystania. L^AT_EX-owy pakiet hyperref posiada opcję „pdftex”, która uaktywnia dostęp do wszystkich cech programu.
3. Ω (Omega), która pracuje wewnętrznie ze znakami kodowanymi 16-bitowo (Unicode); pozwala bezpośrednio pracować jednocześnie z większością tekstów spotykanych na świecie. Wspomaga ona także dynamicznie ładowane tzw. „procesy tłumaczenia Ω ” (OTP), co pozwala użytkownikowi definiować złożone transformacje wykonywane na dowolnych strumieniach wejściowych. Więcej szczegółów znaleźć można na CD-ROM-ie: [texmf/doc/omega/base/doc-1.8.tex](#) (dokumentacja niezbyt aktualna).

ε -T_EX jest stabilny (wersja 2.1), chociaż nowsze wersje dodadzą pewnie nieco więcej funkcjonalności. pdfT_EX (wersja 1.00b) jest także stabilny. Ω (wersja 1.23) jest ciągle rozwijana; wersje wymienionych programów, znajdujące się na CD-ROM-ie, są najbardziej aktualne (stan z czerwca 2002).

Na płycie znalazła się ponadto eksperymentalna wersja NTS, reimplementacji T_EX-a zrealizowanej w Java Script. Do użycia NTS wymagane jest uprzednie zaistalowanie w systemie pakietu Java SDK, dostępnego dla wielu platform.

2. Struktura i zawartość CD-ROM-u

Poniżej wymienione są ważniejsze podkatalogi głównego katalogu CD-ROM.

bin programy rodziny T_EX-owej, rozdzielone w katalogach według platform systemowych;

Books pliki przykładów z różnych książek poświęconych systemowi T_EX.

FAQ najczęściej zadawane pytania, w wersji angielskiej, francuskiej i niemieckiej;

MacOSX programy pomocnicze dla użytkowników MacOSX;

info dokumentacja w formacie GNU „info” dla systemu T_EX;

man dokumentacja w formacie Unixowych stron „man” dla systemu T_EX;

setupw32 zawiera uaktualnienia bibliotek dla starszych wersji Windows 9x oraz pliki pomocnicze do instalacji (patrz część 4 na str. 15);

source źródła wszystkich programów, włącznie z głównymi dystrybucjami Web2c T_EX-a i META-FONT-a; są one umieszczone w skompresowanym programem bzip2 archiwum tar;

support różne programy związane z \TeX -em, które *nie* są domyślnie instalowane, takie jak \MusixTeX , programy pomocnicze, oraz kompletna dystrybucja Ghostscript – interpretera języka PostScript w wersji 7.05. Dla użytkowników systemów Windows dołączono ponadto szereg programów (edytory, „shelle \TeX -owe”), których zwykle brak w typowej instalacji Windows (programy te mogą być przydatne szczególnie dla początkujących). Program \TeX Setup dla Windows pozwala na ich instalację.

texmf główne drzewo z makrami, fontami i dokumentacją;

usergrps materiały dotyczące grup użytkowników systemu \TeX (\TeX User Groups).

Są także dwa skrypty instalacyjne dla systemów uniksowych, `install-cd.sh` i `install-pkg.sh`; będą one omówione w rozdziale 3 na str. 7.

2.1. Pakiety i ich zestawy

Zawartość drzewa `texmf` na **\TeX Live** do celów instalacji została zorganizowana w szereg „zestawów” (*collections*), z których każdy posiada zbiór „pakietów” (*packages*; jest ich na CD-ROM-ie ponad 800). Normalna instalacja pozwala użytkownikowi skopiować z CD-ROM-u na twardy dysk jeden lub więcej zestawów, ale jest też możliwe zainstalowanie tylko jednego pakietu.

Zestawy pozwalają lepiej dobrać instalowane składniki, a także określić, jakie języki będą obsługiwane po instalacji. Najważniejszy zestaw, wymagany dla większości zastosowań, to „*tex-basic*”. Zestawy „*tex-latex*” i „*tex-pdftex*” są rekomendowane dla większości użytkowników. Pozostałe zestawy są opcjonalne. Dostępne zestawy (zdefiniowane w plikach XML umieszczonych w katalogu `texmf/tpm/collections`) i krótki opis ich zawartości:

tex-basic podstawowe programy, pakiety makr i fontów systemu \TeX ; pliki konfiguracyjne dla podstawowych sterowników;

tex-bibtexextra dodatkowa, obszerna biblioteka stylów \BibTeX owych i bazy danych bibliograficznych (podstawowe style znajdują się w zestawie „*basic*”);

tex-chemistry pakiety makr do składu formuł chemicznych;

tex-context pakiet makr do „dialektu” \TeX -a, \ConTeXt , autorstwa Hansa Hagen;

tex-documentation dodatkowe, użyteczne dokumentacje;

tex-etex pomocnicze pliki dla programu \eTeX ;

tex-extrabin różne programy pomocnicze; zestaw zawiera m.in. programy i makra dla systemu `texinfo`, programy do manipulacji na plikach DVI itp.;

tex-fontbin programy konwersji plików fontowych i do testowania oraz instalacji fontów (zestaw do tworzenia fontów wirtualnych, manipulacji z plikami `.gf` i `.pk`, programy `mft`, `fontinst` itp.);

tex-fontsextra obszerna biblioteka różnych fontów w postaci źródłowej oraz pliki definicyjne i stylów \LaTeX -a dla rozmaitych fontów (także komercyjnych);

tex-formatsextra pliki pomocnicze do generowania dodatkowych „formatów” (tj. obszerne zestawy makr służące do wstępnego przetworzenia i utworzenia pliku `.fmt`);

tex-games pakiety do prezentacji zapisu różnych gier (szachy, brydż itp.);

tex-genericextra obszerna biblioteka makr, trudnych do sklasyfikowania, działających z różnymi formatami (Plain, LaTeX itp.);

tex-htmlxml pakiety konwersji \LaTeX do XML/HTML oraz do składu dokumentów XML/SGML;

tex-langaffrican wsparcie dla niektórych języków afrykańskich;

tex-langarmenian armeński;

tex-langcjk pakiety CJK (obsługa chińskiego, japońskiego i koreańskiego);

tex-langcroatian chorwacki;

tex-langcyrillic fonty i makra do składu cyrylicą;

tex-langczechslovak fonty i pakiety makr dla czeskiego i słowackiego;

tex-langdanish duński;

tex-langdutch holenderski;

tex-langfinnish fiński;

tex-langfrench francuski;

tex-langgerman niemiecki;

tex-langgreek grecki;

tex-langhungarian węgierski;

tex-langindic wsparcie dla języków indyjskich;

tex-langitalian włoski;

tex-langlatin łacina;

tex-langmanju język Manju;

tex-langmongolian mongolski;

tex-langnorwegian norweski;

tex-langother wzorce przenoszenia wyrazów dla innych języków;

tex-langpolish polskie pakiety: makra do tworzenia formatu MeX, pakiety platex, mwcls, fonty, makra dodatkowe, przydatne dla polskich użytkowników, dokumentacje w języku polskim;

tex-langportuguese portugalski;

tex-langspanish hiszpański;

tex-langswedish szwedzki;

tex-langtibetan fonty i wsparcie do składu tybetańskiego;

tex-langukenglish angielski brytyjski;

tex-langvietnamese wietnamski;

tex-latex podstawowe pakiety \LaTeX a oraz rekomendowany, często używany zestaw pakietów dodatkowych;

tex-latex3 pakiety-„zwiastuny” $\text{\LaTeX}3$, działające z $\text{\LaTeX}2_{\epsilon}$;

tex-latexextra bogaty wybór dodatkowych pakietów dla \LaTeX -a;

tex-mathextra dodatkowe pakiety do składu zaawansowanej matematyki;

tex-metapost MetaPost i pakiety do tworzenia rysunków z użyciem MetaFonta;

tex-music makra do składu nutowego;

tex-omega rozszerzenie \TeX a, Omega, działające ze znakami kodowanymi 16-bitowo, autorstwa Johna Plaice’a i Yannis Haralambousa;

tex-pdfTeX pomocnicze pliki dla pdf \TeX a, wersji programu \TeX tworzącej pliki w formacie PDF;

tex-pictures pakiety do tworzenia diagramów i innych obiektów graficznych;

tex-plainextra dodatkowe biblioteki makr dla Plain;

tex-psfonts dodatkowe fonty PostScriptowe;

tex-psutils narzędzia do manipulacji z plikami PostScriptowymi;

tex-publishers pakiety makr pochodzące od różnych wydawnictw;

tex-tlutils narzędzia do fontów Type1;

tex-textbooks przykłady z różnych książek poświęconych systemowi \TeX ;

tex-theses style do składu publikacji wydawanych przez różne uniwersytety;

tex-ttfutils narzędzia do konwersji fontów True Type;

win32-support dodatkowe programy wspomagające dla Windows (edytory, proste zintegrowane środowiska do pracy z \TeX em, programy graficzne itp.);

Katalog `texmf/tpm/packages` (wykorzystywany podczas instalacji) zawiera spis wszystkich plików w każdym pakiecie.

Uwaga: CD-ROM jest w formacie ISO 9660 (High Sierra), z rozszerzeniami Rock Ridge i Joliet. Aby skorzystać z wszystkich zalet CD-ROM-u pod Unixem, twój system musi umieć używać rozszerzeń Rock Ridge. Proszę zapoznać się z dokumentacją komendy mount w twoim systemie aby sprawdzić, czy to jest możliwe. Jeżeli używasz wielu różnych maszyn w sieci lokalnej zobacz, czy możesz zamontować CD-ROM na tej, która pozwala na użycie Rock Ridge.

Systemy Linux, FreeBSD, Sun, SGI oraz DEC Alpha powinny umieć bez problemów korzystać z tego CD-ROM-u. Będziemy wdzięczni za wszelkie wskazówki od użytkowników innych systemów, którym udało się skorzystać z niniejszego CD-ROM-u, z których to wskazówek moglibyśmy skorzystać w przyszłych wersjach tej dokumentacji.

Dalsze kroki opisane są przy założeniu, że udało się zamontować CD-ROM w pełnej zgodności z Rock Ridge.

3. Instalacja i użytkowanie pod Unixem

CD-ROM **T_EX Live** można używać na trzy sposoby:

1. zamontować (*mount*) CD-ROM w systemie plików, uruchomić skrypt `install-cd.sh` i wybrać opcję <R> (*do not install files, set up to run off CD-ROM* – bez instalacji plików, dostosuj do uruchamiania z CD-ROM-u). Pozwala to na uruchamianie wszystkich programów bezpośrednio z płytki; na dysku twardym znajdą się jedynie niezbędne pliki konfiguracyjne. Chociaż wydajność nie będzie optymalna, jest to w pełni akceptowalne np. na komputerach PC z Linuxem;
2. zainstalować część lub cały system na lokalnym twardym dysku; jest to najlepsza metoda dla wielu użytkowników, jeżeli tylko mają wystarczającą ilość miejsca na dysku (co najmniej 40 megabajtów, albo około 130 megabajtów w przypadku instalacji zalecanej);
3. doinstalować wybrane pakiety do istniejącego systemu **T_EX** lub systemu **T_EX Live** instalowanego wcześniej.

Każda z tych metod jest dokładnie opisana w dalszych rozdziałach.

3.1. Wstępne kroki przed instalacją dla systemu MacOSX

Jeżeli nie mamy systemu MacOSX, należy pominąć ten podrozdział.

Skrypt `install-cd.sh` jest skryptem powłoki `sh` (rozpoczyna się instrukcją `#!/bin/sh`), ale powłoka `tcsh` systemu MacOSX nie potrafi go wykonać. Najlepiej do tego wykorzystać powłokę `bash`, która w systemie MacOSX nie jest instalowana domyślnie¹.

1. (opcjonalnie) należy sprawdzić, czy powłoka `bash` jest już zainstalowana. W tym celu trzeba uruchomić Terminal (`/Applications/utilities/Terminal`) i w oknie terminala wpisać:

```
>> rehash; which bash
```

¹Prawdopodobnie ulegnie to zmianie, a nawet powłoka `bash` będzie wykorzystywana do emulowania `sh`. Jeśli tak się stanie, to w przyszłych wersjach MacOSX skrypt będzie działał.

odpowiedź może być następująca:

- lokalizacja powłoki bash (na przykład `/bin/bash` albo `usr/local/bin/bash`), jeśli jest ona zainstalowana;
- `bash: command not found`, jeżeli nie została jeszcze zainstalowana.

Jeżeli powłoka bash jest już zainstalowana, można przejść do punktu 4.

2. Instalacja powłoki bash:

Przyjazna procedura macintoshowa Na CD-ROM-ie w katalogu MacOSX należy odszukać obraz dysku o nazwie `bash.dmg`. Po dwukrotnym kliknięciu na nazwie tego pliku obraz dysku (wolumen) zostanie zamontowany. W tym wolumenie uruchamiamy aplikację `i-Installer`. Użytkownik zostanie poproszony o przedstawienie się. Jeśli pierwszy raz widzimy taki komunikat, może się okazać, że nie mamy wystarczających uprawnień do przeprowadzenia instalacji. Należy teraz wpisać nazwę użytkownika i hasło, po czym kliknąć przycisk `Install`. Powłoka bash zostanie zainstalowana w systemie.

Procedura z wykorzystaniem linii poleceń (a) Zalogować się jako użytkownik-administrator, a przynajmniej jako użytkownik o uprawnieniach administratora, użytkownik `sudo` albo Administrator Systemu.

(b) Przejść do katalogu MacOSX znajdującego się na CD-ROM-ie i skopiować plik `bash.tar.gz` do katalogu domowego (`~/`).

(c) Uruchomić Terminal, a następnie wpisać w oknie terminala następujące polecenie:

```
>> (cd /usr/local/; sudo tar xvfz ~/bash.tar.gz)
```

Nacisnąć klawisz [Enter]. Użytkownik zostanie poproszony o hasło, a następnie bash zostanie zainstalowany.

(d) Zamknąć Terminal.

3. Po przeprowadzeniu którejkolwiek z powyższych procedur instalacji należy przejść do kroku 1, aby uzyskać odpowiedź `/usr/local/bin/bash...` (jeśli się nie uda, należy się wylogować, i ponownie zalogować).

4. Instalacja w systemie MacOSX przebiega tak samo jak na innych platformach uniksowych (nie ma w tym nic niezwykłego, ponieważ MacOSX jest systemem Unix). Warto zapoznać się jednak z następującymi wskazówkami:

- zauważmy, że we wszystkich poleceniach z poniższych podrozdziałów `sudo bash` powinno zastąpić `sh`. Na przykład polecenie

```
>> sh install-cd.sh
```

powinno mieć postać

```
>> sudo bash install-cd.sh
```

- w systemie MacOSX CD-ROM-y są montowane automatycznie, więc nie trzeba stosować polecenia `mount`. CD-ROM zostanie zamontowany w katalogu `/Volumes/`. Jeśli chcemy, aby stał się on katalogiem bieżącym, w oknie terminala należy wpisać:

```
>> cd /Volumes/TeXLive-7...
```


w miejsce kropek wstawiamy rzeczywistą nazwę CD-ROM-u, wykorzystując tu funkcję automatycznego uzupełniania (klawisz [Tab]).

3.1.1. Strona kodowa Macintosh CE

Piszący po polsku użytkownicy MacOSX mają do wyboru stosowanie dwóch 8-bitowych stron kodowych:

iso8859-2 – co się wiąże z wykorzystywaniem standardowego oprogramowania uniksowego (Cocoa używa wyłącznie unikodu), jak na przykład Emacs.

Mac CE – co pozwala na stosowanie tradycyjnego (Carbon) oprogramowania jak np.: BBEdit. Weźmy pod uwagę ten przypadek.

Na **T_EX Live 7** po raz pierwszy znalazł się plik `macce.def`, który pozwala L^AT_EX-owi wczytywać polski tekst z wykorzystaniem pakietu `inputenc`:

```
\usepackage[macce]{inputenc}
```

(w najbliższym czasie spodziewane jest pojawienia się tego pliku w standardowej dystrybucji L^AT_EX-a).

Również po raz pierwszy znalazły się na **T_EX Live 7** tablice przekodowań dla Mac CE, niewymagające stosowania pakietu `\inputenc`. Zamiast niego wystarczy wpisać deklarację w pierwszym wierszu pliku:

```
%& --translate-file=macce-pl
```

3.2. Uruchamianie T_EX Live z CD-ROM-u

Struktura oprogramowania Web2c oznacza, że można uruchamiać programy przez zwykłe dodanie do zmiennej `PATH` odpowiedniego podkatalogu katalogu `bin` na CD-ROM-ie. Potrzebne pliki będą znajdowane bez dalszych problemów. Poniżej zamieszczony jest spis dostępnych systemów i odpowiednich nazw podkatalogów. **Podstawowy CD-ROM zawiera jedynie binaria dla Windows, x86 Linux i Mac OSX. Dla innych systemów Unix należy zastosować drugą płytkę.** Jej skompresowany „obraz” udostępniono w sieci.

Compaq Alpha Linux	<code>alpha-linux</code>	CD2
Compaq Alphaev5 OSF 4.0d	<code>alphaev5-osf4.0d</code>	CD2
HP9000 HPUNIX 10.20	<code>hppa2.0-hpux10.20</code>	CD2
IBM RS 6000 AIX 4.2.*	<code>rs6000-aix4.2.1.0</code>	CD2
Intel x86 Solaris 2.8	<code>i386-solaris2.8</code>	CD2
Intel x86 with GNU/Linux	<code>i386-linux</code>	CD1
Mac OSX	<code>powerpc-darwin5.3</code>	CD1
Sun Sparc Solaris 2.7	<code>sparc-solaris2.7</code>	CD2
Windows 9X/ME/NT/2K/XP	<code>win32</code>	CD1

Wydawałoby się, że potrzeba wygenerowania fontów czy zmiana konfiguracji może spowodować problemy, ponieważ nie można zmieniać plików na CD-ROM-ie. Można jednak posiadać na dysku drugie, równoległe drzewo T_EX-owe, w którym można dokonywać zmian. Drzewo takie jest przeszukiwane przed głównym drzewem CD-ROM-u. Domyślną lokalizacją jest `texmf-var` na CD-ROM-ie (katalog,

który faktycznie nie istnieje!), wobec tego *należy* ją zmienić poprzez deklarację zmiennej środowiska VARTEXMF.

Przykładowo, użytkownicy *sh* lub *bash* na Intel PC z systemem Linux mogą zamontować CD-ROM **T_EX Live** w katalogu `/mnt/cdrom` za pomocą polecenia:

```
>> mount -t iso9660 /dev/cdrom /mnt/cdrom
```

Po zmianie katalogu bieżącego na główny katalog CD-ROM-u należy uruchomić skrypt:

```
>> sh install-cd.sh
```

Odpowiednie pliki pomocnicze zostaną zainstalowane na twardym dysku po wybraniu opcji <R> (*do not install files, set up to run off CD-ROM* – bez instalacji pików, dostosuj do uruchamiania z CD-ROM-u). Następnie należy uaktualnić zmienną PATH, dołączając do ścieżki przeszukiwania katalog zawierający binaria dla danej architektury.

```
PATH=/mnt/cdrom/bin/i386-linux:$PATH
export PATH
VARTEXMF=/usr/TeX/texmf-var
export VARTEXMF
```

W systemie MacOSX domyślną powłoką jest *tcsh*:

```
setenv PATH /Volumes/<cd-name>/bin/powerpc-darwin5.3:${PATH}
setenv VARTEXMF /usr/TeX/texmf-var
```

Dla wygody, powyższe wiersze można dodać do skryptu `.profile`. (dla *tcsh* w MacOSX, `~/Library/init/tcsh/rc.mine`).

Po ponownym uruchomieniu sesji polski użytkownik może użyć bezpośrednio m.in. polecenia *mex* (T_EX z „polskim” formatem Plain), *platex* (L^AT_EX z polskimi i anglo-amerykańskimi wzorcami przenoszenia wyrazów) oraz ich wersje dla programu pdfT_EX (tworzącego pliki PDF), odpowiednio: *pdfmex* i *pdfplatex*. Potrzebne pliki formatów zostaną wygenerowane automatycznie podczas pierwszego uruchomienia danego polecenia i zapisane w lokalnym katalogu. Katalog ten, `/usr/TeX/texmf-var/`, zawiera pliki konfiguracyjne, które mogą być modyfikowane przez użytkownika.

3.3. Instalacja T_EX Live na twardym dysku

Instalacja części lub całości dystrybucji na twardym dysku wymaga zamontowania CD-ROM-u w systemie, przejścia do jego katalogu głównego i uruchomienia:

```
>> sh install-cd.sh
```

(W niektórych systemach Unix może zająć potrzeba użycia *sh5* lub *bash*.) Skrypt ten przeszukuje listy zestawów i pakietów z CD-ROM-u, oraz próbuje odgadnąć, z jakim systemem komputerowym ma do czynienia. Skrypt powinien zacząć pracę od wyświetlenia:

```
Initializing collections... Done initializing.
Counting selected collections... Done counting.
Calculating disk space requirements for collections...Done calculating that.
Initializing system packages... Done initializing system.
```

```

====> Note: Letters/digits in <angle brackets> indicate menu items <====
====>         for commands or configurable options         <====

Proposed platform: Intel x86 with GNU/Linux
<P> over-ride system detection and choose platform
<B> binary systems:          1 out of  9
<S> Installation scheme (texlive_recommended)
[customizing installation scheme:
  <C> standard collections  <L> language collections]
1 out of 57, disk space required: 12960 kB
<D> directories:
  TEXDIR      (The main TeX directory)      : /usr/TeX
  TEXMFLOCAL  (Directory for local styles etc): /usr/TeX/texmf-local
  VARTEXMF    (Directory for local config)   : /usr/TeX/texmf-var
<O> options:
  [ ] alternate directory for generated fonts ( )
  [ ] create symlinks in standard directories
  [ ] do not install macro/font doc tree
  [ ] do not install macro/font source tree
<R> do not install files, set up to run off CD-ROM
<I> start installation, <H> help, <Q> quit

Enter command:

```

Rysunek 1: Główny ekran skryptu instalacyjnego

Następnie pokaże główną stronę konfiguracyjną (Rys. 1), umożliwiającą wybór:

1. <P> systemu operacyjnego, dla którego chcemy dokonać instalacji;
2. <S> *installation scheme* (predefiniowanego zestawu instalacyjnego), np. *full* – pełnego, *recommended* – zalecanego, *basic* – podstawowego; mamy tu do wyboru zestaw proponowany dla polskich użytkowników: *GUST scheme*;
3. zestawów, które mogą zmodyfikować wybrany schemat instalacji (są one zgrupowane w dwóch kategoriach: <C> *standard collections* – zestawy standardowe oraz <L> *language collections* – zestawy dla obsługiwanych języków);
4. <D> katalogu bazowego na dysku, przeznaczzonego na instalację;
5. <O> ustawienia trybu pracy niektórych programów.

Opcje wybiera się przez wpisanie litery lub liczby i potwierdzenie jej klawiszem Enter. Przykładowo: wykryty został system Linux ELF, wybrane zostały predefiniowane zestawy, domyślny katalog instalacji to `/usr/TeX`. Zauważ, że wyświetlana jest także ilość miejsca na dysku, potrzebna do zainstalowania pakietów. Jeśli wykonuje się sugerowaną instalację, potrzebne jest około 100 megabajtów wolnego miejsca na dysku. Podstawowa instalacja zajmie jedynie 50 megabajtów; można ją poszerzyć o wybrane pakiety, kiedy zajdzie taka potrzeba.

Użytkownicy MacOSX – dostępne programy instalacyjne (*TeXShop*, *ITeXMac*...) stosują domyślną lokalizację `teTeX`, czyli `/usr/local/teTeX`. Wobec czego użytkownicy Mac dla instalacji **TeX Live** powinni rozważyć wybór katalogu `/usr/local/teTeX` zamiast `/usr/TeX`.

W katalogu wybranym do instalacji, w podkatalogu `bin`, skrypt instalacyjny umieści binaria. Drzewo z plikami `TeX`-owymi zostanie umieszczone w podkatalogu `texmf`. Dodatkowy podkatalog

```

a [X] Essential programs and files  p [ ] LaTeX supplementary packages
b [ ] Extra BibTeX styles          s [ ] Advanced math typesetting
c [ ] Chemical typesetting         t [ ] Music typesetting
d [ ] Context macro package        u [ ] Omega
e [X] Extra documentation          v [X] pdfTeX
f [ ] eTeX                        w [ ] Drawing and graphing packages
g [ ] TeX auxiliary programs       x [ ] Plain TeX extra macros
h [ ] TeX font-related programs    y [ ] Extra PostScript fonts
i [ ] Extra fonts                  z [ ] PostScript utilities
j [ ] Extra formats                A [ ] Support for publishers
k [ ] Games typesetting (chess, etc B [ ] Type1 font manipulation
l [ ] Miscellaneous macros         C [ ] Examples from TeX books
m [ ] HTML/SGML/XML support        D [ ] Styles for University theses
n [X] Basic LaTeX packages         E [ ] TrueType font manipulation
o [ ] Support for latex3           F [ ] Various support tools for win

<-> deselect all <+> select all <R> return to platform menu <Q> quit

Press key to toggle status of collection:

```

Rysunek 2: Wybór zestawów standardowych

```

a [ ] Support for some African scri o [X] Support for Italian
b [ ] Support for Armenian          p [ ] Support for Latin
c [ ] Chinese, Japanese, Korean sup s [ ] Support for Manju
d [ ] Support for Croatian          t [ ] Support for Mongolian
e [ ] Support for Cyrillic          u [ ] Support for Norwegian
f [X] Support for Czech/Slovak      v [ ] Other hyphenation files
g [ ] Support for Danish            w [X] Support for Polish
h [X] Support for Dutch             x [X] Support for Portuguese
i [ ] Support for Finnish           y [X] Support for Spanish
j [X] Support for French            z [ ] Support for Swedish
k [X] Support for German            A [ ] Support for Tibetan
l [ ] Support for Greek             B [X] Support for UK English
m [ ] Support for Hungarian         C [ ] Support for Vietnamese
n [ ] Support for Indic

<-> deselect all <+> select all <R> return to platform menu <Q> quit

Press key to toggle status of collection:

```

Rysunek 3: Wybór zestawu dla wymaganego języka

texmf-var będzie zawierał kopie plików konfiguracyjnych (prócz głównego texmf.cnf), które będą modyfikowane podczas uruchamiania programu texsetup; znajdują się w tym katalogu także generowane pliki formatów.

Po wybraniu opcji <C> – *standard collections*, wyświetlone zostaną dostępne zestawy (rys. 2). Każdy zestaw zawiera szereg pakietów: makr, fontów itp. Zaznaczenie zestawu do zainstalowania polega na wciśnięciu odpowiedniego klawisza; ponowne naciśnięcie tego samego klawisza wyłącza zaznaczenie (Uwaga: duże i małe litery są rozróżniane!).

Po wybraniu opcji <L> – *language collections*, wyświetlone zostaną dostępne zestawy dla różnych języków (rys. 3). Zestawy zawierają najczęściej wzorce przenoszenia wyrazów, czasami dodatkowe pakiety makr i fontów. Aby zainstalować polskie pakiety (fonty, makra, dokumentacje), należy w tym menu zaznaczyć klawiszem <w> Support for Polish.

Po powrocie do głównego menu można dodatkowo wybrać inne opcje. I tak: <0> options pozwala zdecydować, czy nowo tworzone fonty powinny znaleźć się w innym miejscu (jeżeli chcemy, dla większości użytkowników, zainstalować główny pakiet w trybie „tylko do czytania” (*read-only*)), oraz czy utworzyć dowiązania symboliczne (*links*) dla stron man i GNU info w standardowych katalogach systemu. Wymagane do tego są, oczywiście, uprawnienia administratora.

Po zakończeniu ustawień, po powrocie go głównego menu, można rozpocząć instalację (klawisz <I>). Program instalacyjny, uwzględniając wybrane zestawy i systemy, sprawdzi spisy plików na CD-ROM-ie, stworzy listę plików do skopiowania, po czym pliki te zostaną skopiowane na twardy dysk. Następnie uruchamiana jest sekwencja inicjująca (tworzenie plików formatów itd.). Po instalacji wystarczy dodać do zmiennej PATH właściwy podkatalog bin instalacji T_EX-a i można zacząć pracę. Jeśli zajdzie potrzeba, można przenieść binaria wyżej o jeden poziom, np. z /usr/local/bin/alpha-osf3.2 do /usr/local/bin; jednakże po tym należy wyedytować plik texmf/web2c/texmf.cnf (patrz Dodatek 11) i zmienić linię znajdującą się blisko początku pliku:

```
TEXMFMAIN = $SELFAUTOPARENT
```

na

```
TEXMFMAIN = $SELFAUTODIR
```

Jeśli przeniesiemy całość instalacji do innego katalogu, należy dokładnie to określić, modyfikując zmienną TEXMFMAIN w pliku texmf.cnf, a także ustawiając TEXMFCNF na \$TEXMFMAIN/texmf/web2c.

3.4. Instalacja pojedynczych pakietów z T_EX Live na twardym dysku

CD-ROM T_EX Live można używać zarówno do aktualizacji istniejącej instalacji, jak i do dodawania składników do wcześniejszej instalacji z CD-ROM-u. Główny program instalacyjny przeznaczony jest do użycia tylko za pierwszym razem, natomiast później należy używać znajdującego się na CD-ROM-ie skryptu install-pkg.sh. Uruchamia się go przez zamontowanie CD-ROM-u, przejście do katalogu odpowiadającego temu CD-ROM-owi i napisanie

```
>> sh install-pkg.sh opcje
```

Skrypt posiada dziewięć parametrów; pierwsze cztery pozwalają wybrać: pojedynczy pakiet do instalacji, cały zestaw (np. tex-mathextra), nazwę katalogu zamontowanego CD-ROM-u, oraz nazwę katalogu zawierającego pliki ze spisami zawartości pakietów (zwykle te dwa ostatnie ustawiane są automatycznie):

```
--package=nazwa  
--collection=nazwa  
--cddir=nazwa  
--listdir=nazwa
```

Proces instalacji jest określony dokładniej przez cztery kolejne parametry; pierwsze dwa pozwalają wyłączyć z instalacji dokumentację lub pliki źródłowe, trzeci wyłącza domyślne uruchamianie na koniec skryptu mktexlsr, który przebudowuje bazę danych o plikach, a czwarty powoduje jedynie wypisanie plików, które miałyby być zainstalowane:

```
--nodoc
--nosrc
--nohash
--listonly
```

Wreszcie można określić, czy zamiast instalować pliki, skrypt powinien utworzyć archiwum tar w wybranej lokalizacji:

```
--archive=nazwa
```

Tak więc, jeżeli chcielibyśmy tylko zobaczyć przed zainstalowaniem, jakie pliki tworzą pakiet fancyhdr, nasze polecenie wyglądałoby następująco:

```
>> sh install-pkg.sh --package=fancyhdr --listonly
```

```
texmf/doc/latex/fancyhdr/fancyhdr.dvi
texmf/doc/latex/fancyhdr/fancyhdr.tex
texmf/lists/fancyhdr
texmf/source/latex/fancyhdr/README
texmf/source/latex/fancyhdr/fancyheadings.new
texmf/tex/latex/fancyhdr/extramarks.sty
texmf/tex/latex/fancyhdr/fancyhdr.sty
texmf/tex/latex/fancyhdr/fixmarks.sty
```

Oto inne przykłady użycia:

- zainstaluj L^AT_EX-owy pakiet natbib:

```
>> sh install-pkg.sh --package=natbib
```

- zainstaluj L^AT_EX-owy pakiet alg bez plików źródłowych i bez dokumentacji:

```
>> sh install-pkg.sh --package=alg --nosrc --nodoc
```

- zainstaluj wszystkie pakiety dostępne w zestawie dodatkowych makr dla Plain T_EX:

```
>> sh install-pkg.sh --collection=tex-plainextra
```

- spakuj wszystkie pliki wymagane dla PStricks w pliku tar, w katalogu /tmp:

```
>> sh install-pkg.sh --package=pstricks --archive=/tmp/pstricks.tar
```

3.5. Program texconfig

Gdy tylko program instalacyjny skopiuje wszystkie pliki w miejsce, gdzie mają być zainstalowane, można użyć programu texconfig, pozwalającego konfigurować system według własnych potrzeb. Program ten może być w każdej chwili uruchomiony do zmiany instalacji, z interfejsem pełnoekranowym (wymaga on programu dialog, zawartego w pakietach z binariami) lub z linii poleceń. Program texconfig powinien być używany do wszelkich zmian, takich jak ustawianie drukarek czy przebudowywanie bazy danych o plikach. Obydwa tryby pracy mają tekst pomocy opisujący możliwości.



Rysunek 4: Okno programu TeXLive

4. Instalacja i użytkowanie pod Windows

Rozdział ten odnosi się tylko do systemów Windows 9x, ME, 2000, NT lub XP.

Przed instalacją należy sprawdzić, czy Windows potrafi, przy czytaniu CD-ROM-ów, korzystać z rozszerzeń Microsoft Joliet. Uruchamiając Eksplorator wystarczy sprawdzić, czy wyświetlana lista plików CD-ROM-u posiada „długie” nazwy, pisane małymi i wielkimi literami. Jeżeli tak nie jest, nie można używać systemu uruchamialnego bezpośrednio z CD-ROM-u.

Zawarty tu system Win32 T_EX zawiera przeglądarkę plików dvi, Windvi, która w użyciu podobna jest do Unixowego xdvi. Dokumentacja znajduje się w texmf/doc/html/windvi/windvi.html.

4.1. Program TeXLive.exe

Jeśli system Windows jest skonfigurowany tak, aby umożliwiał automatyczny start programów po włożeniu CD-ROM-u do napędu, program TeXLive pokaże okno dialogowe, w którym mamy do wyboru szereg opcji:

- Install->TeXLive – instalacja na dysku twardym;

- Install->Editors – instalacja na dysku edytorów przygotowanych do współpracy z systemem \TeX ;
- Install->Support – dodatkowe pakiety przydatne do pracy z systemem \TeX (Ghostscript, NetPBM itp.);
- Install->TeXLive uninstall – usunięcie systemu \TeX ;
- Explore CD-ROM->Browse CD-ROM – przeglądanie zawartości płytki;
- Explore CD-ROM->Select a text editor – wybór uprzednio zainstalowanego edytora do uruchamiania programów bezpośrednio z CD-ROM;
- Explore CD-ROM->Run TeX off CD-Rom – uruchamianie systemu \TeX bezpośrednio z płytki (dostępne po wyborze edytora);
- Explore CD-ROM->Cleanup CD-Rom Setup – usunięcie tymczasowych plików tworzonych na dysku twardym podczas uruchamiania bezpośrednio z płytki;
- Maintenance – wykonanie niektórych czynności konfiguracyjnych;
- System DLLs update – aktualizacja systemowych bibliotek DLL (jest to wymagane dla starszych wersji win95);
- Documentation – przeglądanie wybranej dokumentacji: **TeX Live** (także w języku polskim), strony TUG, strony dystrybucji \fpTeX , a także uruchamianie TeXdocTK – programu do przeglądania dokumentacji dotyczących systemu \TeX (program ten jest w fazie eksperymentalnej).

Gdy włożenie CD-ROM-u do napędu nie uruchamia automatycznie programu, należy w oknie Eksploratora Windows wskazać bin/win32/TeXLive.exe i uruchomić program dwukrotnym naciśnięciem klawisza myszy.

4.2. Uruchamianie z CD-ROM-u

Wszystkie programy \TeX -owe można uruchamiać bezpośrednio z CD-ROM-u, mając natychmiastowy dostęp do wszystkich makr i fontów. Ceną jest tu wolniejsza praca niż w wypadku instalacji na twardym dysku. Do efektywnej pracy wymagana jest modyfikacja zmiennych środowiska systemu i utworzenie niewielkich, pomocniczych katalogów na dysku twardym. Katalogi te będą zawierały niezbędne pliki konfiguracyjne, pozwalające użytkownikowi na modyfikację ustawień programów i samodzielne generowanie potrzebnych plików formatów. Ponadto, w pomocniczym katalogu będą zapisywane generowane automatycznie fonty.

Wymienione kroki przygotowawcze wykonuje program TeXlive.exe:

1. Po uruchomieniu programu należy wybrać opcję Explore CD-Rom, następnie Select a text editor. Wyświetlone zostanie okno dialogowe, pozwalające wybrać *uprzednio zainstalowany* edytor tekstowy. Edytor ten powinien umożliwiać współpracę ze środowiskiem \TeX : uruchamiać programy do kompilacji, przeglądania itp. Jeśli tego typu edytor nie jest zainstalowany, należy go zainstalować z menu Install, szczegóły w części 4.3.

W systemie Windows nie ma możliwości odgadnięcia, czy wybrany program jest rzeczywiście edytorem tekstowym, wobec tego użytkownik powinien sam to starannie określić. Najczęściej używane edytory to:

GNU Emacs	c:\Program Files\NTEmacs\bin\runemacs.exe
XEmacs	c:\Program Files\XEmacs\XEmacs-21.2\i586-pc-win32\xemacs.exe
WinShell	c:\Program Files\WinShell\WinShell.exe
WinEdt	c:\Program Files\WinEdt Team\WinEdt\WinEdt.exe
TeXnicCenter	c:\Program Files\TeXnicCenter\TEXCNTR.exe

Wybrany program zostanie zapamiętany dla późniejszego uruchamiania. Użytkownikom (szczególnie początkującym) można polecić dostępny na płycie WinShell, łatwo konfigurowalne, posiadające spolonizowane menu i komunikaty środowisko do pracy z systemem \TeX . Bardziej zaawansowani użytkownicy wybiorą zapewne NTEmacs.

2. Z menu Explore CD-ROM należy wybrać Run \TeX off CD-ROM. Do tymczasowego katalogu skopiowane zostaną pliki konfiguracyjne oraz wygenerowane będą pliki najczęściej używanych formatów: tex (Plain) i latex. Po utworzeniu lokalnej bazy danych o instalacji (plik ls-R) ustawione zostaną tymczasowe zmienne środowiska: PATH oraz TEXMFCNF, po czym zostanie uruchomiony wybrany edytor, umożliwiający edycję plików, ich przetwarzanie, podgląd wyniku składu itp.

Jeśli w systemie nie zostanie wykryty program Ghostscript (interpreter PostScriptu), ukaże się ostrzeżenie, iż wyświetlenie wyniku składu może być wadliwe. Ghostscript można zainstalować z menu Install->Support (patrz 4.3).

3. Wybór innego edytora jest w każdej chwili możliwy.

Podczas pierwszego uruchomienia WinShell należy wybrać język menu i komunikatów (np. polski), a następnie kodowanie użytego w edytorze fontu (menu Opcje -> Font -> Zmień -> Skrypt – należy tu wybrać: „Europa Srodkowa”).

4. Menu Cleanup CD-Rom setup pozwala na usunięcie wszystkich plików wymaganych do uruchamiania programów \TeX -owych. Dodatkowo zainstalowane (z dysku bądź z sieci) programy nie będą usunięte, np. jeśli zainstalowano WinShell lub NTEmacs, nie będą one usunięte.

Edytor jest uruchamiany w ramach zmienionego środowiska. W katalogu przeznaczonym na pliki tymczasowe tworzone jest drzewo katalogów texmf, zgodne z TDS. Jest to potrzebne do automatycznego generowania plików fontowych, a także przechowywania plików konfiguracyjnych (można je modyfikować w razie potrzeby). Z uruchomionego edytora² mamy dostęp do pełnego środowiska programów **\TeX Live**, znajdujących na płycie.

Bardziej zaawansowani użytkownicy (szczególnie ci, którzy chcą uruchamiać programy z wiersza poleceń i samodzielnie modyfikować ustawienia programów) mogą przygotować system, wykorzystując plik wsadowy mkloctex.bat (katalog setupw32). Po skopiowaniu pliku na dysk twardy należy go uruchomić z wiersza poleceń (okno DOS/CMD), podając dwa parametry: literę CD-ROM-u i literę dysku twardego, na którym chcemy utworzyć pomocnicze drzewko \TeX -owe. Po zapoznaniu się z końcowymi komunikatami wyświetlanymi przez program zalecane jest wykonanie sugerowanych tam czynności.

²Obecnie możliwe jest także uruchomienie zamiast edytora np. procesora poleceń (okno DOS lub CMD). Takie środowisko otrzymuje prawidłowe ustawienia zmiennych do uruchamiania programów bezpośrednio z CD-ROM-u z wiersza poleceń. W oknie możliwe jest zatem uruchamianie dowolnego edytora tekstowego.

4.3. Instalacja edytorów i innych programów pomocniczych

Program TeXLive.exe (jak i dalej opisany TeXSetup.exe) może być wykorzystany do zainstalowania pojedynczego programu, niekoniecznie związanego z systemem T_EX. Może to być edytor, jak WinShell lub NTEmacs, a także inny program, np. NetPBM (zestaw programów do konwersji plików graficznych) bądź Ghostscript (interpreter PostScript).

Niektóre programy posiadają ograniczenia licencyjne, wobec czego nie znalazły się na płycie, ale dostępne są z sieci. Aby je zainstalować, należy uaktywnić menu Install->Enable Internet access, które wykryje aktywne połączenie z Internetem bądź uruchomi procedurę połączenia modemem. Uwaga: *jeśli komputer nie ma w ogóle możliwości połączenia z Internetem, czas zwłoki może być dość długi (30 sek. lub więcej)*. Zatem zaleca się uaktywnianie menu jedynie wtedy, jeśli użytkownik jest pewien, że posiada możliwość połączenia.

Tylko nieliczne z przydatnych programów znalazły się na płycie. Najważniejsze to wymienione już: edytory NTEmacs i WinShell, Ghostscript i NetPBM (jest on wymagany np. do uruchamiania programu T_EX4ht).

Programy do zainstalowania z sieci są nieraz spore, np. Perl 10MB, XEmacs 50MB, a więc pobranie ich i instalacja może trwać długo, zależnie od możliwości połączenia. Program TeXSetup.exe nie jest w stanie tego oszacować. Podczas instalacji niektórych programów TeXSetup.exe pracuje w trybie nie wymagającym interwencji użytkownika. Inne programy posiadają własne procedury instalacyjne, wymagają zatem reakcji typu wybór katalogu itp.

Najbardziej zalecanym do instalacji z sieci programem jest niewątpliwie GSView, interfejs graficzny dla GhostScript, pozwalający na przeglądanie na ekranie plików postscriptowych (PS, EPS), również plików PDF, a także ich drukowanie. Na szczęście jest on rozsądnych rozmiarów (ok. 1,5MB).

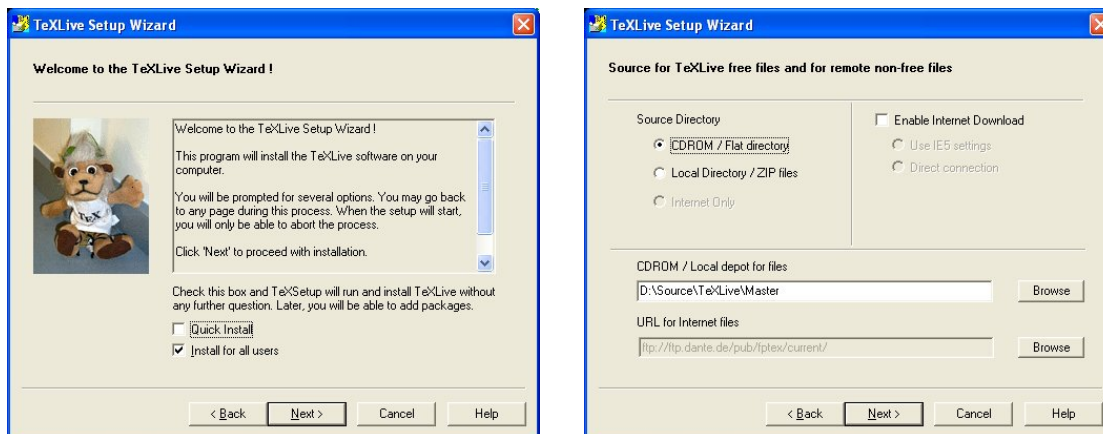
Pakiety bez specyficznego programu instalacyjnego są zazwyczaj rozpakowywane i konfigurowane na dysku wybranym do instalacji pozostałych programów. Przykładowo, instalowane NTEmacs lub NetPBM zawarte są w skompresowanych plikach zawierających strukturę katalogów (NTEmacs\ i NetPBM\). Zostaną one zainstalowane w ramach podanego do instalacji dysku i katalogu (c:\Local lub c:\Program Files).

4.4. Instalacja na twardym dysku

Instalację rozpoczyna się od uruchomienia (automatycznie – po włożeniu płytki do napędu, lub bezpośrednio – z katalogu bin\win32) programu TeXlive.exe i wybrania opcji Install -> TeXLive, która uruchamia główny program instalacyjny TeXSetup.exe (można go też uruchomić bezpośrednio z katalogu bin\win32). Program posiada szereg ekranów dialogowych:

Ekran ‘Welcome...’ (rys. 5, po lewej); można tu wybrać Quick Install – szybką instalację, bez interwencji użytkownika, z domyślnym wyborem zestawów (tex-basic, tex-latex i tex-pdftex). Jeśli jednak chcemy zainstalować dodatkowe programy pomocnicze (edytor, Ghostscript), nie należy używać tej opcji. W wersjach Windows gdzie ma to zastosowanie można uaktywnić opcję Install for all users – udostępnienie instalacji dla wszystkich użytkowników (wymagane są do tego prawa administratora!);

Ekran ‘Source...’ zawiera nieco więcej elementów; pozwala wybrać źródło instalacji (dla CD T_EX Live ustawiany jest domyślnie napęd CD-ROM, może to też być lokalny katalog na dysku twardym), ponadto pozwala zainstalować dodatkowe programy bezpośrednio z Internetu (rys. 5, po prawej).



Rysunek 5: Program TeXSetup wizar

Do czego potrzebne mogą być różne źródła instalacji? Właściwe pliki systemu **T_EX Live** są zawarte na płycie, ale brak niektórych pakietów, przydatnych dla użytkowników Windows, czy to z powodu braku miejsca, czy też z powodu ograniczeń licencyjnych. Aby zainstalować dodatkowe programy z Sieci należy włączyć opcję **Enable Internet Download**.

Nie należy traktować powyższego jako ograniczenie: CD-ROM pozwala zainstalować w pełni funkcjonalny system. Programy dodatkowe (np. WinEdt) można doinstalować później, w dogodnej chwili.

Pliki mogą więc być zainstalowane:

- z CD-ROM-u lub podobnego drzewa katalogów, dostępnego w którymś z katalogów (oznacza to, że CD-ROM może być zamontowany na zdalnym komputerze i udostępniony w sieci);
- ze skompresowanych plików .zip, pobranych z Sieci i dostępnych na dysku (jest to przypadek dystrybucji fpT_EX);
- z Internetu, w tym przypadku program pobierze z sieci pliki .zip, rozpakuje je i zainstaluje.

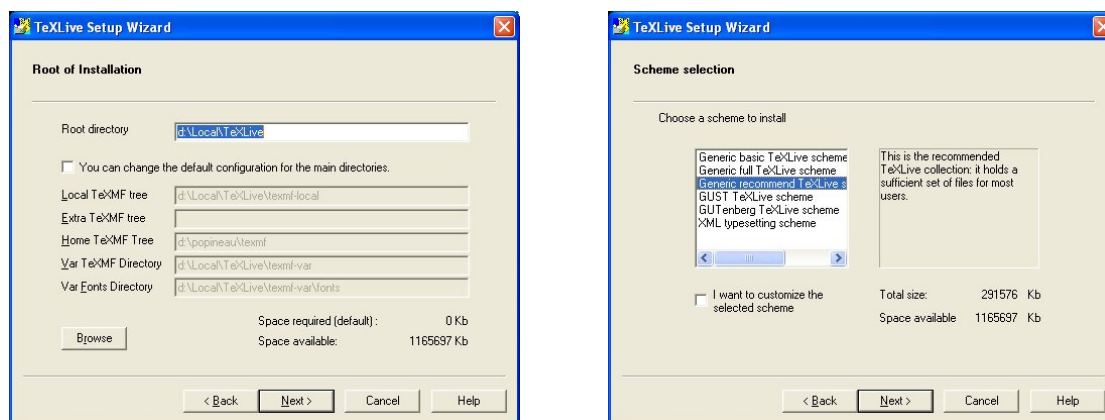
Ostatnia opcja jest dostępna tylko wtedy, gdy włączymy możliwość pobierania plików z Internetu (prawa strona ekranu dialogowego). Możemy tu skonfigurować sposób pobierania plików: wykorzystując bibliotekę wininet.dll Internet Explorera 5x lub bezpośredni dostęp (ftp, http). Wreszcie możemy określić lokalny (*local source directory*) i sieciowy katalog źródłowy plików instalacji (*remote source directory*). Przycisk browse (przeglądaj) pozwala na łatwy wybór zarówno pierwszego, jak i drugiego (adres sieciowy URL, z listy dostępnych repozytoriów) źródła instalacji.

Root of Installation kolejny ekran pozwala na wybór katalogów (rys. 6, po lewej), w których program ma umieścić główną dystrybucję i lokalną konfigurację (z powodu „dziwnych” właściwości Windows 2000 domyślnie proponowany jest dysk C: i katalog \Program Files\TeXLive,

w innych wersjach Windows może być to po prostu katalog `\TeX` dowolnego dysku – uwaga tłumacza). Jedynie wybór głównego katalogu (*Root directory*) jest naprawdę istotny; inne katalogi są ustawiane względem katalogu głównego. W okienku *Extra TeXMF tree* można wskazać dowolny katalog, zgodny ze strukturą TDS, zawierający dodatkowe zasoby \TeX -owe. Z kolei katalog określany przez zmienną `$HOMETEXMF` ustawiany jest domyślnie w zależności od tego, jak nasza instalacja Windows „interpretuje” zmienną `HOME` (w instalacjach jednostanowiskowych jest to po prostu katalog *Moje dokumenty*, w którym można utworzyć zgodne ze strukturą TDS lokalne drzewo `texmf`). Oczywiście można zrezygnować z tworzenia dodatkowych katalogów *Local*, *Extra* i *Home*, usuwając wpisane w okienkach wartości.

Główny i dodatkowe drzewa katalogów instalacji

Wybór predefiniowanych zestawów



Rysunek 6: \TeX Live-Setup: Katalogi instalacji / Predefiniowane zestawy

Ekran ‘**Get TPM**’ nie wymaga interwencji użytkownika. Pliki `.tpm`, które opisują zawartość zestawów i pakietów, są pobierane (niekiedy z Internetu), rozpakowywane (jeśli wymaga tego potrzeba) i analizowane.

‘**Scheme selection**’ menu pozwala wybrać do zainstalowania predefiniowane zestawy pakietów (rys. 6, po prawej): podstawowy, rekomendowany, pełny, albo zalecane zestawy dla grup użytkowników (np. GUTenberg dla frankofonów lub GUST dla Polaków). Gdy wybrano zestaw (*Scheme*) nadal możliwa jest jego dodatkowa specyfikacja (dodanie bądź usunięcie pakietów) po zaznaczeniu okienka obok napisu *I want to customize the selected scheme*. W takim wypadku wyświetlony zostanie ekran wyboru pakietów. Gdy nie włączymy opcji, program przeskoczy od razu do ekranu przeglądowego, wyświetlającego listę pakietów do instalacji.

‘**Packages Selection**’ menu pozwala wybrać do zaistalowania zestawu bądź poszczególne pakiety (opis zestawów i pakietów znajduje się w rozdziale 2.1). Wyboru dokonuje się przez rozwijanie poszczególnych gałęzi zestawów – podobnie jak to ma miejsce w Eksploratorze Windows – i zaznaczaniu/odznaczaniu potrzebnych elementów (rys. 7, po lewej). Można zrezygnować z instalacji pojedynczego zestawu lub pakietu, ale akcja ta powiedzie się jedynie wtedy, gdy wybrany element nie będzie zależny od innych. Przykładowo, nie można odznaczyć zestawu

tex-basic bez uprzedniej eliminacji wszystkich zestawów zależnych. Trzy zestawy (tex-basic, tex-latex i tex-pdftex) są zaznaczone domyślnie jako najczęściej wymagane (zalecane). Zaznaczone jest także instalowanie dokumentacji pakietów. Można ponadto wybrać instalowanie plików źródłowych pakietów makr (Source Files), choć nie są one potrzebne do bezpośredniej pracy. **Uwaga:** Wybór zestawu zawierającego polskie pakiety wymaga zaznaczenia tex-langpolish (jest on domyślnie zaznaczony gdy uprzednio wybrano schemat instalacji dla GUST).

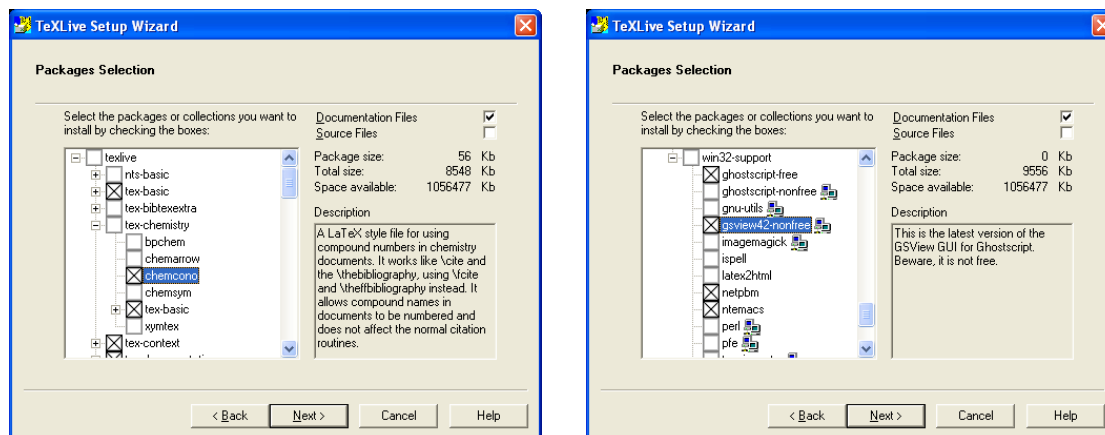
Dodatkowo, w zestawie win32-support, można wybrać do zainstalowania edytor dla T_EX-a (WinShell, TeXshell, PFE lub Emacs) oraz PostScriptową przeglądarkę Ghostscript i inne przydatne programy pomocnicze (rys. 7, po prawej). *Żaden z tych programów nie jest instalowany domyślnie! Nie można także wybrać do instalacji całego zestawu, co zapobiega pochopnemu pobieraniu dużych pakietów z sieci.* Niektóre z wymienionych w zestawie pakietów dostępne są, z powodu ograniczeń licencyjnych, jedynie w Internecie (zaznaczono je ikonką przedstawiającą połączone komputery) i można je zainstalować gdy uprzednio wybrano opcję pobierania programów z sieci. Na CD-ROM-ie dostępny jest stabilny Ghostscript. Warto doinstalować z sieci interfejs graficzny dla Windows, program GSView, (pakiet gsview-nonfree).

Ekran udostępnia ponadto informacje o objętości pakietów oraz wymaganej i dostępnej pojemności dysku.

‘Review Page’ (przegląd wyboru dokonanego przez użytkownika) – ekran ten prezentuje podsumowanie dokonanych wyborów (rys. 8, po lewej). Nadal możliwe jest cofnięcie się do poprzednich menu i zmiana decyzji.

Wybór pakietów

Programy dla Win32



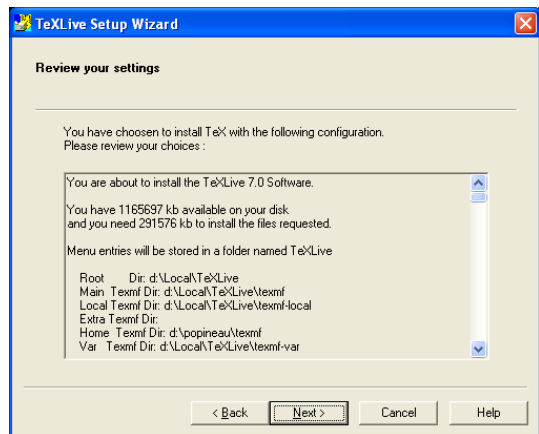
Rysunek 7: Wybór pakietów i Dodatkowe programy dla Win32

Ekran ‘File Copy’ prezentuje postęp procesu instalacji – kopiowanie plików (w miarę potrzeby pobieranych z Internetu i rozpakowywanych), rys. 8, po prawej.

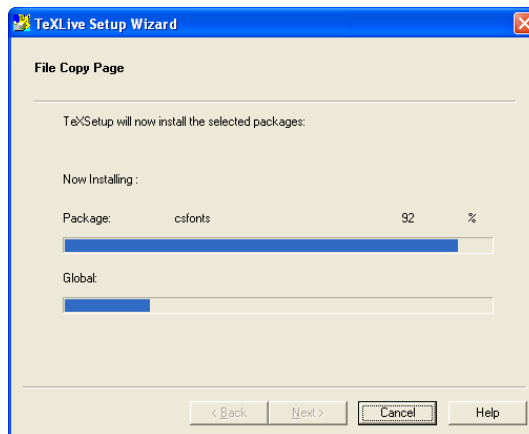
‘Configuration’ wiele pakietów, a także cały system, wymaga uprzedniej konfiguracji (rys 9, po lewej). Na tym etapie generowane są plików formatów, tworzone są bazy danych instalacji, ustawiane

zmienne środowiska itp. Czynności te są czasochłonne i niekiedy wymagają od użytkownika nieco cierpliwości.

Ekran przeglądu



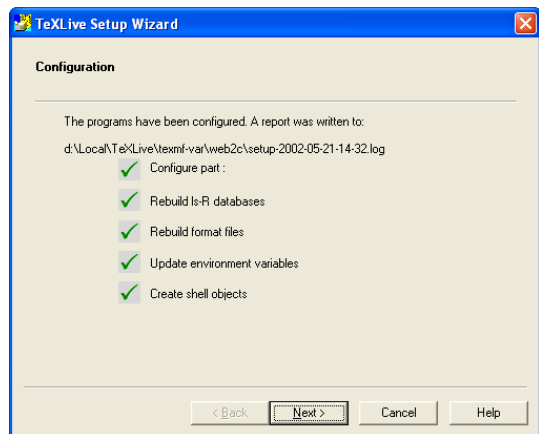
Kopiowanie plików



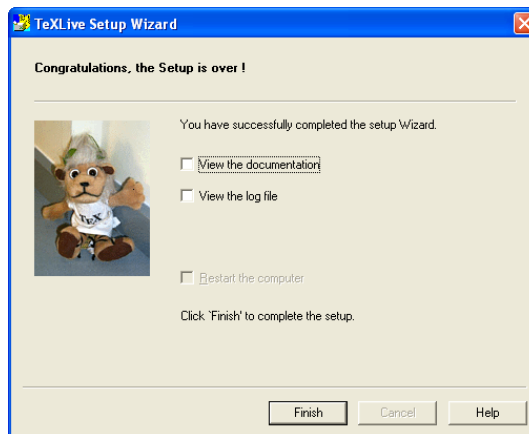
Rysunek 8: Przegląd pakietów i Kopiowanie plików

Ekran **‘Congratulations...’** końcowy ekran (rys. 9, po prawej) pozwala obejrzeć plik .log instalacji. Jeśli zachodzi taka potrzeba (Win9x/ME), proponowany jest restart systemu.

Ekran konfiguracji



Ekran końcowy



Rysunek 9: Ekran konfiguracji i Ekran końcowy

Należy być świadomym, że rozmiar klastra (*cluster*) partycji dysku DOS-owego może mieć radykalny wpływ na rozmiar instalacji T_EX-a. Drzewo z plikami T_EX-owymi zawiera setki małych

plików i często zdarza się, że kompletna instalacja zajmuje do czterech razy więcej miejsca niż objętość samego CD-ROM-u.

5. Zarządzanie systemem T_EX Live w Windows

5.1. Różnice w stosunku do standardowego Web2c w systemach Win32

Web2c działający w systemach Win32 posiada szczególne cechy, o których warto wspomnieć.

Kpathsea – tablice haszowe, tworzone w T_EX Live przez Kpathsea, są bardzo duże. Aby skrócić czas uruchamiania dowolnego programu, wykorzystującego bibliotekę Kpathsea, tablice haszowe zostają umieszczone w pamięci współdzielonej przez procesy. W ten sposób zmniejszono znacznie czas uruchamiania programów wywoływanych kolejno, jeden przez drugi (np. `tex` wywołujący `mpost` wywołujący `tex`). Zmianę tę ukryto przed użytkownikiem, z wyjątkiem sytuacji gdy użyje on parametru `-1` dla diagnostyki działania Kpathsea; można wtedy śledzić wykorzystanie pamięci współdzielonej.

kpsecheck – program ten oferuje kilka opcji, które niezbyt pasowałyby, gdyby znalazły się w programie `kpsewhich`. Pozwala na wylistowanie wszystkich plików, które występują wielokrotnie w dostępnych drzewach katalogów `texmf`. Może to być przydatne, choć często możemy otrzymywać niepożądane komunikaty o dziesiątkach plików typu `README`³. Aby tego uniknąć, należy użyć opcji `-multiple-occurences` wraz z dwiema innymi, pozwalającymi podać wzorce nazw do pominięcia lub włączenia w procesie przeszukiwania (można podać kilka wzorców).

`kpsecheck` raportuje także stan pamięci współdzielonej: używanej i nieużywanej. Warto czasem o tym wiedzieć bo pokazywany status „pamięć w użyciu” może oznaczać, że działa kilka procesów i efekt wykonania polecenia `mktexlsr` zostanie każdorazowo opóźniony do czasu, aż nie będzie pracował żaden program korzystający z Kpathsea.

`kpsecheck` pozwala także określić położenie programu Ghostscript. W systemach Win32 łatwiej jest wielu programom skorzystać z biblioteki DLL programu Ghostscript i znaleźć ją na podstawie rejestrów systemowych, niż na podstawie zmienionej przez użytkownika zmiennej `PATH` (ma ona poza tym ograniczoną pojemność).

Web2c – programy posiadają kilka dodatkowych opcji nieobecnych w standardowych wersjach i jedną działającą nieco inaczej:

- opcja `-fmt` działa inaczej; standardowo ma dwa znaczenia, w zależności od trybu „ini” bądź „vir”. W Win32 jest podobnie: należy załadować plik formatu wyspecyfikowanego przez tę opcję; znaczenie tworzenia formatu w trybie „ini” jest obecnie przypisywane nowej opcji `-job-name`;
- opcja `-job-name` pozwala określić nazwę pliku otrzymywanego w wyniku kompilacji; w trybie normalnym działania programów pozwala na zmianę nazwy wszystkich produkowanych plików (`.dvi`, `.aux` itp.), podczas gdy w trybie „ini” określa nazwę tworzonego formatu;

³Warto tu wspomnieć, że takie pliki powodują „zapychanie” się mechanizmu haszowego Kpathsea; na szczęście biblioteka ta nigdy ich nie wyszukuje.

- opcja `-halt-on-error` pozwala zatrzymać przetwarzanie na pierwszym napotkanym błędzie;
- opcja `-job-time` ustawia czas tworzonych plików na taki sam jak plik przetwarzany;
- `-time-statistics` wyświetla statystyki dotyczące czasu przetwarzania; warto tu wspomnieć, że systemy Win9x, nie będąc prawdziwymi systemami wielozadaniowymi, nie mają prawidłowo działającego pomiaru czasu dla krótkotrwałych procesów, wobec tego wyświetlane wartości są jedynie przybliżone; w NT/2K/XP wyniki są bardziej dokładne; w Win32 brak polecenia `time` znanego z systemów Unix.

5.2. Dodawanie pakietów do instalacji

Po pierwszej instalacji istnieje możliwość doinstalowania pojedynczych pakietów. Z menu systemowego: Start -> Programy -> TeX Live -> Maintenance należy wybrać -> Add TeX Package, co uruchomi program TeXSetup z odpowiednimi parametrami (uruchomienie TeXSetup --help wyświetli dostępne opcje programu). Po wczytaniu plików definiujących pakiety (.tpm) wyświetlony zostanie od razu ekran wyboru zestawów/pakietów (podobnie jak podczas instalacji początkowej). W trybie doinstalowania lista pakietów już zainstalowanych jest porównywana z listą pakietów dostępnych z katalogu źródłowego. Pakiety niezainstalowane wyświetlane są zielonym kolorem, zdezaktualizowane kolorem czerwonym, zaś aktualne, zainstalowane – kolorem czarnym. Dzięki temu można nie tylko dodać nowe komponenty z CD-ROM-u lub z Sieci, ale i zaktualizować posiadane pakiety o dostępne w Sieci nowsze wersje.

Gdy zamierzamy dodać pakiety, których brak w dystrybucji **TeX Live** (lub `fpTeX`), warto instalować je w katalogu specyfikowanym przez zmienną `$TEXMFLOCAL`. W ten sposób zabezpieczymy te pakiety podczas aktualizacji/wymiany oprogramowania **TeX Live**.

Katalog wskazywany przez `$TEXMFLOCAL` jest początkowo pusty. Przykładowo: dodajemy pakiet dla Maple (przetwarzanie symboliczne); pliki stylów należy skopiować do katalogu `c:\Program Files\TeXlive\texmf-local\tex\latex\maple\` zaś pliki dokumentacji do katalogu `c:\Program Files\TeXlive\texmf-local\doc\latex\maple\`

Na zakończenie należy **koniecznie** zaktualizować bazę danych instalacji `ls-R`: albo poprzez wybór menu Start -> Programy -> TeXLive -> Maintenance, albo przez uruchomienie z linii poleceń `mktextlsr`.

5.3. Usuwanie TeX Live z dysku twardego

Procedura deinstalacji oprogramowania dostępna jest z programu `TeXLive.exe`, z menu systemowego TeXLive bądź z Panelu sterowania (Dodaj/usuń programy). Procedura ta usuwa z dysku większość plików **TeX Live**. Ponieważ TeX jest systemem, który sam tworzy wiele plików, niełatwo śledzić wszystkie z nich. Dodatkowe programy posiadają swoje własne procedury deinstalacyjne, wobec tego należy je uruchomić odrębnie. Pliki umieszczone w katalogu wskazywanym przez zmienną `$TEXMFLOCAL` nie będą oczywiście usuwane, a więc jeśli zachodzi taka potrzeba, należy je usunąć samodzielnie.

5.4. Uruchamianie TeXSetup z linii poleceń

Program `TeXSetup.exe` posiada szereg interesujących opcji, które mogą być wyświetlone po uruchomieniu z linii poleceń (okno DOS):


```
c:\>TeXSetup --help
```

Oto opis dostępnych opcji programu:

- automatic-reboot po zakończeniu instalacji restart systemu, bez czekania na potwierdzenie użytkownika;
- dry-run symulacja instalacji, nic nie jest instalowane, tworzony jest tylko plik .log rejestrujący czynności, które program wykonuje podczas zwykłego uruchamiania bez tej opcji;
- quick wykonanie instalacji rekomendowanej, bez interwencji użytkownika;
- net-method (ie5/direct) umożliwienie pobierania z Internetu dodatkowych pakietów (wymagane jest aktywne połączenie, albo bezpośrednim łączem, albo modemem, z wykorzystaniem bibliotek DLL Internet Explorera wersji co najmniej 5; należy pamiętać, że niektóre pobierane pakiety są raczej spore);
- remote-source-directory <url> bazowy adres url dla pakietów pobieranych z sieci;
- local-source-directory <dir> domyślnie jest to katalog nadrzędny w stosunku do położenia programu TeXSetup; gdy zaktualizujemy sam program, opcja ta pozwala nam na jego uruchomienie bezpośrednio z dysku twardego (np. `texsetup --local-source-dir=e:\` gdzie `e:` jest stacją CD);
- installation-directory <dir> główny katalog, w którym instalowany jest system \TeX ; domyślna lokalizacja to `c:\Program Files\TeXLive`;
- texmfmain-directory <dir>
- texmflocal-directory <dir>
- texmfextra-directory <dir>
- texmfhome-directory <dir>
- vartexmf-directory <dir>
- vartexfonts-directory <dir> wszystkie te opcje pozwalają wyspecyfikować katalogi dla różnych drzew texmf i zostaną zarejestrowane w zmiennych specyfikowanych w pliku `texmf.cnf`;
- with-source-files(=yes/no) instalacja plików źródłowych pakietów (domyślnie nie są one instalowane);
- with-documentation-files(=yes/no) instalacja plików z dokumentacją poszczególnych pakietów; uwaga: ogólna dokumentacja będzie zainstalowana nawet bez użycia tej opcji;
- program-folder <folder> nazwa folderu w menu startowym, w ramach którego znajdą się menu instalacji;
- add-package <pkg> opcja używana do dodawania pakietów;
- scheme <pkg> instalacja wyspecyfikowanego zestawu, zamiast zestawu rekomendowanego domyślnie;

- maintenance działanie podobne jak --add-package, bez specyfikowania pakietów do doinstalowania w linii poleceń;
- uninstall opcja usuwania pakietów zainstalowanych z CD-ROM-u; jeśli dodano pliki z innego źródła, utworzono formaty, zainstalowano pomocnicze programy itp., nie będą one usunięte;
- help pokazuje dostępne opcje programu.

5.5. Instalacja sieciowa

Biblioteka Kpathsea rozpoznaje nazwy UNC wobec tego możemy ich użyć dla specyfikacji drzew katalogów TEXMF udostępnianych w sieci komputerowej. Wszystkie pliki z wyjątkiem zawartych w katalogu bin/win32 mogą być współdzielone z instalacjami \TeX lub **TeX Live** systemów Unix/Linux. Oznacza to, że można użyć Samby do zamontowania serwera NT dla stacji roboczej Unix lub odwrotnie: serwer np. Linux może udostępniać zasoby dla stacji Windows. Dostępne są możliwości:

- Instalacja całego oprogramowania na serwerze. Wystarczy dodać binaria dla wymaganych systemów w podkatalogach bin, np. bin/win32 i bin/i386-linux, następnie należy ustawić główne zmienne środowiska. W Win32 można użyć nazw UNC dla wskazania odpowiednich katalogów.
- Instalacja lokalnych kopii binariów i plików formatów. W tym wypadku należy przypisać zmienną \$TEXMFMAIN do katalogu dostępnego na zdalnej maszynie (w sieci), a zmienną \$VARTEXMF do katalogu, który będzie przechowywał lokalne pliki konfiguracyjne i inne generowane pliki.

5.6. Konfiguracja indywidualna

5.6.1. WinShell

Program uruchamiamy z menu Start bądź z ikony/skrótu na pulpicie. Dostosowanie do polskiej wersji Windows wymaga, prócz określenia języka dla menu i komunikatów, użycia menu Opcje (Options) -> Font -> Zmień i wyboru w wyświetlonym oknie Skrypt napisu Europa Środkowa. Dzięki temu wpisywane za pomocą prawego klawisza Alt polskie znaki diakrytyczne będą prawidłowo wyświetlane.

Aby skonfigurować uruchamianie programów otwieramy menu Options -> Program Calls.

- W zakładce DVIWin, w polu „exe-file”, powinno być już wpisane windvi.exe.
- W zakładce Ghostview należy sprawdzić, czy prawidłowo podano ścieżkę dla gsview32.exe: C:\ghostgum\gsview\gsview32.exe (dla wersji 3.6 i nowszych).

Wciskając przycisk OK zatwierdza się zmiany.

Instalacja programu ustawia w Windows skojarzenie plików .tex z uruchamianiem WinShell. Dopóki nie zamierzamy korzystać z innego edytora (jak WinEdt lub Emacs) jest to poprawne.

WinShell niestety nie posiada wbudowanej korekty pisowni, niemniej jednak po zainstalowaniu zestawu tex-extrabin mamy do dyspozycji program Ispell (program korekty pisowni, znany z systemów Unix). Ponieważ znajduje się on w ścieżce przypisanej zmiennej PATH, ispell.exe może być uruchamiany z linii poleceń (okno DOS). Informację o Ispell można znaleźć w: texmf/doc/html/manpages/ispell.html Na CD-ROM-ie dostępny jest słownik Ispell dla języka polskiego; typowe

uruchamianie programu: `ispell -t -b -d polish-win plik.tex` powoduje wczytanie słownika w kodowaniu Windows cp1250.

Ispell można uruchamiać z poziomu WinShell; w podrozdziale 5.9.4 opisano jak dodać uruchamianie polecenia do menu edytora.

Niezły i niedrogi program komercyjny można znaleźć w <http://www.microspell.com>.

WinShell posiada pomoc, dostępną po naciśnięciu symbolu `?` na belce z narzędziami. Więcej informacji na temat WinShell można znaleźć w rozdziale 5.9 na str. 29.

5.6.2. Dvips

Plik konfiguracyjny `config.ps` programu Dvips znajduje się w katalogu `C:\Program Files\TeXlive\texmf-var\dvips\config\`. Plik ten można otworzyć w dowolnym edytorze (np. w WinShell) po to, aby zmienić niektóre parametry działania Dvips. Najważniejsze z nich zgrupowano w blokach deklaracji:

fonty – można zmienić domyślne parametry generowania fontów przez METAFONT-a dla danej drukarki (parametr `M`) i rozdzielczość (parametry `D`, `X`, `Y`). Standardowo fonty generowane są dla drukarek laserowych, zgodnych z Hewlett Packard (tryb `ljfour`) i rozdzielczości 600 dpi. Domyślnie ustawiono wykorzystanie fontów CM (oraz PL) w ich wersji obwiedniowej (Type 1), wobec tego Dvips nie powinien odwoływać się zbyt często do programu `mktexpk`.

drukarka – parametr `'o'` określa nazwę portu/przekierowania wydruku. Gdy nie wpisano nazwy drukarki (portu), wynik pracy będzie zapisywany w pliku PostScriptowym `.ps`. Drukarkę specyfikujemy jednym z poniższych sposobów:

```
o lpt1:
% o | lpr -S server -P myprinter
% o \\server\myprinter
```

mapy fontowe – do standardowej mapy fontowej `psfonts.map` (wiersz: `p psfonts.map`) możemy dołączyć dodatkowe mapy fontowe, np.:

```
p +bsr.map
p +hoekwater.map
```

parametr `p` musi znajdować się w pierwszej kolumnie; nazwy plików z dołączanymi mapami poprzedzamy znakiem `+`; zbędne mapy można wyłączyć, wstawiając na początku wiersza znak komentarza (`%`). Wymienione w przykładzie mapy dotyczą dołączania wersji obwiedniowej fontów CM produkcji Blue Sky Research (`bsr.map`) i dodatkowych fontów logo, wasy, stmaryrd, wykonanych przez Taco Hoekwatera (`hoekwater.map`).

Uwaga: Obecna dystrybucja **T_EX Live** posiada zaimplementowaną po raz pierwszy procedurę automatycznej aktualizacji map fontowych dla Dvips i Pdftex, wykonywaną przez program `updmap` podczas instalacji, a także podczas dodawania pakietów z fontami. Jeśli fonty dodawane są „ręcznie”, należy zmodyfikować plik `updmap.cfg` w katalogu `$VARTEXMF/web2c` i następnie uruchomić program `updmap`.

5.6.3. Pdftex

Konfiguracja programu pdf(la)tex wymaga modyfikacji pliku

C:\Program Files\TeXlive\textmf-var\pdftex\config\pdftex.cfg

W pliku tym możemy deklarować dodatkowe mapy fontowe wczytywane przez program (patrz „Uwaga” zamieszczona powyżej). Stosowana jest tu nieco inna notacja niż w config.ps:

```
map pdftex.map
map +lw35.map
%map +rawfonts.map
```

Ponadto można ustawić inne parametry pracy programu pdftex (rozmiar papieru, stopień kompresji itp.). Parametr output_format z wartością 1 oznacza, że pdftex będzie tworzył domyślnie pliki od razu w formacie pdf (wartość 0 oznacza tworzenie plików .dvi, identycznie jak programem tex).

5.6.4. GSView

Począwszy od wersji zgodnej z Ghostscript 6.50, GSView jest niestety programem shareware, wobec tego nie mógł znaleźć się na CD-ROM-ie.

Parametry pracy programu są ustawiane m.in. w menu Media. W podmenu Display Settings można poprawić (kosztując szybkości pracy) sposób wyświetlania: parametry Text Alpha i Graphics Alpha należy ustawić na 4 bity. W menu Media można też określić rozmiar papieru i jego orientację.

Uwagi dotyczące drukowania zawarto poniżej, w podrozdziale 5.8.

5.6.5. WinDvi

Podczas instalacji TeXSetup.exe ustawia skojarzenie plików .dvi z uruchamianiem programu Windvi.

W menu View ->Options można ustawić niektóre parametry pracy Windvi, np. możliwość uruchamiania programów zewnętrznych (Allow shell), automatyczne odświeżanie zmienionego pliku .dvi (Autoscan), tryb pracy METAFONT-a i rozdzielczość generowanych fontów (MF mode i Pixels per inch), współpracę z Ghostscriptem (Postscript) itp. Po zakończeniu pracy programu parametry zapisywane są w pliku \$HOME/windvi.cnf (lub Moje dokumenty/windvi.cnf). Położenie tego pliku można ustalić po uruchomieniu z wiersza poleceń:

```
c:\>kpsewhich --expand-var $HOME/windvi.cnf
```

W wypadku błędnego ustawienia parametrów pracy Windvi plik konfiguracyjny wystarczy skasować.

5.7. Testowanie

Dla szybkiej kontroli poprawności instalacji należy uruchomić WinShell i otworzyć plik sample2e.tex (znajduje się on w katalogu C:\Program Files\TeXlive\textmf\tex\latex\base\). Na belce z narzędziami wcisnąć ikonę uruchamiania programu latex, następnie wcisnąć ikonę podglądu DVI. Pierwsze uruchomienie windvi zazwyczaj automatycznie generuje wymagane fonty ekranowe. Po kilku sesjach komunikaty dotyczące generowania fontów staną się rzadsze. Warto ponadto wypróbować uruchamianie Dvips i GSView z programu WinShell.

Uwaga: jeśli program latex nie może znaleźć pliku do przetwarzania, naciśnięcie **Ctrl-z** spowoduje natychmiastowe zakończenie pracy.

5.8. Drukowanie

Drukowanie bezpośrednio z programu Windvi odbywa się z wykorzystaniem uniwersalnego sterownika druku systemu Windows. Z definicji powinien być on zgodny z większością drukarek, jednakże ma on tendencję do tworzenia olbrzymich plików kolejek (spool) i niektóre (starsze) wersje Windows zwyczajnie tego nie akceptują. Zaletą sterownika jest jedynie możliwość obsługi grafik BMP i WFM. Stosując metodę druku via sterownik Windows należy prawidłowo ustawić parametry rozdzielczości (patrz podrozdział 5.6.5), w przeciwnym wypadku otrzymany wydruk może być przeskalowany (np. drukując w rozdzielczości 600dpi na drukarce 300dpi otrzymamy niespodziewanie 1/4 zamierzonego rozmiaru wydruku).

Drukowanie znacznie szybsze i bardziej niezawodne zapewni konwersja pliku .dvi do .ps programem Dvips i wykorzystanie GSView. Z menu Plik należy wybrać Drukuj, co wyświetli okno dialogowe z szeregiem opcji. Gdy mamy drukarkę PostScriptową, należy wybrać PostScript printer. Drukowanie na drukarkach bez interpretera PostScript wymaga zaznaczenia opcji Ghostscript device (w sekcji „Print Method” okna) i wybrania przyciskiem po prawej stronie odpowiedniego urządzenia.

5.9. Więcej na temat WinShell

5.9.1. Instalacja poprawek

Ingo de Boer, autor WinShell, udostępnia od czasu do czasu kolejne wersje beta programu. Można je pobrać z <http://www.winshell.de>. Zwykle są to pliki .zip, które należy rozpakować w katalogu, w którym już zainstalowano WinShell (najczęściej w c:\Program Files\WinShell), zezwalając na nadpisanie dotychczasowej zawartości. Program rozpakowujący unzip.exe można znaleźć na CD w katalogu support/.

5.9.2. Użycie plików „projektu”

Wielokrotnie mamy do czynienia z dokumentami składającymi się z wielu części. Najwygodniej gdy są one zapisane w odrębnych plikach. WinShell umożliwia utworzenie pliku projektu (*Project file*), w którym deklarujemy poszczególne części większego dokumentu. Nazwy pliku projektu, traktowanego jako główny edytowany plik (*Main file*), oraz plików części są wyświetlane w lewym oknie edytora. Dwukrotne kliknięcie myszą na nazwie pliku pozwala na szybki wybór i wyświetlenie potrzebnego pliku w głównym oknie edytora. Naciśnięcie ikony LaTeX na belce narzędziowej uruchamia zawsze przetwarzanie głównego pliku, bez względu na to który plik jest aktualnie wyświetlany.

Na belce narzędziowej mamy dodatkowe przyciski pozwalające wyłączyć lub włączyć wyświetlanie okna projektu (po lewej stronie) i okna zapisu sesji (log) – na dole. Gdy plik projektu nie jest używany, lewą część okna można zamknąć, wykorzystując całą szerokość ekranu dla edytowanego pliku.

5.9.3. Drukowanie z WinShell na drukarce PostScriptowej

Ikona Dvips belki narzędziowej uruchamia program, który wynik pracy zapisuje zwykle w pliku .ps. Oglądanie i/lub drukowanie takiego pliku wymaga uruchomienia GSView. WinShell pozwala zdefiniować wygodne wywołanie polecenia dvips, które wynik przetwarzania wyśle od razu do drukarki posiadającej interpreter PostScript. Przykładowo, dla drukarki o nazwie vc1w, postępowanie będzie następujące:

1. Deklaracja danej drukarki w programie:

- otworzyć WinShell, wybrać menu *Opcje* -> *Programy użytkownika*.
- wybrać Tool 1 w spisie po prawej stronie i wypełnić pola po stronie lewej, np.:
Name: Drukuj
exe file: dvips
cmd-line: -D600 %c.dvi -o vclw
skasować zaznaczenie okienka „najpierw DVIPS”
- nacisnąć OK.

2. Dodanie polecenia „Drukuj” do belki narzędziowej:

- wybrać menu *Opcje* -> *Przeglądaj* -> *Dostosuj paski*;
- w okienku *Kategoria* wybrać *Programy użytkownika*;
- wskazać Drukuj i przeciągnąć go myszą na belkę narzędziową, umieszczając w dowolnym miejscu obok innych poleceń;
- w kolejno wyświetlanym oknie dokonać wyboru: „Tylko obrazek” (Image only), „Tylko tekst” lub „Tekst i obrazek”; najprościej wybrać „Tylko tekst” i zatwierdzić guzikiem **OK**. Napis Drukuj ukaże się na belce (bardziej wymagający użytkownicy mogą wybrać opcję „Tylko obrazek”, następnie „Edytuj” i skonstruować sobie indywidualną ikonkę).

Uruchamianie polecenia zostało zdefiniowane. Aby drukować bezpośrednio na drukarce wystarczy teraz nacisnąć guzik z napisem „Drukuj”.

5.9.4. Dodanie Ispell do WinShell

1. Deklaracja polecenia Ispell:

- otworzyć WinShell, wybrać menu *Opcje* -> *Wywołanie programów* -> *User defined*.
- wybrać Tool 1 (lub Tool 2, jeśli Tool 1 zostało już użyte) i wypełnić pola po lewej stronie okna:
Name: Ispell
exe file: ispell
cmd-line: -t -d polish-win %c.tex
skasować zaznaczenie okienek „najpierw LaTeX” i „najpierw DVIPS”
- nacisnąć OK.

2. Dodanie polecenia „Ispell” do belki narzędziowej – analogicznie jak powyżej, dla polecenia „Drukuj”.

Gdy mamy otwarty dokument \TeX -owy, naciśnięcie guzika „Ispell” uruchomi w odrębnym oknie program Ispell, który rozpocznie sprawdzanie pisowni. Niepoprawne (lub nieznane) słowo zostanie wyświetlone, poniżej pojawi się kontekst i czasem sugestie jak naprawić błąd (naciśnięcie klawisza cyfry umieszczonej obok sugerowanego słowa dokona zamiany błędnie napisanego słowa). Inne reakcje na błąd są wyświetlane u dołu ekranu, np. naciśnięcie klawisza spacji

ignoruje dany błąd i wznowia wyszukiwanie. Więcej informacji na temat Ispell można znaleźć w: C:\Program Files\TeXlive\texmf\doc\html\manpages\ispell.html.

Uwaga: poprawione słowa nie ukażą się w oknie WinShell dopóki plik nie zostanie zamknięty (przez naciśnięcie X w prawym górnym rogu okna edytora) i ponownie otwarty (z menu Plik).

5.10. Wskazówki dla użytkowników Win32

5.10.1. Różne wcielenia Win32

To, co nazywamy Win32 nie jest właściwie systemem operacyjnym a jedynie zbiorem funkcji, z których wiele (ok. 12000) można użyć do pisania programów dla różnych systemów operacyjnych rodziny Windows. Spotykamy różne wersje Windows:

- Win95 i Win98, które *nie są prawdziwymi systemami wielozadaniowymi i wielowątkowymi*. Są to ostatnie (na szczęście) metamorfozy systemu DOS, co można sprawdzić podczas startu systemu, wymuszając ładowanie tylko wiersza poleceń: polecenie `ver` zwróci wtedy np. napis „MS-DOS 7.0”.
- NT, nowy system operacyjny, napisany od podstaw, wyposażony w prawdziwe możliwości wielozadaniowości i inne zaawansowane cechy.
- Windows 2000, oparty na NT, z wszystkimi cechami zewnętrznymi Win98.
- Windows XP, występujący w odmianach Personal i Pro, jest ostatnim krokiem połączenia cech obu linii produktów Win9x i NT.

Win9x mają możliwość uruchamiania jednocześnie programów 16- i 32-bitowych, ale ponieważ sam system nie jest w całości napisany dla trybu 32-bitowego i nie posiada ochrony pamięci, programy 16-bitowe mogą nadpisywać część pamięci wykorzystywanej przez system! Niektóre elementy systemu, jak GDI (*Graphical Device Interface*) zarządzają w sposób ograniczony zasobami (bitmapy, fonty itp.) gdy wszystkie uruchomione programy jednocześnie korzystają z tych zasobów. Wszystkie nagłówki bitmap dostępnych w tym samym czasie nie mogą przekroczyć granicy 64KB. Jest to przyczyna spotykanego często gwałtownego spadku wydajności systemu i jego zawieszania.

NT i W2K i XP nie posiadają takich ograniczeń. Są to systemy w pełni wielozadaniowe i z ochroną pamięci. Wydają się bardziej stabilne ponieważ lepiej zarządzają pamięcią, posiadają lepszy system plików itp.

5.10.2. Wiersz poleceń

Wielu użytkowników może zadać pytanie: „po co używać wiersza poleceń w systemie Windows?”. Dobre pytanie! Problem ma naturę bardziej ogólną. Otóż nie wszystkie zadania daje się wykonać jedynie przy użyciu interfejsu graficznego (GUI). Wiersz poleceń dostarcza wielu możliwości rasowego programowania (zakładając posiadanie przyzwoitego interpretera poleceń).

T_EX jest programem działającym *wsadowo* (*batch*), a nie interakcyjnie: wylicza najlepszy wygląd każdej strony, rozwiązuje odesłania itp. Taki charakter pracy możliwy jest jedynie podczas globalnego przetwarzania dokumentu, nie ma tu możliwości interakcji. Oznacza to, że program T_EX powinien być używany z linii poleceń. W istocie nie jest to takie złe. Programy działające wsadowo, przeznaczone do skomplikowanych zadań, są zwykle lepiej opracowane i pozbawione błędów, ponieważ nie są związane

z graficznym interfejsem. Do uruchamiania takich programów można zawsze opracować graficzny interfejs przekazujący zadanie do wykonania i inne parametry. Przykładem takiego właśnie działania może być graficzny edytor (patrz 5.6.1).

Niemniej jednak częste są sytuacje, gdy przydaje się możliwość uruchamiania programów bezpośrednio z linii poleceń, np. diagnostyka błędów w wypadku wystąpienia problemów (patrz 7.2.4).

Win9x linia poleceń dostępna jest z menu Start->Programy->Tryb MS DOS albo Start->Uruchom: należy tu wpisać `command.com`.

NT, 2K, XP dostęp do linii poleceń z menu Start->Akcesoria-> Command Prompt; można także w menu Start->Uruchom wpisać `cmd.exe` co uruchomi interpreter poleceń (wyjaśnia to dlaczego trudno w systemach opartych na NT mówić o „oknie DOS”).

5.10.3. Separatory ścieżek

API Win32 obsługuje zarówno `/`, jak i `\` jako separatory ścieżek. Niestety interpretery poleceń obsługują wyłącznie `\`, a więc odwołując się do programu w linii poleceń możemy użyć tylko tego znaku (wynika to z potrzeby kompatybilności: znak `/` używany jest często jako prefiks dodatkowego parametru). Gdy nazwa ścieżki jest używana przez program oba separatory mogą być stosowane wymiennie, a nawet mieszane.

Nie należy się dziwić gdy często spotkamy ścieżki pisane w konwencji Unix (a więc separowane znakiem `/`). `fpTEX` jest implementacją Web2c i z zasady musi być zgodny między różnymi platformami. Dlatego właśnie np. wszystkie pliki konfiguracyjne stosują konwencję uniksową notacji ścieżek.

5.10.4. System plików

Najgorszą cechą Win9x z punktu widzenia systemu T_EX jest prawdopodobnie system plików zwany FAT. T_EX używa bardzo wielu niewielkich plików (1–3KB) i na współczesnych dyskach wielkiej pojemności każdy taki plik potrafi zająć nawet 32KB. Wykorzystanie dysku jest więc bardzo nieefektywne zaś instalacja wszystkich plików z CD-ROM-u może zająć wielokrotnie więcej miejsca niż łączna objętość plików.

Bardziej nowoczesne systemy plików FAT32 i NTFS dostępne są w nowszych wersjach Windows i są one pozbawione ujemnych cech FAT: minimalna jednostka alokacji dla pliku to zwykle 4KB, choć w systemie NTFS można ją zredukować nawet do 512 bajtów.

5.10.5. Jak dodać katalog do zmiennej PATH

Większość programów korzysta ze zmiennych, którym można przypisać pewne wartości. Zestaw takich przypisań (par zmienna-wartość) nazywamy środowiskiem. Każdy uruchomiony program korzysta z kopii środowiska i może w ramach tej kopii zmieniać przypisania niektórych zmiennych (nie są one dostępne dla innych uruchomionych równolegle programów).

PATH jest specjalną zmienną wykorzystywaną do lokalizowania programów w systemie. Poszczególne wersje Windows mają różne procedury deklarowania PATH:

Windows 95/98 zmienna PATH jest tu deklarowana w pliku `autoexec.bat` (znajduje się on w głównym katalogu dysku startowego). Po wczytaniu pliku do edytora (np. Notatnik, WinShell) należy odszukać wiersz z napisem `PATH=` i z szeregiem ścieżek oddzielanych separatorem `;` (średnik). Można dopisać kolejne ścieżki wskazujące położenie programów, np.


```
PATH=c:\windows;c:\windows\system;c:\"Program Files"\TeXlive\bin\win32
```

Proszę zauważyć, że ścieżki zawierające spację wymagają ujęcia w parę znaków ". Poczynione zmiany aktywne będą po restarcie systemu.

Windows ME należy uruchomić specjalny program `c:\windows\system\msconfig.exe` i wybrać opcję „Środowisko” (‘Environment’). Po dodaniu lub modyfikacji zmiennej należy ponownie uruchomić system.

Windows NT/2K/XP należy wybrać menu Start -> Settings -> Panel Sterowania, w opcji „System” otworzyć zakładkę „Środowisko” i tu modyfikować deklaracje zmiennych, zgodnie z posiadanymi uprawnieniami. Istniejącą zmienną PATH modyfikujemy naciskając lewy klawisz myszy na napisie PATH. Pole „Wartość” (Value) pokaże listę katalogów rozdzielonych separatorem ;. Należy tu dopisać katalog zawierający programy środowiska T_EX-owego (np. `c:\Program Files\TeXlive\bin\win32`). Gdy brak jest zmiennej PATH, należy kliknąć pole „Zmienna”, wpisać PATH, przenieść kursor do pola „Wartość” i wpisać tu odpowiedni katalog.

Uwaga: przed zamknięciem okna dialogowego należy zatwierdzić zmiany przyciskiem „Zastosuj”.

Dokonane zmiany można sprawdzić otwierając konsolę i uruchamiając polecenie:

```
set NAZWA_ZMIENNEJ
```

co wyświetli aktualnie aktywne wartości dla danej zmiennej.

5.10.6. „Maszyny” T_EX

Z dokumentacji Web2c można wyczytać, że różnie nazwane programy wykorzystują tę samą „maszynę”, np. `tex.exe` i `latex.exe` są dokładnymi kopiami tego samego programu, ale uruchamiają się z różnymi plikami formatu, zgodnie z nazwą podaną w wywołaniu.

W systemach Unix cecha ta jest zaimplementowana poprzez tzw. dowiązanie symboliczne, co pozwala na zaoszczędzenie wiele miejsca na dysku: ten sam program uruchamiany jest z wieloma formatami.

Interfejs aplikacji systemów Windows (API) nie zna pojęcia dowiązania. Aby niepotrzebnie nie zajmować miejsca na dysku, zastosowano tu pomysł umieszczenia głównej „maszyny” T_EX w pliku dynamicznie wiązanej biblioteki (DLL). Z kolei programy uruchomieniowe (.exe) umieszczono w niewielkich plikach przekazujących przez nazwę parametr wywołania (nazwę formatu). Na dysku znajdziemy zatem:

13/05/2002	17:06	3 584	latex.exe
13/05/2002	17:06	266 240	tex.dll
13/05/2002	17:06	3 584	tex.exe

Każdy z tych *identycznych* programów (`tex.exe`, `latex.exe`, `mex.exe` itp.) korzysta ze wspólnego pliku `tex.dll`. Podobne rozwiązanie zastosowano dla grupy programów `mktex*.exe`, z których każdy jest powiązany z biblioteką `mktex.dll`. W dystrybucji dostępny jest program `lnexe.exe`, który tworzy w systemach Win32 odpowiedniki dowiązań systemu Unix.

5.11. W razie problemu

5.11.1. Co zrobić gdy latex nie może znaleźć potrzebnych plików?

Należy użyć programu kpsewhich, który jest narzędziem do diagnostyki wszelkich problemów związanych ze znajdowaniem plików (patrz 7.2.4 na str. 47).

5.11.2. Co zrobić gdy instalacja nadal nie działa prawidłowo?

Należy odpowiedzieć sobie na szereg pytań:

1. czy program `tex.exe` jest w ścieżce wymienionej w zmiennej `PATH`?
2. czy zmienna `TEXMFCONF` jest prawidłowo ustawiona (standardowo `c:/Program Files/TexLive/texmf-var/web2c`)?
3. czy są jakieś błędy w pliku `.log` utworzonym przez program `TeXSetup.exe`? W pliku `.log` należy wyszukać słowo „Error”.
4. diagnostyka błędów omówiona została w rozdziale 7.2.4 na str. 47.
5. opis problemów i ewentualne rozwiązanie oraz poprawki można znaleźć pod adresem <http://www.tug.org/tex-live.html>.
6. dystrybucja dla Windows zawarta na płycie jest w zasadzie identyczna z dystrybucją `fpTeX`, wobec tego można odwiedzić stronę <http://www.fptex.org>; warto też zapisać się na listę dyskusyjną `fpTeX` (<http://www.tug.org/mailman/listinfo/fptex>).

Oprogramowanie zawarte na **TeX Live** jest bardzo złożone i składa się z ponad 250 programów i około 40000 plików różnego źródła. Niesłuchanie trudno przewidzieć wszystkie możliwe przyczyny problemów, niemniej jednak dołożymy wszelkich starań aby pomóc je rozwiązać.

5.12. Kompilacja plików źródłowych

Wszystkie pliki źródłowe programów zawarte są w skompresowanym archiwum `source/source.tar.bz2` na CD-ROM-ie. Kompilacja w środowisku Windows wymaga:

- Windows 2000;
- Microsoft Visual Studio .Net;
- zestawu narzędzi Unix (`sed`, `grep`, `gawk` itp.), a także Perl, Flex i Bison;
- dostosowania ścieżek w pliku `win32/make/common.mak`, zgodnie z konkretną instalacją;
- dostosowania ścieżek w skrypcie perlowym `win32/perl/build.pl`;
- uruchomienia kompilacji z katalogu `win32/` poleceniem:

```
c:\texlive\source\win32>perl ./perl/build.pl --install --log=install.log
```

Uczynienie tego procesu łatwiejszym i bardziej klarownym wymaga jeszcze sporo pracy.

6. Budowa systemu na nowej platformie Unixowej

Jeżeli mamy do czynienia z platformą, dla której nie są dostarczone binaria, trzeba będzie skompilować \TeX -a oraz towarzyszące mu programy. Nie jest to takie trudne jak się wydaje. Wszystko, czego potrzebujemy, znajduje się na CD-ROM-ie w katalogu `source`.

Najpierw należy zainstalować główne drzewo plików \TeX -owych z CD-ROM-u **\TeX Live** (najlepiej jest wykonać instalację podstawową, bez wybierania binariów systemowych).

6.1. Warunki wstępne

Do skompilowania całego \TeX -a i jego programów pomocniczych potrzeba około 100 megabajtów wolnego miejsca na dysku. Potrzebny jest także kompilator ANSI C, oprogramowanie `make`, skaner leksykalny oraz generator parserów. Programy użytkowe GNU (`gcc`, GNU `make`, `m4`, `flex`, `bison`) są najlepiej przetestowanymi na różnych platformach. Prawdopodobnie powinny pracować `gcc-2.7.*`, `flex-2.4.7` i GNU `make-3.72.1` lub nowszy. Istnieje możliwość pracy z innymi kompilatorami C i programami `make`, wymaga to jednak dobrego zrozumienia zasad tworzenia programów Unixowych, aby poradzić sobie z problemami. W wyniku wykonania polecenia `uname` powinniśmy otrzymać jakąś rozsądną wartość.

6.2. Konfiguracja

Najpierw należy rozpakować źródła ze skompresowanego pliku `tar` z katalogu `source` na twardy dysk, a potem przejść do katalogu, do którego zostały skopiowane. Teraz trzeba zdecydować się, gdzie będzie główny katalog instalacji, np. `/usr/local` lub `/usr/local/TeX`. Naturalnie, należy użyć tej samej lokalizacji, która została wykorzystana przy instalowaniu głównego drzewa \TeX -owego.

Uruchomić `configure` z linii poleceń:

```
>> ./configure --prefix=/usr/local/TeX
```

Katalog „`prefix`” to ten, w którym zainstalowane zostało główne drzewo \TeX -owe. Zostanie wykorzystany następujący układ katalogów (`$TEXDIR` oznacza wybrany katalog):

<code>\$TEXDIR/man</code>	strony manuala Unixowego,
<code>\$TEXDIR/share/texmf</code>	główne drzewo \TeX -owe, zawierające makra itp.,
<code>\$TEXDIR/info</code>	podręczniki w formacie GNU info,
<code>\$TEXDIR/bin/\$PLATFORM</code>	binaria

Można uniknąć stosowania „`share/`” przed katalogiem `texmf`, ponieważ `$TEXDIR/share/texmf` i `$TEXDIR/texmf` są wykrywane automatycznie przez skrypt `configure`. W przypadku wybrania innego katalogu, trzeba wyszczególnić ten katalog przez parametr `--datadir` skryptu `configure`.

Jeżeli nie chcemy używać katalogu `$PLATFORM` (tj. umieścić binaria bezpośrednio w `$TEXDIR/bin`), w wywołaniu skryptu `configure` należy użyć parametru `--disable-multiplatform`.

Więcej dostępnych parametrów (np. pomijanie pakietów opcjonalnych, takich jak Ω czy ε - \TeX) można zobaczyć uruchamiając `./configure --help`.

6.3. Uruchamianie `make`

Po upewnieniu się, że nie jest ustawiona zmienna powłoki `noclobber`, należy napisać

```
>> make world
```

i wygodnie poczekać...

Mogłoby też być użyteczne zapisanie w pliku .log całego procesu kompilacji:

```
>> sh -c "make world >world.log 2>&1" &
```

Zanim stwierdzimy, że wszystko jest w porządku, warto sprawdzić czy w pliku .log nie ma komunikatów błędów (kiedy polecenie zwraca kod błędu, GNU make zawsze używa sekwencji „Error:”) i sprawdzić czy utworzone zostały wszystkie binaria:

```
>> cd /usr/local/TeX/bin/i586-pc-linux-gnu
```

```
>> ls | wc
```

Oczekiwany rezultat jest 213. Polecenie `make world` jest równoważne z `make all install strip`. Jeśli do uruchamiania `make install` potrzebne są specjalne uprawnienia, można uruchomić dwie oddzielne sesje `make`:

```
>> make all
```

```
>> su
```

```
>> make install strip
```

6.4. Końcowe kroki konfiguracyjne

Należy uaktualnić zmienną `PATH` o nazwę katalogu zawierającego właśnie zainstalowane binaria (np. `/usr/local/TeX/bin/mips-sgi-irix6.3`); podobnie postąpić trzeba z `MANPATH` i `INFOPATH` i dołączyć do nich odpowiednie nazwy podkatalogów, tj. `$TEXDIR/man` oraz `$TEXDIR/info`.

Program `texconfig` pozwala na ustawienie domyślnych wzorców przenoszenia wyrazów, rozmiaru papieru, poleceń do druku, trybu `METAFONT`-a itd. Polecenia tego użyć można interaktywnie. Aby zobaczyć jakie funkcje program udostępnia wystarczy napisać

```
>> texconfig help
```

Przykładowo, jeżeli nie używamy formatu A4, możemy ustawić jako format domyślny „lettersize” pisząc:

```
>> texconfig dvips paper letter
```

```
>> texconfig xdvi paper us
```

7. Instrukcja obsługi systemu Web2c

Web2c to zestaw programów związanych z \TeX -em, tj. sam \TeX , `METAFONT`, `METAPOST`, `BibTeX` itp. Oryginalna implementacja wykonana została przez Tomasza Rokickiego, który w roku 1987 stworzył pierwszy system \TeX -to-C, adaptując pliki wymiany (*change files*) pod Unix-em (pierwotnie były one dziełem Howarda Trickey’a oraz Paveła Curtisa). W czasie gdy Tim Morgan zajmował się systemem, jego nazwa została zmieniona na Web-to-C. W 1990 roku prace nad projektem przejął Karl Berry wraz z dziesiątkami współpracowników, a w roku 1997 pałeczkę przejął Olaf Weber. Ostatnim rezultatem jest Web2c w wersji 7.3, udostępniony w marcu 1999 roku, stanowiący podstawę niniejszej płyty CD-ROM **TeX Live**. Nasza wersja ma nieco uaktualnień i jest identyfikowana jako 7.3.7.

Web2c 7.3 działa na platformach systemowych takich jak Unix, Windows 3.1, 9x/NT/2K/XP, DOS, Amiga i innych. System wykorzystuje oryginalne źródła \TeX -owe autorstwa Donalda Knutha oraz inne programy napisane w web i tłumaczy je na kod źródłowy C. Ponadto system udostępnia spory zestaw makr i funkcji stworzonych dla zwiększenia funkcjonalności oryginalnych zasobów oprogramowania związanego z \TeX -em. Podstawowymi składnikami systemu są:

bibtex tworzenie spisów bibliograficznych;
dmp konwersja troff do MPX (rysunki METAPOST-owe);
dvcopy modyfikowanie pliku DVI;
dvitomp konwersja DVI do MPX (rysunki METAPOST-owe);
dvitype konwersja DVI na plik tekstowy (ASCII);
gftodvi zamiana fontu GF na plik DVI;
gftopk zamiana fontów w formacie GF na font spakowany PK;
gftype zamiana fontu GF na plik tekstowy (ASCII);
makempx skład etykiet METAPOST-owych;
mf generowanie fontów bitmapowych w formacie GF;
mft skład plików źródłowych METAFONT-a;
mpost tworzenie rysunków oraz diagramów technicznych;
mpto ekstrakcja etykiet METAPOST-owych;
newer porównywanie czasów modyfikacji;
patgen tworzenie wzorców przenoszenia wyrazów;
pktogf zamiana fontów w formacie PK na fonty GF;
pktype zamiana fontu PK na plik tekstowy (ASCII);
pltotf konwersja tekstowej listy właściwości do TFM;
pooltype wyświetlanie web-owych plików pool;
tangle konwersja web do języka Pascal;
tex skład tekstu;
tftopl konwersja TFM do tekstowej listy właściwości (PL);
vftovp konwersja fontów wirtualnych do wirtualnej listy właściwości (VPL);
vptovf konwersja wirtualnej listy właściwości do fontów wirtualnych;

weave konwersja web do \TeX -a.

Dokładny opis funkcji oraz składni tych programów zawarty jest w dokumentacji poszczególnych pakietów samego Web2c. Jednak do optymalnego wykorzystania instalacji Web2c pomocna będzie znajomość kilku zasad rządzących całą rodziną programów.

Wszystkie programy obsługują standardowe opcje GNU:

--help podaje podstawowe zasady użytkowania;

--verbose podaje dokładny raport z działania programu;

--version podaje informację o wersji, po czym kończy działanie programu.

W celu lokalizacji plików programy Web2c używają biblioteki do przeszukiwania ścieżek zwanej Kpathsea. Dla optymalizacji przeszukiwania \TeX -owego drzewa podkatalogów biblioteka ta używa kombinacji zmiennych środowiskowych oraz kilku plików konfiguracyjnych. Web2c potrafi obsługiwać jednocześnie więcej niż jedno drzewo podkatalogów, co jest użyteczne w przypadku, gdy chce się przechowywać standardową dystrybucję \TeX -a jak i lokalne rozszerzenia w dwóch różnych drzewach katalogów. Aby przyspieszyć poszukiwanie plików, katalog główny każdego drzewa ma swój plik `ls-R`, zawierający pozycje określające nazwę i względną ścieżkę dla wszystkich plików zawartych w tym katalogu.

7.1. Przeszukiwanie ścieżek przez Kpathsea

Opiszemy najpierw ogólny mechanizm przeszukiwania ścieżek przez bibliotekę Kpathsea.

Tym, co nazywamy *ścieżką przeszukiwania* jest, rozdzielona dwukropkami lub średnikami, lista *elementów ścieżki*, które zasadniczo są nazwami podkatalogów. Ścieżka przeszukiwania może pochodzić z (kombinacji) wielu źródeł. Aby odnaleźć plik „my-file” w ścieżce „./dir”, Kpathsea sprawdza każdy element ścieżki w następującej kolejności: najpierw ./my-file, potem /dir/my-file, zwracając pierwszy odnaleziony (lub możliwie wszystkie).

Aby optymalnie zaadaptować się do konwencji wszystkich systemów operacyjnych, na systemach nieunixowych Kpathsea może używać jako separatorów nazw ścieżek znaków innych niż dwukropek („.”) oraz „ciach” („/”).

W celu sprawdzenia konkretnego elementu p ścieżki, Kpathsea najpierw sprawdza, czy zbudowana wcześniej baza danych (patrz „Baza nazw plików” na stronie 42) odnosi się do p , tj. czy baza danych znajduje się w podkatalogu z prefiksem p . Jeżeli tak, to specyfikacja ścieżki jest porównywana z zawartością bazy.

Jeśli baza danych nie istnieje, lub nie odnosi się do danego elementu ścieżki, albo nie zawiera elementów zgodnych, przeszukiwany jest system plików (jeżeli nie zostało to zabronione przez specyfikację rozpoczynającą się od „!” oraz jeżeli poszukiwany plik musi istnieć). Kpathsea konstruuje listę podkatalogów, które korespondują z danym elementem ścieżki, a następnie sprawdza w każdym z nich, czy nie ma tam poszukiwanego pliku.

Warunek mówiący, że „plik musi istnieć” dotyczy np. plików „.vf” i plików dołączanych \TeX -owym poleceniem `\openin`. Takiego pliku może nie być (np. `cmr10.vf`), błędne byłoby zatem poszukiwanie go na dysku. Jeśli więc zapomnisz o aktualizacji `ls-R` po instalacji nowego pliku „.vf”, nie zostanie on odnaleziony. Każdy element ścieżki sprawdzany jest w następującej kolejności: najpierw w bazie danych, potem na dysku. Jeżeli plik się znajdzie, przeszukiwanie zostanie zatrzymane i zwrócony zostanie wynik.

Ponieważ najprostszym i najbardziej powszechnym elementem ścieżki jest nazwa katalogu, Kpathsea korzysta z dodatkowych możliwości w przeszukiwaniu ścieżek: wielowarstwowych wartości domyślnych, zmiennych środowiskowych, wartości pliku konfiguracyjnego, lokalnych podkatalogów użytkownika oraz rekursywnego przeszukiwania podkatalogów. Można więc powiedzieć, że Kpathsea *rozwija* element ścieżki, tzn. transformuje wszystkie specyfikacje do podstawowej nazwy lub nazw katalogów. Jest to opisane w kolejnych akapitach, w kolejności, w jakiej ma to miejsce.

Trzeba zauważyć, że jeżeli nazwa poszukiwanego pliku jest absolutna lub jawnie względna, tj. zaczyna się od „/” lub „./” lub „../”, Kpathsea ogranicza się do sprawdzenia, czy ten plik istnieje.

7.1.1. Źródła ścieżek

Nazwa przeszukiwanej ścieżki może pochodzić z wielu źródeł. Oto kolejność, w jakiej Kpathsea ich używa:

1. Zmienna środowiskowa ustawiana przez użytkownika, np. `TEXINPUTS`. Zmienne środowiskowe z dołączoną kropką i nazwą programu zastępują inne, np. jeżeli „`latex`” jest nazwą uruchomionego programu, wtedy zamiast `TEXINPUTS` wykorzystana zostanie zmienna `TEXINPUTS.latex`.
2. Plik konfiguracyjny konkretnego programu, np. linia „`S /a:/b`” w pliku `config.ps` programu `dvips`.
3. Plik konfiguracyjny Kpathsea `texmf.cnf`, zawierający taką linię, jak „`TEXINPUTS=/c:/d`” (patrz poniżej).
4. Wartości domyślne dla uruchamianych programów.

Każdą z tych wartości dla danej ścieżki przeszukiwania można zobaczyć, używając opcji diagnostyki błędów (patrz „Diagnostyka błędów” na stronie 47).

7.1.2. Pliki konfiguracyjne

Kpathsea szuka ścieżek przeszukiwania i innych definicji w *plikach konfiguracyjnych* o nazwach `texmf.cnf`. Ścieżka przeszukiwania używana do znajdowania tych plików określana jest zmienną `TEXMFCNF` (domyślnie taki plik znajduje się w podkatalogu `texmf/web2c`). Czytane będą *wszystkie* pliki `texmf.cnf` w ścieżce przeszukiwania, a definicje we wcześniejszych plikach zastąpią te w późniejszych. Tak więc w ścieżce `:$TEXMF`, wartości pochodzące z `./texmf.cnf` zastąpią te z `$TEXMF/texmf.cnf`.

Czytając zamieszczony poniżej opis formatu pliku `texmf.cnf` proszę także porównać treść załącznika 11, który zaczyna się na stronie 53, i w którym przedstawiono zawartość pliku `texmf.cnf` niniejszej dystrybucji.

- Komentarze zaczynają się od „%”, a kończą na końcu linii.
- Linie puste nie są brane pod uwagę.
- Znak \ na końcu linii działa jako znak kontynuacji, tzn. oznacza, że kolejna linia jest kontynuacją bieżącej. Spacja na początku kolejnej linii nie jest ignorowana.
- Pozostałe linie mają postać:

zmienna [.program] [=] *wartość*

gdzie „=” i otaczające spacje są opcjonalne.

- „*zmienna*” zawierać może dowolne znaki poza spacją, „=”, lub „.”, najbezpieczniej jest jednak używać znaków z zakresu „A-Za-z”.
- Zapis „.program” ma zastosowanie w wypadku, gdy uruchamiany program nosi nazwę *program* lub *program.exe*. Pozwala to różnym odmianom T_EX-a posiadać różne ścieżki przeszukiwania.
- „*wartość*” zawierać może dowolne znaki poza „%” oraz „@”. Nie można używać konstrukcji „\$*zmienna.program*” po prawej stronie. Zamiast tego trzeba zastosować dodatkową zmienną. Średnik „;” użyty w „*wartość*” zamieniany jest na „:” jeżeli systemem operacyjnym jest Unix; umożliwia to posiadanie jednego pliku *texmf.cnf* dla systemów Unix, MSDOS oraz Windows.
- Wszystkie definicje czytane są zanim cokolwiek zostanie rozwinięte, tak więc do zmiennych odwoływać się można przed ich zdefiniowaniem.

Fragment pliku konfiguracyjnego ilustrujący większość powyższych cech pokazany jest poniżej:

```
TEXMF          = {$TEXMFLOCAL;!!$TEXMFMAIN}
TEXINPUTS.latex = .;$TEXMF/tex/{latex;generic;}//
TEXINPUTS.fontinst = .;$TEXMF/tex//;$TEXMF/fonts/afm//
% e-TeX related files
TEXINPUTS.elatex = .;$TEXMF/{etex;tex}/{latex;generic;}//
TEXINPUTS.etex   = .;$TEXMF/{etex;tex}/{eplain;plain;generic;}//
```

7.1.3. Rozwijanie ścieżek

Kpathsea rozpoznaje w ścieżkach przeszukiwania pewne specjalne znaki oraz konstrukcje, podobne do tych, dostępnych w powłokach Unixa. Jako ogólny przykład: złożona ścieżka ~\$USER/{foo,bar}//baz rozwija się do wszystkich podkatalogów pod katalogami foo i bar w katalogu głównym \$USER, które zawierają katalog lub plik baz. Rozwinięcia te opisane są w poniższych podrozdziałach.

7.1.4. Rozwijanie domyślne

Jeżeli ścieżka przeszukiwania największego uprzywilejowania (patrz „Źródła ścieżek” na stronie 39) zawiera *dodatkowy dwukropek* (np. na początku, na końcu lub podwójny) to Kpathsea wstawia w tym miejscu następną zdefiniowaną w hierarchii uprzywilejowania ścieżkę przeszukiwania. Jeżeli ta wstawiona ścieżka ma dodatkowy dwukropek, dzieje się dalej to samo. Przykładowo, jeżeli ustawić zmienną środowiskową

```
>> setenv TEXINPUTS /home/karl:
```

oraz wartość TEXINPUTS pobraną z *texmf.cnf*

```
.: $TEXMF//tex
```

końcową wartością użytą w przeszukiwaniu będzie:

```
/home/karl:.$TEXMF//tex
```

Ponieważ nieużytecznym byłoby wstawiać wartość domyślną w więcej niż jedno miejsce, Kpathsea zmienia tylko jeden dodatkowy „:” i pozostawia inne bez zmian; najpierw szuka dwukropków na początku linii, potem na końcu, a następnie podwójnych.

7.1.5. Rozwijanie nawiasów

Użyteczną cechą jest możliwość rozwijania nawiasów, co oznacza, że np. $v\{a,b\}$ rozwija się do $vaw:vbw$. Możliwe jest zagnieżdżanie nawiasów. Funkcja ta może być użyta do zaimplementowania różnych hierarchii T_EX-owych przez przypisanie listy nawiasów do \$TEXMF. Przykładowo, w pliku `texmf.cnf` znaleźć można następującą definicję:

```
TEXMF = {$HOMETEXMF,$TEXMFLOCAL,!!$VARTEXMF,!!$TEXMFMAIN}
```

Używając jej można następnie napisać coś w rodzaju

```
TEXINPUTS = .;$TEXMF/tex//
```

co oznacza, że po szukaniu w katalogu bieżącym przeszukane będą kolejno *tylko* \$HOMETEXMF/tex, \$TEXMFLOCAL/tex, \$VARTEXMF/tex i \$TEXMFMAIN/tex (dwie ostatnie ścieżki *wyłącznie* na podstawie zawartości pliku `ls-R`). Jest to wygodny sposób dla uruchamiania dwóch równoległych struktur T_EX-owych, jednej „zamrożonej” (np. na CD-ROM-ie), a drugiej ciągle uaktualnianej nowymi, pojawiającymi się wersjami. Używając zmiennej \$TEXMF we wszystkich definicjach, jest się pewnym, że najpierw przeszukiwane jest drzewo uaktualnione.

7.1.6. Rozwijanie podkatalogów

Dwa lub więcej kolejnych „ciachów” („/”) w elemencie ścieżki, występujących po nazwie katalogu *d*, zastępowany jest przez wszystkie podkatalogi *d*: najpierw podkatalogi znajdujące się bezpośrednio pod *d*, potem te pod powyższymi i tak dalej. Na każdym etapie kolejność, w jakiej przeszukiwane są katalogi, jest *nieokreślona*.

Jeśli wyszczególni się człony nazwy pliku po „/”, uwzględnione zostaną tylko te podkatalogi, które zawierają powyższe człony. Na przykład „/a//b” rozwija się do katalogów /a/1/b, /a/2/b, /a/1/1/b itd., ale nie do /a/b/c czy /a/1.

Możliwe jest wielokrotne użycie „/” w ścieżce, jednakże „/” występujące na początku ścieżki nie jest brane pod uwagę.

7.1.7. Lista znaków specjalnych i ich znaczeń – podsumowanie

Poniższa lista podsumowuje znaczenie znaków specjalnych w plikach konfiguracyjnych.

- : znak rozdzielający w specyfikacji ścieżki; umieszczony na początku lub na końcu ścieżki zastępuje domyślne rozwinięcie ścieżki;
- ; znak rozdzielający dla systemów nieUnix-owych (działa tak jak :);
- \$ rozwijanie zmiennej;
- ~ oznacza katalog główny użytkownika;
- {...} rozwijanie nawiasów, np. $a\{1,2\}b$ zmieni się w $a1b:a2b$;
- // rozwijanie podkatalogów (może wystąpić gdziekolwiek w ścieżce, poza jej początkiem);
- % początek komentarza;
- \ znak kontynuacji (pozwala na przełamanie wiersza z wyrażeniem);
- !! przeszukiwanie *tylko* bazy danych, a *nie* dysku.

7.2. Bazy nazw plików

Dla celów przeszukiwania Kpathsea stara się zminimalizować dostęp do dysku. Niemniej jednak, w przypadku instalacji ze zbyt dużą liczbą katalogów, przeglądanie każdego możliwego katalogu w poszukiwaniu danego pliku może zabierać sporo czasu (ma to miejsce zwłaszcza, jeżeli przeszukać trzeba setki katalogów z fontami). Dlatego też Kpathsea może używać zewnętrznego pliku z „bazą danych” o nazwie `ls-R`, który to zawiera przypisania plików do katalogów. Unika się w ten sposób potrzeby wyczerpującego przeszukiwania dysku.

Drugi plik z bazą danych – `aliases` – pozwala na nadawanie dodatkowych nazw plikom zawartym w `ls-R`. Może to być pomocne do adaptacji do DOS-owej konwencji „8.3” nazewnictwa plików w plikach źródłowych.

7.2.1. Baza nazw plików

Jak to wytłumaczono powyżej, plik zawierający główną bazę nazw plików musi nosić nazwę `ls-R`. W katalogu podstawowym każdej hierarchii T_EX-owej (domyślnie `$TEXMF`), którą chcemy włączyć w mechanizm przeszukiwania, umieszczać można po jednym pliku `ls-R`; w większości przypadków istnieje tylko jedna hierarchia. Kpathsea szuka pliku `ls-R` w ścieżce `TEXMFDBS`.

Najlepszym sposobem stworzenia i utrzymywania pliku `ls-R` jest uruchomienie skryptu `mktexlsr`, będącego składnikiem dystrybucji. Jest on wywoływany przez różne skrypty typu „`mktex...`”. W zasadzie skrypt ten jedynie wykonuje polecenie

```
cd /your/texmf/root && ls -LAR ./ >ls-R
```

zakładając, że polecenie `ls` danego systemu utworzy właściwy format strumienia wyjściowego (GNU `ls` działa prawidłowo). Aby mieć pewność, że baza danych jest zawsze aktualna, wygodnie jest przebudowywać ją regularnie za pomocą demona `cron`.

Jeśli pliku nie ma w bazie danych, Kpathsea domyślnie przechodzi do przeszukiwania dysku. Jeżeli jednak dany element ścieżki zaczyna się od „`!`”, w poszukiwaniu tego elementu sprawdzona zostanie *jedynie* baza danych, a nigdy dysk.

7.2.2. `kpsewhich` – program do przeszukiwania ścieżek

Przeszukiwanie ścieżek przez program `kpsewhich` jest niezależne od jakiegokolwiek aplikacji. Może on być przydatny jako rodzaj programu `find`, za pomocą którego lokalizować można pliki w hierarchiach T_EX-owych (jest on używany intensywnie w skryptach „`mktex...`” tej dystrybucji).

```
>> kpsewhich opcje... nazwa-pliku...
```

Parametry wyszczególnione w „*opcje*” mogą zaczynać się zarówno od „`-`” jak i od „`--`”, i dozwolony jest każdy jednoznaczny skrót.

Kpathsea traktuje każdy argument nie będący parametrem jako nazwę pliku i zwraca pierwszą odnaniezoną nazwę. Nie ma parametru nakazującego zwracanie wszystkich nazw plików o określonej nazwie (w tym celu można wykorzystać Unix-owy program „`find`”).

Ważniejsze parametry opisane są poniżej.

`--dpi=num` Ustaw rozdzielczość na „*num*”; ma to tylko wpływ na przeszukiwanie fontów „`gf`” i „`pk`”.

Dla zgodności z `dvips` parametr „`-D`” działa identycznie. Domyślną wartością jest 600.

--format=*nazwa*

Ustawienie formatu (typu pliku) przeszukiwania na „*nazwa*”. Domyślnie format odpadywany jest z nazwy pliku. Dla formatów, które nie mają przydzielonego jednoznacznego rozszerzenia, takich jak niektóre pliki METAPOST-owe, czy pliki konfiguracyjne dvips-a, należy wyszczególnić nazwę, którą znaleźć można w pierwszej kolumnie tabeli 1. Zawiera ona spis aktualnie rozpoznawanych nazw, opis, przypisane zmienne środowiskowe⁴, oraz możliwe rozszerzenia nazw.

Tabela 1: Typy plików Kpathsea

<i>Nazwa</i>	<i>Opis</i>	<i>Zmienne</i>	<i>Rozszerzenia</i>
afm	Metryki czcionek Adobe	AFMFONTS	.afm
base	Bazy (formaty) Metafonta	MFBASES, TEXMFINI	.base
bib	Źródła bibliograficzne BibTeX-a	BIBINPUTS, TEXBIB	.bib
bst	Pliki stylu BibTeX	BSTINPUTS	.bst
	fonty bitmapowe	GLYPHFONTS, TEXFONTS	
cnf	Pliki konfiguracyjne	TEXMFCNF	.cnf
dvips config	Pliki konfiguracyjne Dvips, np. config.ps i psfonts.map	TEXCONFIG	.map
fmt	Formaty TeX-a	TEXFORMATS, TEXMFINI	.fmt, .efmt, .efm
gf	fonty GF (bitmapowe)	GFFONTS	.gf
graphic/figure	Rysunki w formacie EPS	TEXPICTS, TEXINPUTS	.eps, .epsi
ist	Pliki stylów makeindex	TEXINDEXSTYLE, INDEXSTYLE	.ist
ls-R	Bazy nazw plików	TEXMFDBS	
map	Mapy fontowe	TEXFONTMAPS	.map
mem	Bazy METAPOST-owe	MPMEMS, TEXMFINI	.mem
mf	Pliki źródłowe METAFONT-a	MFINPUTS	.mf
mfpool	pliki napisów METAFONT-a	MFPOOL, TEXMFINI	.pool
mft	Pliki stylu MFT	MFTINPUTS	.mft
	fonty inne	MISCFONTS	
mp	Pliki źródłowe METAPOST-a	MPINPUTS	.mp
mppool	pliki napisów METAPOST-a	MPPPOOL, TEXMFINI	.pool
MetaPost support	pliki pomocnicze METAPOST-a, wykorzystywane przez DMP	MPSUPPORT	
ocp	Ω – skompilowane pliki pomocnicze	OCPINPUTS	.ocp
ofm	Metryki fontów Ω	OFMFONTS, TEXFONTS	.ofm, .tfm
opl	Ω – listy właściwości	OPLFONTS, TEXFONTS	.opl
otp	Ω – pliki przekodowań	OTPIINPUTS	.otp
ovf	Fonty wirtualne Ω	OVPFONTS, TEXFONTS	.ovf
ovp	Wirtualne listy właściwości Ω	OVPFONTS, TEXFONTS	.ovp
pk	Fonty spakowane (PK)	programFONTS (gdzie program to XDVl, etc.), PKFONTS, TEXPKS, GLYPHFONTS, TEXFONTS	.pk
PostScript header	Ładowalne programy PostScript-owe	TEXPSHEADERS, PSHEADERS	.pro, .enc

⁴Definicje zmiennych środowiskowych znaleźć można w pliku texmf.cnf (str. 53).

Typy plików Kpathsea *cd*.

<i>Nazwa</i>	<i>Opis</i>	<i>Zmienne</i>	<i>Rozszerzenia</i>
<code>tex</code>	Pliki źródłowe dla \TeX -a	TEXINPUTS	<code>.tex</code> , <code>.cls</code> , <code>.sty</code> , <code>.clo</code> , <code>.def</code>
TeX system documentation	Dokumentacja systemu \TeX	TEXDOCS	
TeX system sources	Pliki źródłowe systemu \TeX	TEXSOURCES	
<code>texpool</code>	pliki napisów \TeX -a	TEXPOOL, TEXMFINI	<code>.pool</code>
<code>tfm</code>	Metryki fontów \TeX -owych	TFMFonts, TEXFonts	<code>.tfm</code>
Troff fonts	Fonty dla Troff, używane przez DMP	TRFonts	
truetype fonts	Fonty TrueType	TTFFonts	<code>.ttf</code> , <code>.ttc</code>
type1 fonts	Fonty Type 1 (PostScript-owe)	T1Fonts, T1INPUTS, TEXPSHEADERS, DVIPSHEADERS	<code>.pfa</code> , <code>.pfb</code>
type42 fonts	Fonty Type 42 (PostScript-owe)	T42Fonts	
<code>vf</code>	Fonty wirtualne	VFFFonts, TEXFonts	<code>.vf</code>
<code>web2c files</code>	Pomocnicze pliki Web2c	WEB2C	
other text files	Pliki tekstowe używane przez foo	FOOINPUTS	
other binary files	Pliki binarne używane przez foo	FOOINPUTS	

Ostatnie dwie pozycje w tabeli 1 to przypadki szczególne, gdzie ścieżki i zmienne środowiskowe zależą od nazwy programu: nazwa zmiennej budowana jest poprzez zapisanie nazwy programu wersalikami a następnie dodanie INPUTS.

Zmienne środowiskowe domyślnie ustawiane są w pliku konfiguracyjnym `texmf.cnf`. W wypadku gdy chce się unieważnić wartości zmiennych wyszczególnione w tym pliku, można narzucić ich ustawienie w środowisku, w którym uruchamiane są programy.

Zauważyć trzeba, że parametry „`--format`” oraz „`--path`” wzajemnie się wykluczają.

`--mode=napis`

Ustaw nazwę trybu na „*napis*”; dotyczy to jedynie szukania fontów „*gf*” oraz „*pk*”. Brak jest wartości domyślnej – odnaleziony zostanie dowolny wyszczególniony tryb.

`--must-exist`

Zrób wszystko co możliwe aby znaleźć pliki, włączając w to przede wszystkim przeszukanie dysku. Domyślnie, w celu zwiększeniu efektywności działania, sprawdzana jest tylko baza `ls-R`.

`--path=napis`

Szukaj w ścieżce „*napis*” (rozdzielonej, jak zwykle, dwukropkami), zamiast zgadywać ścieżkę przeszukiwania z nazwy pliku. „`./`” i wszystkie zwykłe rozszerzenia są możliwe. Parametry „`--path`” oraz „`--format`” wzajemnie się wykluczają.

`--programe=nazwa`

Ustaw nazwę programu na „*nazwa*”. Może to mieć wpływ na ścieżkę przeszukiwania poprzez „*program*” w plikach konfiguracyjnych. Ustawieniem domyślnym jest „*kpsewhich*”.

`--show-path=nazwa`

Pokazuje ścieżkę używaną do poszukiwania plików typu „*nazwa*”. Użyte może być zarówno rozszerzenie („.pk”, „.vf”, etc.), jak i nazwa pliku, tak jak w przypadku parametru „--format”.

`--debug=num`

ustawia parametry wykrywania błędów na „*num*”.

7.2.3. Przykłady użycia

Przyjrzyjmy się teraz jak działa Kpathsea.

```
>> kpsewhich article.cls  
/usr/local/texmf/tex/latex/base/article.cls
```

Szukamy pliku `article.cls`. Ponieważ rozszerzenie „.cls” jest jednoznaczne, nie musimy zaznaczać, że poszukujemy pliku typu „tex” (katalogi plików źródłowych T_EX-a). Znajdujemy go w podkatalogu `tex/latex/base` pod katalogiem nadrzędnym „TEXMF”. Podobnie wszystkie poniższe pliki odnajdowane są bezproblemowo dzięki swoim jednoznacznym rozszerzeniom.

```
>> kpsewhich array.sty  
/usr/local/texmf/tex/latex/tools/array.sty  
>> kpsewhich latin1.def  
/usr/local/texmf/tex/latex/base/latin1.def  
>> kpsewhich size10.clo  
/usr/local/texmf/tex/latex/base/size10.clo  
>> kpsewhich small2e.tex  
/usr/local/texmf/tex/latex/base/small2e.tex  
>> kpsewhich tugboat.bib  
/usr/local/texmf/bibtex/bib/beebe/tugboat.bib
```

Ostatni plik to BibT_EX-owa baza bibliograficzna dla artykułów *TUGBoat*.

```
>> kpsewhich cmr10.pk
```

Pliki czcionek bitmapowych typu .pk używane są przez sterowniki takie jak dvips oraz xdvi. W tym wypadku nie zostaną zwrócone żadne rezultaty przeszukiwania, ponieważ w systemie brak gotowych wygenerowanych czcionek „.pk” Computer Modern (wynika to z faktu używania na CD-ROM-ie fontów PostScript-owych Type1).

```
>> kpsewhich ecrm1000.pk  
/usr/local/texmf/fonts/pk/ljfour/jknappen/ec/ecrm1000.600pk
```

Dla rozszerzonych plików Computer Modern musieliśmy wygenerować pliki „.pk”, a ponieważ domyślnym METAFONT-owym trybem naszej instalacji jest `ljfour` z podstawową rozdzielczością 600dpi, zwracany jest taki właśnie rezultat.

```
>> kpsewhich -dpi=300 ecrm1000.pk
```

W tym wypadku, kiedy zaznaczamy, że interesujemy się rozdzielczością 300dpi (`-dpi=300`) widzimy, że taka czcionka nie jest dostępna w naszej instalacji. Program taki jak `dvips` czy `xdvi` zatrzymałby się aby utworzyć pliki `.pk` w wymaganej rozdzielczości (używając skryptu `mktexpk`).

Zwróćmy teraz naszą uwagę na pliki nagłówkowe i konfiguracyjne programu `dvips`. Najpierw szukamy pliku PostScript-owego prologu `tex.pro`, wykorzystywanego dla potrzeb \TeX -a. Drugi przykład pokazuje poszukiwanie pliku konfiguracyjnego `config.ps`, zaś trzeci – szukanie pliku mapy czcionek PostScriptowych `psfonts.map`. Jako, że rozszerzenie „.ps” nie jest jednoznaczne, musimy zaznaczyć wyraźnie jaki typ jest wymagany dla pliku `config.ps` („`dvips config`”).

```
>> kpsewhich tex.pro  
/usr/local/texmf/dvips/base/tex.pro  
>> kpsewhich --format="dvips config" config.ps  
/usr/local/texmf/config/config.ps  
>> kpsewhich psfonts.map  
/usr/local/texmf/dvips/base/psfonts.map
```

Teraz przyjrzyjmy się bliżej plikom pomocniczym URW Times PostScript. Ich nazwą w schemacie nazewnictwa fontów (zaproponowanym przez K. Berry’ego) jest „utm”. Pierwszy plik, którego szukamy to plik konfiguracyjny, zawierający nazwę pliku z przemapowaniem fontów:

```
>> kpsewhich --format="dvips config" config.utm  
/usr/local/texmf/dvips/psnfss/config.utm
```

W pliku tym znajduje się wiersz

```
p +utm.map
```

wskazujący na plik `utm.map`, który chcemy zlokalizować w następnej kolejności.

```
>> kpsewhich --format="dvips config" utm.map  
/usr/local/texmf/dvips/psnfss/utm.map
```

Ten plik z przemapowaniem definiuje nazwy czcionek PostScriptowych Type1 w zestawie URW, zaś jego zawartość wygląda następująco (pokazane są tylko fragmenty linii):

```
utmb8r NimbusRomNo9L-Medi ... <utmb8a.pfb  
utmbi8r NimbusRomNo9L-MediItal... <utmbi8a.pfb  
utmr8r NimbusRomNo9L-Regu ... <utmr8a.pfb  
utmri8r NimbusRomNo9L-ReguItal... <utmri8a.pfb  
utmbo8r NimbusRomNo9L-Medi ... <utmb8a.pfb  
utmro8r NimbusRomNo9L-Regu ... <utmr8a.pfb
```

Weźmy na przykład font Times Regular `utmr8a.pfb` i znajźmy jego miejsce w drzewie katalogów `texmf`, używając przeszukiwania plików z fontami Type1:

```
>> kpsewhich utmr8a.pfb  
/usr/local/texmf/fonts/type1/urw/utm/utmr8a.pfb
```

Powyższe przykłady pokazują, jak łatwo można znajdować lokalizację danego pliku. Ważne jest to zwłaszcza, gdy istnieje podejrzenie, że gdzieś zawieruszyła się zła wersja jakiegoś pliku; `kpsewhich` pokaże tylko pierwszy napotkany plik.

7.2.4. Diagnostyka błędów

Czasami niezbędne są informacje o tym, jak program radzi sobie z odniesieniami do plików. Aby dało się to wykonywać w wygodny sposób, Kpathsea oferuje różne poziomy diagnostyki błędów:

- 1 wywołania stat (testy pliku). Podczas uruchamiania z uaktualnioną bazą danych `ls-R` nie powinno to przeważnie dawać żadnego wyniku.
- 2 Zapis odwołań do tablic mieszających (*hash tables*), takich jak baza `ls-R`, pliki przemapowań, pliki konfiguracyjne.
- 4 Operacje otwarcia i zamknięcia pliku.
- 8 Ogólne informacje o ścieżkach dla typów plików szukanych przez Kpathsea; użyteczne dla znalezienia ścieżki zdefiniowanej dla danego pliku.
- 16 Lista katalogów dla każdego z elementów ścieżki (odnosi się tylko do poszukiwań na dysku).
- 32 Poszukiwania plików.

Wartość `-1` ustawia wszystkie powyższe opcje: w praktyce, potrzebując wykryć przyczyny błędów, prawdopodobnie zawsze używać będziesz tych poziomów.

Podobnie, w przypadku programu `dvips`, ustawiając kombinację przełączników wykrywania błędów, można dokładnie śledzić skąd pochodzą pliki. W wypadku gdy plik nie zostanie odnaleziony, widać, w których katalogach program szukał danego pliku, dzięki czemu można się zorientować w czym problem.

Ogólnie mówiąc, ponieważ programy odwołują się wewnętrznie do biblioteki Kpathsea, opcje wykrywania błędów wybrać można przy użyciu zmiennej środowiskowej `KPATHSEA_DEBUG`, ustawiając ją na opisaną powyżej wartość (kombinację wartości).

Uwaga dla użytkowników Windows: w systemie tym niełatwo jest przekierować komunikaty programu do pliku. Do celów diagnostycznych można w tym celu ustawić chwilowo zmienne (w oknie DOS):

```
SET KPATHSEA_DEBUG_OUTPUT=err.log
SET KPATHSEA_DEBUG=-1
```

Rozważmy na przykład mały \LaTeX -owy plik źródłowy `hello-world.tex`, który zawiera co następuje:

```
\documentclass{article}
\begin{document}
Hello World!
\end{document}
```

Ten mały plik używa jedynie fontu `cmr10`. Przyjrzyjmy się jak `dvips` przygotowuje plik PostScript-owy (chcemy użyć wersji Type1 fontu Computer Modern, stąd opcja `-Pcms`).

```
>> dvips -d4100 hello-world -Pcms -o
```

Mamy tu do czynienia jednocześnie z czwartą klasą wykrywania błędów programu `dvips` (ścieżki fontowe) oraz z rozwijaniem elementu ścieżki przez Kpathsea (patrz *Dvips Reference Manual*, texmf/doc/html/dvips/dvips_toc.html). Komunikaty z uruchomienia programu (lekko zmodyfikowane)

```

debug:start search(file=texmf.cnf, must_exist=1, find_all=1,
  path=./usr/local/bin/texlive:/usr/local/bin:
    /usr/local/bin/texmf/web2c:/usr/local:
    /usr/local/texmf/web2c/././teTeX/TeX/texmf/web2c:).
kdebug:start search(file=ls-R, must_exist=1, find_all=1,
  path=~/.tex:/usr/local/texmf).
kdebug:search(ls-R) => /usr/local/texmf/ls-R
kdebug:start search(file=aliases, must_exist=1, find_all=1,
  path=~/.tex:/usr/local/texmf).
kdebug:search(aliases) => /usr/local/texmf/aliases
kdebug:start search(file=config.ps, must_exist=0, find_all=0,
  path=~/.tex:/usr/local/texmf/dvips/).
kdebug:search(config.ps) => /usr/local/texmf/dvips/config/config.ps
kdebug:start search(file=/root/.dvipsrc, must_exist=0, find_all=0,
  path=~/.tex:/usr/local/texmf/dvips/).
search(file=/home/goossens/.dvipsrc, must_exist=1, find_all=0,
  path=~/.tex/dvips/./usr/local/texmf/dvips/).
kdebug:search($HOME/.dvipsrc) =>
kdebug:start search(file=config.cms, must_exist=0, find_all=0,
  path=~/.tex/dvips/./usr/local/texmf/dvips/).
kdebug:search(config.cms)
=>/usr/local/texmf/dvips/cms/config.cms

```

Rysunek 10: Szukanie pliku konfiguracyjnego

```

kdebug:start search(file=texc.pro, must\_exist=0, find\_all=0,
  path=~/.tex/dvips/./usr/local/texmf/dvips/./:
    ~/.tex/fonts/type1/./usr/local/texmf/fonts/type1/).
kdebug:search(texc.pro) => /usr/local/texmf/dvips/base/texc.pro

```

Rysunek 11: Szukanie pliku prologu

```

kdebug:start search(file=cmr10.tfm, must\_exist=1, find\_all=0,
  path=~/.tex/fonts/tfm/./usr/local/texmf/fonts/tfm/./:
    /var/tex/fonts/tfm/).
kdebug:search(cmr10.tfm) => /usr/local/texmf/fonts/tfm/public/cm/cmr10.tfm
kdebug:start search(file=texps.pro, must\_exist=0, find\_all=0,
  ...
<texps.pro>
kdebug:start search(file=cmr10.pfb, must\_exist=0, find\_all=0,
  path=~/.tex/dvips/./usr/local/texmf/dvips/./:
    ~/.tex/fonts/type1/./usr/local/texmf/fonts/type1/).
kdebug:search(cmr10.pfb) => /usr/local/texmf/fonts/type1/public/cm/cmr10.pfb
<cmr10.pfb>[1]

```

Rysunek 12: Szukanie pliku fontu

znajdują się na rys. 10. Program dvips zaczyna pracę od zlokalizowania potrzebnych mu plików. Najpierw znajduje plik `texmf.cnf`, który zawiera ścieżki przeszukiwania dla innych plików. Potem znajduje bazę danych `ls-R` (dla optymalizacji szukania plików), następnie plik `aliases`, który umożliwia deklarowanie różnych nazw (np. krótkie DOS-owe „8.3” i bardziej naturalne dłuższe wersje) dla tych samych plików. Następnie dvips znajduje podstawowy plik konfiguracyjny `config.ps`, zanim poszuka pliku z ustawieniami użytkownika `.dvipsrc` (który w tym wypadku *nie* zostaje odnaleziony). W końcu dvips lokalizuje plik konfiguracyjny `config.cms` dla fontów PostScript-owych Computer Modern (jest to inicjowane przez dodanie parametru `-Pcms` przy uruchamianiu programu). Plik ten zawiera listę plików z „mapami”, które definiują relacje pomiędzy \TeX -owymi, PostScript-owymi i systemowymi nazwami fontów.

```
>> more /usr/local/texmf/dvips/cms/config.cms
p +ams.map
p +cms.map
p +cmbkm.map
p +amsbkm.map
```

W ten sposób dvips wyszukuje wszystkie te pliki oraz główny plik z przemapowaniem `psfonts.map`, który ładowany jest zawsze (zawiera on deklaracje często używanych fontów PostScript-owych; więcej szczegółów odnośnie PostScript-owych plików przemapowań fontów można znaleźć w ostatniej części rozdziału 7.2.3).

W tym miejscu dvips zgłasza się użytkownikowi:

```
This is dvips 5.86 Copyright 1999 Radical Eye Software (www.radicaleye.com)
```

...potem szuka pliku prologu `texc.pro`,

```
kdebug:start search(file=texc.pro, must_exist=0, find_all=0,
  path=.:~/tex/dvips/./:/usr/local/texmf/dvips/./:
  ~/tex/fonts/type1/./:/usr/local/texmf/fonts/type1/./).
kdebug:search(texc.pro) => /usr/local/texmf/dvips/base/texc.pro
```

Po znalezieniu szukanego pliku, dvips podaje datę i czas oraz informuje o generowaniu pliku `hello-world.ps`. Ponieważ potrzebuje pliku z fontem `cmr10`, a jest on zadeklarowany jako dostępny, wyświetla komunikat:

```
TeX output 1998.02.26:1204' -> hello-world.ps
Defining font () cmr10 at 10.0pt
Font cmr10 <CMR10> is resident.
```

Teraz trwa poszukiwanie pliku `cmr10.tfm`, który zostaje znaleziony, a następnie dvips powołuje się na kilka innych plików startowych (nie pokazanych). W końcu przykładowy font `Type1 cmr10.pfb` zostaje zlokalizowany i dołączony do pliku wynikowego (patrz ostatnia linia).

```
kdebug:start search(file=cmr10.tfm, must_exist=1, find_all=0,
  path=.:~/tex/fonts/tfm/./:/usr/local/texmf/fonts/tfm/./:
  /var/tex/fonts/tfm/./).
kdebug:search(cmr10.tfm) => /usr/local/texmf/fonts/tfm/public/cm/cmr10.tfm
kdebug:start search(file=texps.pro, must_exist=0, find_all=0,
  ...
```

```
<texps.pro>
kdebug:start search(file=cmr10.pfb, must_exist=0, find_all=0,
  path=.:~/tex/dvips//:!!/usr/local/texmf/dvips//:
    ~/tex/fonts/type1//:!!/usr/local/texmf/fonts/type1//).
kdebug:search(cmr10.pfb) => /usr/local/texmf/fonts/type1/public/cm/cmr10.pfb
<cmr10.pfb>[1]
```

7.3. Parametry kontrolujące działanie programów

Inną użyteczną cechą Web2c jest możliwość ustalania wielu parametrów określających wielkość pamięci za pomocą pliku `texmf.cnf`. Załącznik 11 zaczynający się na stronie 53 zawiera listing pliku `texmf.cnf`. Ustawienia wszystkich parametrów znajdują się w części trzeciej pliku. Najważniejszymi zmiennymi są:

main_memory Całkowita wielkość pamięci dostępnej dla \TeX -a, METAFONT-a i METAPOST-a.

Dla każdego nowego ustawienia tej zmiennej należy wykonać nowy format. Przykładowo, możesz wygenerować „ogromną” wersję formatu \TeX i nazwać taki plik `hugetex.fmt`. Dzięki standardowemu sposobowi nazywania programów przez Kpathsea, określona wartość zmiennej `main_memory` będzie przeczytana z pliku `texmf.cnf` (por. wartość standardową oraz wartość „ogromną”, wykorzystywaną przez program `hugetex`, itd.).

extra_mem_bot Dodatkowa wielkość pamięci przeznaczona na „duże” struktury danych \TeX -a, takie jak: pudełka, kleje itd.; przydatna, zwłaszcza w przypadku korzystania z pakietu $\P\TeX$.

font_mem_size Wielkość pamięci przeznaczona przez \TeX -a na informacje o fontach. Jest to mniej więcej ogólna wielkość wczytywanych przez \TeX -a plików TFM.

hash_extra Dodatkowa wielkość pamięci przeznaczona na tablicę zawierającą nazwy instrukcji.

Tablica główna może zmieścić w przybliżeniu 10000 nazw; wielkość ta może okazać się zbyt mała, np. w wypadku obszernej książki zawierającej liczne odwołania. Odpowiednie wiersze pliku `texmf.cnf` pokazują, że uruchomienie programów `hugetex` i `pdflatex` wymaga dodatkowej pamięci na 15000 nazw instrukcji.

Oczywiście, powyższa możliwość nie zastąpi prawdziwej, dynamicznej alokacji pamięci. Jest to jednak niezwykle trudne do zaimplementowania w obecnej wersji \TeX -a i dlatego powyższe parametry stanowią praktyczny kompromis, pozwalając na pewną elastyczność.

8. Podziękowania

Niniejsza edycja **\TeX Live** została opracowana przez Sebastiana Rahtza przy głównym współudziale: Fabrice Popineau, który niezmordowanie przygotowywał pakiety dla systemów win32 (szczególnie `setup!`) i wniósł wielki wkład w formie pomysłów, porad i kodu oraz Staszka Wawrykiewicza, który sprawdzał, modyfikował i testował dystrybucję, a także koordynował przygotowanie polskich pakietów. Kaja Christiansen odegrała wielką rolę nieustannie rekompilując programy na różnych platformach uniksowych. Robin Laakso koordynował produkcję dla TUG. Vladimir Volovich wykonał wielką pracę czyszczenia plików źródłowych i wprowadzania wielu ulepszeń. Gerben Wierda wykonał wszystkie prace związane z systemem MacOSX.

Szczególne słowa podziękowania za dawniejszą i aktualną pomoc należą się:

- Niemieckiej Grupie \TeX -owej (DANTE e.V.), która udostępniła komputer, na którym przygotowywano niniejszy CD-ROM; a w szczególności Rainerowi Schöpfowi i Reinhardowi Zierke, którzy opiekowali się tym komputerem;
- Firmie Perforce, która udostępniła bezpłatną kopię świetnego oprogramowania do zarządzania zmianami w przygotowywanej dystrybucji;
- Karlowi Berry’emu, za rady, zachętę i (oczywiście) za opracowanie oryginalnej wersji dystrybucji Web2c;
- Mimi Burbank, która umożliwiła dostęp do wielu różnych komputerów Florida State University School of Computational Science and Information Technology, na których można było kompilować \TeX -a. Ponadto Mimi, kiedykolwiek poproszona, działała jako najlepszy królik doświadczalny;
- Kaji Christiansen, za wiele cennych uwag, wskazówek i poprawek, kompilację programów oraz pomoc przy tworzeniu dokumentacji;
- Thomasowi Esserowi, za wspaniały pakiet $\text{te}\text{\TeX}$, bez którego ten CD-ROM z całą pewnością by nie powstał, a także którego stała pomoc umożliwiła uczynienie z tego CD-ROM-u coraz lepszego produktu;
- Eitanowi Gurari, za program TeX4ht wykorzystany do tworzenia tej dokumentacji w wersji HTML, a w szczególności za niezmordowaną pracę nad jego ulepszaniem i błyskawicznie wykonywane poprawki;
- Arthurowi Ogawie and Pati Monohon, którzy koordynowali wydania dla TUG;
- Petrowi Olšákowi, który koordynował i starannie sprawdzał przygotowanie czeskich i słowackich pakietów;
- Olafowi Weberowi, za cierpliwe pielęgnowanie Web2c;
- Grahamowi Williamsowi, za pracę nad katalogiem pakietów.

Gerhard Wilhelms, Volker Schaa, Fabrice Popineau, Janka Chlebková, Staszek Wawrykiewicz, Erik Frambach i Ulrik Vieth tłumaczyli w różnych momentach dokumentację na ich rodzinne języki. Ponadto sprawdzili oryginalną dokumentację oraz zgłosili wiele cennych uwag i poprawek.

9. Historia

Dystrybucja **\TeX Live** jest wspólnym przedsięwzięciem wielu grup Użytkowników Systemu \TeX z: Niemiec, Holandii, Wielkiej Brytanii, Francji, Czech, Słowacji, Polski, Indii i Rosji, oraz międzynarodowej TUG (*\TeX Users Group*). Dyskusje nad projektem rozpoczęły się pod koniec 1993 roku, kiedy Holenderska Grupa Użytkowników \TeX -a rozpoczęła prace nad swoim 4All \TeX CD-ROM dla użytkowników MS-DOS. W tym też czasie pojawiły się nadzieje na opracowanie jednego CD-ROM dla wszystkich systemów. Projekt taki był wprawdzie zbyt ambitny, ale zrodził nie tylko bardzo popularny i uwieńczony dużym powodzeniem projekt 4All \TeX CD-ROM, a także spowodował powstanie Grupy Roboczej TUG ds. Standardu Katalogów \TeX -owych (*\TeX Directory Structure*), określającego w jaki

sposób tworzyć zgodne i łatwe do zarządzania zestawy pakietów \TeX -owych. Końcowy raport TDS został opublikowany w grudniowym numerze *TUGboat*-a, i jasnym się stało, że jednym z oczekiwanych rezultatów wprowadzenia tego standardu mogłaby być modelowa struktura na CD-ROM-ie. Niniejszy CD-ROM jest bezpośrednim rezultatem rozważań i zaleceń Grupy Roboczej ds. TDS. Jasne także było, że sukces 4All \TeX CD-ROM pokazał, że użytkownicy Unixa także wiele by zyskali, mogąc korzystać z podobnie łatwego w instalacji/pielęgnacji i użytkowaniu systemu. Było to jednym z celów projektu **\TeX Live**.

Projekt przygotowania nowego, opartego na standardzie TDS CD-ROM, zorientowanego na systemy Unixowe, rozpoczął się jesienią 1995 roku. Szybko zdecydowaliśmy się na wykorzystanie \TeX Thomasa Essera, ponieważ oferował on wsparcie dla wielu platform i został zaprojektowany z myślą o przenośności pomiędzy różnymi systemami plików. Thomas zgodził się pomóc i prace rozpoczęły się na serio na początku 1996 roku. Pierwsze wydanie ukazało się w maju 1996 roku. Na początku 1997 roku Karl Berry udostępnił nową, poważnie zmienioną wersję swojego pakietu Web2c, zawierającą prawie wszystkie funkcje wprowadzone do \TeX -a przez Thomasa Essera. W związku z tym zdecydowaliśmy się oprzeć drugie wydanie CD-ROM na standardowej bibliotece Web2c, z dodaniem skryptu `texconfig` z pakietu \TeX . Trzecie wydanie CD-ROM było oparte na Web2c wersji 7.2, przygotowanej przez Olafa Webera. W tym samym czasie została przygotowana nowa wersja \TeX -a i **\TeX Live** udostępniał prawie wszystkie jego nowe funkcje. Czwarta edycja była przygotowana podobnie, wykorzystując nową wersję \TeX -a i nową wersję Web2c (7.3). Wtedy to też zainaugurowano kompletną dystrybucję dla Windows.

Edycja piąta (marzec 2000) zawierała wiele poprawek i uzupełnień; zaktualizowano setki pakietów. Szczegóły zawartości pakietów zostały zapisane w plikach XML. Główną zmianą w \TeX Live5 było usunięcie programów, które nie miały statusu *public domain*. Zawartość całej płytki powinna odpowiadać ustaleniom Debian Free Software Guidelines (<http://www.debian.org/intro/free>). Dołożyliśmy wszelkich starań aby sprawdzić warunki licencyjne pakietów, ale bardzo prosimy o zgłaszanie zauważonych pomyłek.

Szósta edycja (lipiec 2001) zawierała aktualizacje całego materiału. Główną zmianą było wprowadzenie nowej koncepcji programów instalacyjnych: użytkownik ma obecnie możliwość dokładniejszego wyboru potrzebnych zestawów i pakietów. Zestawy dotyczące obsługi poszczególnych języków zostały całkowicie zreorganizowane, dzięki czemu wybór jednego z nich nie tylko instaluje potrzebne makra i fonty, ale także przygotowuje odpowiedni plik `language.dat` i generuje podstawowe formaty (`latex`, `pdflatex`) ze wzorcami przenoszenia wyrazów dla danego języka.

\TeX Live7 zawiera oprogramowanie także dla MacOSX i, jak zwykle, aktualizację wszelkich programów i pakietów. Ważnym zadaniem, które wykonano, było ujednolicenie plików źródłowych programów z dystrybucją \TeX . W programach instalacyjnych wprowadzono możliwość wyboru bardziej ogólnych, predefiniowanych zestawów pakietów (m.in. dla użytkowników francuskojęzycznych oraz polskich). Nowością jest także wprowadzenie procedury aktualizacji map fontowych dla Dvips i Pdftex podczas instalacji oraz doinstalowywania pakietów fontowych.

10. Przyszłe wersje

Niniejszy CD-ROM nie jest doskonały! Planujemy wydawanie go raz na rok, i chcielibyśmy żeby zawierał jeszcze więcej pomocnej dokumentacji, więcej programów użytkowych, lepsze programy instalacyjne oraz (oczywiście) stale ulepszany i poprawiany katalog makr i fontów. Zadanie to jest wykonywane przez ciężko pracujących ochotników, poświęcających na to wiele swojego wolnego czasu. Wiele jeszcze pozostało do zrobienia. Jeżeli możesz pomóc, nie zastanawiaj się i przyłącz się do nas.

Poprawki, wskazówki i uzupełnienia dla przyszłych wersji powinny być przesyłane na adres:

Sebastian Rahtz
7 Stratfield Road
Oxford OX2 7BG
United Kingdom
rahtz@tug.org

Uaktualnienia, ogłoszenia i wskazówki będą dostępne w archiwum CTAN w katalogu `info/texlive`. Informacja i szczegóły dotyczące zakupu znajdują się także na stronie <http://www.tug.org/texlive.html>.

11. Plik konfiguracyjny `texmf.cnf`

```
1 % TeX Live texmf.cnf
2 % What follows is a super-summary of what this .cnf file can
3 % contain. Please read the Kpathsea manual for more information.
4 %
5 % texmf.cnf is generated from texmf.in, by replacing @var@ with the
6 % value of the Make variable 'var', via a sed file texmf.sed, generated
7 % (once) by kpathsea/Makefile (itself generated from kpathsea/Makefile.in
8 % by configure).
9 %
10 % Any identifier (sticking to A-Za-z_ for names is safest) can be assigned.
11 % The '=' (and surrounding spaces) is optional.
12 % No % or @ in texmf.in, for the sake of autogeneration.
13 % (However, %'s and @'s can be edited into texmf.cnf or put in envvar values.)
14 % $foo (or ${foo}) in a value expands to the envvar or cnf value of foo.
15 %
16 % Earlier entries (in the same or another file) override later ones, and
17 % an environment variable foo overrides any texmf.cnf definition of foo.
18 %
19 % All definitions are read before anything is expanded, so you can use
20 % variables before they are defined.
21 %
22 % If a variable assignment is qualified with '.PROGRAM', it is ignored
23 % unless the current executable (last filename component of argv[0]) is
24 % named PROGRAM. This foo.PROGRAM construct is not recognized on the
25 % right-hand side. For environment variables, use FOO_PROGRAM.
26 %
27 % Which file formats use which paths for searches is described in the
28 % various programs' and the kpathsea documentation.
29 %
30 % // means to search subdirectories (recursively).
31 % A leading !! means to look only in the ls-R db, never on the disk.
32 % A leading/trailing/doubled ; in the paths will be expanded into the
33 % compile-time default. Probably not what you want.
34 %
35 % You can use brace notation, for example: /usr/local/{mytex:othertex}
36 % expands to /usr/local/mytex:/usr/local/othertex. Instead of the path
37 % separator you can use a comma: /usr/local/{mytex,othertex} also expands
38 % to /usr/local/mytex:/usr/local/othertex. However, the use of the comma
39 % instead of the path separator is deprecated.
40 %
41 % The text above assumes that path separator is a colon (:). Non-UNIX
42 % systems use different path separators, like the semicolon (;).
43
44 % Part 1: Search paths and directories.
45
46 % You can set an environment variable to override TEXMF if you're testing
47 % a new TeX tree, without changing anything else.
48 %
```

```

49 % You may wish to use one of the $SELFAUTO... variables here so TeX will
50 % find where to look dynamically. See the manual and the definition
51 % below of TEXMFCNF.
52
53 % The main tree, which must be mentioned in $TEXMF, below:
54 TEXMFMAIN =                $SELFAUTOPARENT/texmf
55 % A place for local additions to a "standard" texmf tree.
56 TEXMFLOCAL =               $SELFAUTOPARENT/texmf-local
57
58 % User texmf trees can be catered for like this...
59 HOMETEXMF=$HOME/texmf
60
61 % A place where texconfig stores modifications (instead of the TEXMFMAIN
62 % tree). texconfig relies on the name, so don't change it.
63 VARTEXMF =                  $SELFAUTOPARENT/texmf-var
64
65 % Now, list all the texmf trees. If you have multiple trees,
66 % use shell brace notation, like this:
67 %   TEXMF =                  {$HOMETEXMF,!!$VARTEXMF,!!$TEXMFLOCAL,!!$TEXMFMAIN}
68 % The braces are necessary.
69 %
70 % A place where to store other TeX support files. It can be a remote
71 % texmf tree, or a tree to store non-free stuff, or ...
72 %   TEXMFEXTRA=$SELFAUTOPARENT/texmf-extra
73 % If you set this, add $TEXMFEXTRA in the list below
74 %
75 TEXMF =                      {$HOMETEXMF,!!$VARTEXMF,$TEXMFLOCAL,!!$TEXMFMAIN}
76
77 % The system trees. These are the trees that are shared by all the users.
78 SYSTEXMF =                    $TEXMF
79
80 % The temporary area
81 TEMP =                        /var/tmp
82
83 % Where generated fonts may be written. This tree is used when the sources
84 % were found in a system tree and either that tree wasn't writable, or the
85 % varfonts feature was enabled in MT_FEATURES in mktex.cnf.
86 VARTEXFONTS =                 $VARTEXMF/fonts
87
88 % Where to look for ls-R files. There need not be an ls-R in the
89 % directories in this path, but if there is one, Kpathsea will use it.
90 TEXMFDBS =                     $TEXMF
91
92 % It may be convenient to define TEXMF like this:
93 %   TEXMF =                   {$HOMETEXMF,!!$TEXMFLOCAL,!!$TEXMFMAIN,$HOME}
94 % which allows users to set up entire texmf trees, and tells TeX to
95 % look in places like ~/tex and ~/bibtex. If you do this, define TEXMFDBS
96 % like this:
97 %   TEXMFDBS =                 $HOMETEXMF;$TEXMFLOCAL;$TEXMFMAIN;$VARTEXFONTS
98 % or mktexlsr will generate an ls-R file for $HOME when called, which is
99 % rarely desirable. If you do this you'll want to define SYSTEXMF like
100 % this:
101 %   SYSTEXMF =                 $TEXMFLOCAL;$TEXMFMAIN
102 % so that fonts from a user's tree won't escape into the global trees.
103 %
104 % On some systems, there will be a system tree which contains all the font
105 % files that may be created as well as the formats. For example
106 %   VARTEXMF =                  /var/lib/texmf
107 % is used on many Linux systems. In this case, set VARTEXFONTS like this
108 %   VARTEXFONTS =               $VARTEXMF/fonts
109 % and do not mention it in TEXMFDBS (but _do_ mention VARTEXMF).
110
111
112 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
113 % Usually you will not need to edit any of the other variables in part 1. %
114 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
115
116 % WEB2C is for Web2C specific files. The current directory may not be

```

```

117 % a good place to look for them.
118 WEB2C = $TEXMF/web2c
119
120 % TEXINPUTS is for TeX input files -- i.e., anything to be found by \input
121 % or \openin, including .sty, .eps, etc.
122
123 % LaTeX-specific macros are stored in latex.
124 TEXINPUTS.latex = .;$TEXMF/tex/{latex,generic,}//
125 TEXINPUTS.hugelatex = .;$TEXMF/tex/{latex,generic,}//
126
127 % Fontinst needs to read afm files.
128 TEXINPUTS.fontinst = .;$TEXMF/{tex/{fontinst,},fonts/afm}//
129
130 % Plain TeX. Have the command tex check all directories as a last
131 % resort, we may have plain-compatible stuff anywhere.
132 TEXINPUTS.tex = .;$TEXMF/tex/{plain,generic,}//
133 % other plain-based formats
134 TEXINPUTS.amstex = .;$TEXMF/tex/{amstex,plain,generic,}//
135 TEXINPUTS.ftex = .;$TEXMF/tex/{formate,plain,generic,}//
136 TEXINPUTS.texinfo = .;$TEXMF/tex/{texinfo,plain,generic,}//
137 TEXINPUTS.eplain = .;$TEXMF/tex/{eplain,plain,generic,}//
138
139 % e-TeX.
140 TEXINPUTS.elatex = .;$TEXMF/{etex,tex}/{latex,generic,}//
141 TEXINPUTS.etex = .;$TEXMF/{etex,tex}/{plain,generic,}//
142
143 % PDFTeX. This form of the input paths is borrowed from TeTeX. A certain
144 % variant of TDS is assumed here, unaffected by the build variables.
145 TEXINPUTS.pdfetexinfo = .;$TEXMF/{pdfetex,tex}/{texinfo,plain,generic,}//
146 TEXINPUTS.pdfplatex = .;$TEXMF/{pdfetex,tex}/{latex,generic,}//
147 TEXINPUTS.pdfetex = .;$TEXMF/{pdfetex,tex}/{plain,generic,}//
148 TEXINPUTS.pdfelatex = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{latex,generic,}//
149 TEXINPUTS.pdfetex = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{plain,generic,}//
150
151 % Omega.
152 TEXINPUTS.lambda = .;$TEXMF/{omega,tex}/{lambda,latex,generic,}//
153 TEXINPUTS.omega = .;$TEXMF/{omega,tex}/{plain,generic,}//
154
155 % Context macros by Hans Hagen:
156 TEXINPUTS.context = .;$TEXMF/{pdfetex,pdftex,etex,tex}/{context,plain,generic,}//
157
158 % cstex, from Petr Olsak
159 TEXINPUTS.cslatex = .;$TEXMF/tex/{cslatex,csplain,latex,generic,}//
160 TEXINPUTS.csplain = .;$TEXMF/tex/{csplain,plain,generic,}//
161 TEXINPUTS.pdfcslatex = .;$TEXMF/{pdfetex,tex}/{cslatex,csplain,latex,generic,}//
162 TEXINPUTS.pdfcsplain = .;$TEXMF/{pdfetex,cstex,tex}/{csplain,plain,generic,}//
163
164 % Polish
165 TEXINPUTS.platex = .;$TEXMF/{tex}/{platex,latex,generic,}//
166 TEXINPUTS.pdfplatex = .;$TEXMF/{pdfetex,tex}/{platex,latex,generic,}//
167 TEXINPUTS.pdfmex = .;$TEXMF/{pdfetex,tex}/{mex,plain,generic,}//
168 TEXINPUTS.pdfmex = .;$TEXMF/{pdfetex,pdftex,tex}/{mex,plain,generic,}//
169 TEXINPUTS.mex = .;$TEXMF/tex/{mex,plain,generic,}//
170
171 % french
172 TEXINPUTS.frlatex = .;$TEXMF/{mltex,tex}/{french,latex,generic,}//
173 TEXINPUTS.frtex = .;$TEXMF/{mltex,tex}/{french,plain,generic,}//
174 TEXINPUTS.frpdlatex = .;$TEXMF/{mltex,pdftex,tex}/{french,latex,generic,}//
175 TEXINPUTS.frpdlatex = .;$TEXMF/{mltex,pdftex,tex}/{french,plain,generic,}//
176
177 % MLTeX
178 TEXINPUTS.mltex = .;$TEXMF/{mltex,tex}/{plain,generic,}//
179 TEXINPUTS.mllatex = .;$TEXMF/{mltex,tex}/{latex,generic,}//
180
181 % odd formats needing their own paths
182 TEXINPUTS.lollipop = .;$TEXMF/tex/{lollipop,generic,plain,}//
183 TEXINPUTS.lamstex = .;$TEXMF/tex/{lamstex,generic,plain,}//
184

```

```

185 % David Carlisle's xmltex
186 TEXINPUTS.xmltex = .;$TEXMF/tex/{xmltex,latex,generic,}//
187 TEXINPUTS.pdfxmltex = .;$TEXMF/{pdftex,tex}/{xmltex,latex,generic,}//
188
189 % Sebastian Rahtz' jadetex for DSSSL
190 TEXINPUTS.pdfjadetex = .;$TEXMF/{pdftex,tex}/{jadetex,generic,plain,}//
191 TEXINPUTS.jadetex = .;$TEXMF/tex/{jadetex,generic,plain,}//
192
193 % Earlier entries override later ones, so put this last.
194 TEXINPUTS = .;$TEXMF/tex/{generic,}//
195
196 % Metafont, MetaPost inputs.
197 MFINPUTS = .;$TEXMF/metafont//;$TEXMF/fonts,$VARTEXFONTS}/source//
198 MPINPUTS = .;$TEXMF/metapost//
199
200 % mft
201 MFTINPUTS = .;$TEXMF/mft//
202
203 % Web and CWeb input paths.
204 WEBINPUTS = .;$TEXMF/web//
205 CWEBINPUTS = .;$TEXMF/cweb//
206
207 % Dump files (fmt/base/mem) for vir{tex,mf,mp} to read (see web2c/INSTALL),
208 % and string pools (.pool) for ini{tex,mf,mp}. It is silly that we have six
209 % paths and directories here (they all resolve to a single place by default),
210 % but historically ...
211 TEXFORMATS = .;$TEXMF/web2c
212 MFBASES = $TEXFORMATS
213 MPMEMS = $TEXFORMATS
214 TEXPOOL = $TEXFORMATS
215 MFPOOL = $TEXFORMATS
216 MPPPOOL = $TEXFORMATS
217
218 % Device-independent font metric files.
219 VFFONTS = .;$TEXMF/fonts/vf//
220 TFMFONTS = .;$TEXMF/fonts,$VARTEXFONTS}/tfm//
221
222 % The $MAKETEX_MODE below means the drivers will not use a cx font when
223 % the mode is ricoh. If no mode is explicitly specified, kpse_prog_init
224 % sets MAKETEX_MODE to /, so all subdirectories are searched. See the manual.
225 % The modeless part guarantees that bitmaps for PostScript fonts are found.
226 PKFONTS = .;$TEXMF/fonts,$VARTEXFONTS}/pk/{$MAKETEX_MODE,modeless}//
227
228 % Similarly for the GF format, which only remains in existence because
229 % Metafont outputs it (and MF isn't going to change).
230 GFFONTS = .;$TEXMF/fonts/gf/$MAKETEX_MODE//
231
232 % A backup for PKFONTS and GFFONTS. Not used for anything.
233 GLYPHFONTS = .;$TEXMF/fonts
234
235 % For texfonts.map and included map files used by mktexpk.
236 % See ftp://ftp.tug.org/tex/fontname.tar.gz.
237 TEXFONTMAPS = .;$TEXMF/fontname
238
239 % BibTeX bibliographies and style files.
240 BIBINPUTS = .;$TEXMF/bibtex/{bib,}//
241 BSTINPUTS = .;$TEXMF/bibtex/{bst,}//
242
243 % PostScript headers, prologues (.pro), encodings (.enc) and fonts;
244 % this is also where pdftex finds included figures files!
245
246 TEXPSHEADERS.pdflatex = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
247 TEXPSHEADERS.pdfelatex = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
248 TEXPSHEADERS.pdftexinfo = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
249 TEXPSHEADERS.pdfcslatex = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
250 TEXPSHEADERS.pdfcsplain = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
251 TEXPSHEADERS.pdfetex = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
252 TEXPSHEADERS.pdfjadetex = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//

```



```

253 TEXPSHEADERS.pdfplatex = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
254 TEXPSHEADERS.pdfxmltex = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
255 TEXPSHEADERS.pdfmex = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
256 TEXPSHEADERS.pdftex = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
257 TEXPSHEADERS.pdftexinfo = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
258 TEXPSHEADERS.cont-de = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
259 TEXPSHEADERS.cont-en = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
260 TEXPSHEADERS.cont-nl = .;$TEXMF/{tex,pdftex,dvips,fonts/{type1,pfb}}//
261 TEXPSHEADERS.context = .;$TEXMF/{etex,tex,pdftex,dvips,fonts/{type1,pfb}}//
262 TEXPSHEADERS = .;$TEXMF/{dvips,fonts/{type1,pfb},pdftex}//
263
264 % PostScript Type 1 outline fonts.
265 T1FONTS = .;$TEXMF/fonts/{type1,pfb};;$TEXMF/fonts/misc/hbf//
266
267 % PostScript AFM metric files.
268 AFMFONTS = .;$TEXMF/fonts/afm//
269
270 % TrueType outline fonts.
271 TTFONTS = .;$TEXMF/fonts/{truetype,ttf}//
272 TTF2TFMINPUTS = .;$TEXMF/ttf2pk//
273
274 % Type 42 outline fonts.
275 T42FONTS = .;$TEXMF/fonts/type42//
276
277 % A place to puth everything that doesn't fit the other font categories.
278 MISCFONTS = .;$TEXMF/fonts/misc//
279
280 % Dvips' config.* files (this name should not start with 'TEX!').
281 TEXCONFIG = .;$TEXMF/dvips//
282
283 % Makeindex style (.ist) files.
284 INDEXSTYLE = .;$TEXMF/makeindex;;$TEXMF/tex//
285
286 % Used by DMP (ditroff-to-mpx), called by makempx -troff.
287 TRFONTS = /usr/lib/font/devpost
288 MPSUPPORT = .;$TEXMF/metapost/support
289
290 % For xdvi to find mime.types and .mailcap, if they do not exist in
291 % $HOME. These are single directories, not paths.
292 % (But the default mime.types, at least, may well suffice.)
293 MIMELIBDIR = $SELFAUTOPARENT/etc
294 MAILCAPLIBDIR = $SELFAUTOPARENT/etc
295
296 % TeX documentation and source files, for use with kpsewhich.
297 TEXDOCS = .;$TEXMF/doc//
298 TEXSOURCES = .;$TEXMF/source//
299
300 % allow for compressed files, and various extenions
301 TEXDOCSSUFFIX = .:dvi:.pdf:.ps:.html:.txt:.tex
302 TEXDOCSCOMPRESS = .:gz:.bz2:.zip:.Z
303 TEXDOCEXT = ${TEXDOCSSUFFIX}${TEXDOCSCOMPRESS}
304
305 % Omega-related fonts and other files. The odd construction for OFMFONTS
306 % makes it behave in the face of a definition of TFMFONTS. Unfortunately
307 % no default substitution would take place for TFMFONTS, so an explicit
308 % path is retained.
309 OFMFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/{ofm,tfm};;$TFMFONTS
310 OPLFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/opl//
311 OVFFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/ovf//
312 OVPFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/ovp//
313 OTPINPUTS = .;$TEXMF/omega/otp//
314 OCPINPUTS = .;$TEXMF/omega/ocp//
315
316 %dvipdfm
317 DVIPDFMINPUTS = .;$TEXMF/dvipdfm//
318
319 %% t4ht utility, sharing files with TeX4ht
320 TEX4HTFONTSET=alias,iso8859

```

```

321 TEX4HTINPUTS = .;$TEXMF/tex4ht/base//;$TEXMF/tex4ht/ht-fonts/{$TEX4HTFONTSET}//
322 T4HTINPUTS= .;$TEXMF/tex4ht/base//
323 %% The mktex* scripts rely on KPSE_DOT. Do not set it in the environment.
324
325 XDVIINPUTS=.$TEXMF/{xdvi,dvips}//
326 KPSE_DOT = .
327
328 % This definition isn't used from this .cnf file itself (that would be
329 % paradoxical), but the compile-time default in paths.h is built from it.
330 % The SELFAUTO* variables are set automatically from the location of
331 % argv[0], in kpse_set_programe.
332 %
333 % About the /. construction:
334 % 1) if the variable is undefined, we'd otherwise have an empty path
335 % element in the compile-time path. This is not meaningful.
336 % 2) if we used $VARIABLE, we'd end up with // if VARIABLE is defined,
337 % which would search the entire world.
338 %
339 % The TETEXDIR stuff isn't likely to be relevant unless you're using teTeX,
340 % but it doesn't hurt.
341 %
342 TEXMFCNF = .;$VARTEXMF/web2c;{$SELFAUTOLOC,$SELFAUTODIR,$SELFAUTOPARENT}\
343 {,{/share,}/texmf{.local,}/web2c};c:/TeX/texmf/web2c
344
345
346 % Suggestions for editor settings under Windows. Uncomment your
347 % preferred option. The corresponding MFEDIT can also be set for use with
348 % Metafont.
349 %
350 % Winedt:
351 % TEXEDIT=C:\WinEdt\WinEdt.exe "[Open('%s');SelLine(%d,7)]
352 % Textpad:
353 % TEXEDIT = c:\Progra~1\TextPad\System\Ddeopn32 TextPad %s(%d)
354 % UltraEdit (newer Win32 versions):
355 % TEXEDIT = uedit32 %s/%d/1
356 % WinTeXShell32:
357 % TEXEDIT = texshell.exe /l=%d %s
358 % vi, vim, gvim. here we show Windows gvim.exe:
359 % TEXEDIT = gvim.exe %s +%d
360 % PFE:
361 % TEXEDIT=pfe32/g%d %s
362 % MED:
363 % TEXEDIT=med.exe "%s" %d
364 % TSE:
365 % TEXEDIT=e32.exe "%s" -n%d
366 % Epsilon (Lugaru) http://www.lugaru.com/
367 % TEXEDIT="c:\Program Files\eps90\bin\e32.exe" +%d %s
368 % WinShell
369 % TEXEDIT=C:\Progra~1\WinShell\WinShell.exe -c %s -l %d
370
371 % For unix
372 %
373 % vi, vim, NEdit, (X)Emacs, pico, jed
374 % TEXEDIT = vi +%d %s
375 % TEXEDIT = vim +%d %s
376 % TEXEDIT = nedit +%d %s
377 % TEXEDIT = xemacs +%d %s
378
379 %(x)fte:
380 % TEXEDIT = xfte -l%d %s
381
382
383 %-----
384 % Write .log/.dvi/etc. files here, if the current directory is unwritable.
385 % TEXMFOUTPUT = /tmp
386
387 % If a dynamic file creation fails, log the command to this file, in
388 % either the current directory or TEXMFOUTPUT. Set to the

```

```

389 % empty string or 0 to avoid logging.
390 MISSFONT_LOG =          missfont.log
391
392 % Set to a colon-separated list of words specifying warnings to suppress.
393 % To suppress everything, use TEX_HUSH =          all; this is equivalent to
394 % TEX_HUSH =          checksum:lostchar:readable:special
395 % TEX_HUSH =          none
396
397 % Enable system commands via \write18{...}?
398 shell_escape =          f
399
400 % Allow TeX \openout/\openin on filenames starting with '.' (e.g., .rhosts)?
401 % a (any) : any file can be opened.
402 % r (restricted) : disallow opening "dotfiles".
403 % p (paranoid) : as 'r' and disallow going to parent directories, and
404 % restrict absolute paths to be under $TEXMFOUTPUT.
405 openout_any =          p
406 openin_any =          a
407 % Allow TeX, MF, and MP to parse the first line of an input file for
408 % the %&format construct.
409 parse_first_line =          t
410
411 % Allow TeX, eTeX, Omega to include 'src:' specials in the dvi file.
412 % These specials are used by viewers to jump from the viewer into
413 % the editor at the right page/lineno.
414 % Possible values : none auto cr display hbox math par parend vbox
415 src_specials =          none
416
417 % Disable search on multiple suffixes filenames. In many case, when 'foo.bar'
418 % is looked for, you do not want to look for 'foo.bar.tex' before. This flag
419 % disables searching for standard suffixes if the file name has already an
420 % extension of 3 characters. Default value is true (old behaviour).
421 % allow_multiple_suffixes =          f
422
423 % Enable the mktex... scripts by default? These must be set to 0 or 1.
424 % Particular programs can and do override these settings, for example
425 % dvips's -M option. Your first chance to specify whether the scripts
426 % are invoked by default is at configure time.
427 %
428 % These values are ignored if the script names are changed; e.g., if you
429 % set DVIPSMMAKEPK to 'foo', what counts is the value of the environment
430 % variable/config value 'FOO', not the 'MKTEXPK' value.
431 %
432 % MKTEXTEX =          0
433 % MKTEXPK =          0
434 % MKTEXMF =          0
435 % MKTEXTFM =          0
436 % MKOCP =          0
437 % MKOFM =          0
438
439 % What MetaPost runs to make MPX files. This is passed an option -troff
440 % if MP is in troff mode. Set to '0' to disable this feature.
441 MPXCOMMAND =          makempx
442
443
444 % Part 3: Array and other sizes for TeX (and Metafont and MetaPost).
445 %
446 % If you want to change some of these sizes only for a certain TeX
447 % variant, the usual dot notation works, e.g.,
448 % main_memory.hugetex =          20000000
449 %
450 % If a change here appears to be ignored, try redumping the format file.
451 %
452
453 %-----
454 % Memory. Must be less than 8,000,000 total.
455 %
456 % main_memory is relevant only to initex, extra_mem_* only to non-ini.

```

```

457 % Thus, have to redump the .fmt file after changing main_memory; to add
458 % to existing fmt files, increase extra_mem_*. (To get an idea of how
459 % much, try \tracingstats=2 in your TeX source file;
460 % web2c/tests/memtest.tex might also be interesting.)
461 %
462 % To increase space for boxes (as might be needed by, e.g., PiTeX),
463 % increase extra_mem_bot.
464 %
465 %
466 main_memory = 1500000 % words of inmemory available; also applies to inif&mp
467 extra_mem_top = 0 % extra high memory for chars, tokens, etc.
468 extra_mem_bot = 0 % extra low memory for boxes, glue, breakpoints, etc.
469
470 % Words of font info for TeX (total size of all TFM files, approximately).
471 font_mem_size = 200000
472
473 % Total number of fonts. Must be >= 50 and <= 2000 (without tex.ch changes).
474 font_max = 1000
475
476 % Extra space for the hash table of control sequences (which allows 10K
477 % names as distributed).
478 hash_extra = 15000
479
480 % Max number of characters in all strings, including all error messages,
481 % help texts, font names, file names, control sequences.
482 % These values apply to TeX and MP.
483
484 % Minimum pool space after TeX/MP's own strings; must be at least
485 % 25000 less than pool_size, but doesn't need to be nearly that large.
486 string_vacancies = 45000
487 max_strings = 65000 % max number of strings
488 pool_free = 47500 % min pool space left after loading .fmt
489 pool_size = 125000
490 % Hyphenation trie. As distributed, the maximum is 65535; this should
491 % work unless 'unsigned short' is not supported or is smaller than 16
492 % bits. This value should suffice for UK English, US English, French,
493 % and German (for example). To increase, you must change
494 % 'ssup_trie_opcode' and 'ssup_trie_size' in tex.ch (and rebuild TeX);
495 % the trie will then consume four bytes per entry, instead of two.
496 %
497 % US English, German, and Portuguese: 30000.
498 % German: 14000.
499 % US English: 10000.
500 %
501 trie_size = 262000
502
503 % Buffer size. TeX uses the buffer to contain input lines, but macro
504 % expansion works by writing material into the buffer and reparsing the
505 % line. As a consequence, certain constructs require the buffer to be
506 % very large. As distributed, the size is 50000; most documents can be
507 % handled within a tenth of this size.
508 buf_size = 200000
509
510 % Parameter specific to MetaPost.
511 % Maximum number of knots between breakpoints of a path.
512 % Set to 2000 by default.
513 % path_size.mpost = 10000
514
515 % These are pdftex-specific.
516 obj_tab_size = 200000 % PDF objects
517 dest_names_size=300000 % destinations
518
519 % These are Omega-specific.
520 ocp_buf_size = 500000 % character buffers for ocp filters.
521 ocp_stack_size = 10000 % stacks for ocp computations.
522 ocp_list_size = 1000 % control for multiple ocps.
523
524 % These work best if they are the same as the I/O buffer size, but it

```

```

525 % doesn't matter much. Must be a multiple of 8.
526 dvi_buf_size = 16384 % TeX
527 gf_buf_size = 16384 % MF
528
529 % It's probably inadvisable to change these. At any rate, we must have:
530 % 45 < error_line < 255;
531 % 30 < half_error_line < error_line - 15;
532 % 60 <= max_print_line;
533 % These apply to Metafont and MetaPost as well.
534 error_line = 79
535 half_error_line = 50
536 max_print_line = 79
537 stack_size = 1500 % simultaneous input sources
538 save_size = 5000 % for saving values outside current group
539 param_size = 1500 % simultaneous macro parameters
540 max_in_open = 15 % simultaneous input files and error insertions
541 hyph_size = 1000 % number of hyphenation exceptions, >610 and <32767
542 nest_size = 500 % simultaneous semantic levels (e.g., groups)
543
544
545 %-----
546 %
547 main_memory.hugetex = 1500000
548 param_size.hugetex = 1500
549 stack_size.hugetex = 1500
550 hash_extra.hugetex = 15000
551 string_vacancies.hugetex = 45000
552 pool_free.hugetex = 47500
553 nest_size.hugetex = 500
554 save_size.hugetex = 5000
555 pool_size.hugetex = 1250000
556 max_strings.hugetex = 65000
557
558 main_memory.mf = 800000
559 main_memory.mpost = 1000000
560 pool_size.mpost = 500000
561
562 buf_size.context = 200000 % needed for omega bug
563 extra_mem_bot.context = 4000000
564 extra_mem_top.context = 2000000
565 font_mem_size.context = 500000
566 hash_extra.context = 50000
567 main_memory.context = 1500000
568 max_strings.context = 100000
569 nest_size.context = 500
570 obj_tab_size.context = 300000
571 param_size.context = 5000
572 pool_free.context = 47500
573 pool_size.context = 1250000
574 save_size.context = 50000
575 stack_size.context = 5000
576 string_vacancies.context = 90000
577 % Context's metafun
578 main_memory.metafun = 2500000
579 pool_size.metafun = 1000000
580
581 % cslatex
582 main_memory.cslatex = 1500000
583 param_size.cslatex = 1500
584 stack_size.cslatex = 1500
585 hash_extra.cslatex = 15000
586 string_vacancies.cslatex = 45000
587 pool_free.cslatex = 47500
588 nest_size.cslatex = 500
589 save_size.cslatex = 5000
590 pool_size.cslatex = 1250000
591 max_strings.cslatex = 55000
592 font_mem_size.cslatex = 400000

```

```

593
594 main_memory.lambda = 1500000
595
596 % redundant. all LaTeX should be huge
597 main_memory.hugelatex = 1500000
598 param_size.hugelatex = 1500
599 stack_size.hugelatex = 1500
600 hash_extra.hugelatex = 15000
601 string_vacancies.hugelatex = 45000
602 pool_free.hugelatex = 47500
603 nest_size.hugelatex = 500
604 save_size.hugelatex = 5000
605 pool_size.hugelatex = 1250000
606 max_strings.hugelatex = 55000
607 font_mem_size.hugelatex= 400000
608
609 % standard LaTeX is itself huge
610
611 main_memory.latex = 1500000
612 param_size.latex = 1500
613 stack_size.latex = 1500
614 hash_extra.latex = 15000
615 string_vacancies.latex = 45000
616 pool_free.latex = 47500
617 nest_size.latex = 500
618 save_size.latex = 5000
619 pool_size.latex = 1250000
620 max_strings.latex = 55000
621 font_mem_size.latex= 400000
622
623 main_memory.jadetex = 1500000
624 param_size.jadetex = 1500
625 stack_size.jadetex = 1500
626 hash_extra.jadetex = 15000
627 string_vacancies.jadetex = 45000
628 pool_free.jadetex = 47500
629 nest_size.jadetex = 500
630 save_size.jadetex = 5000
631 pool_size.jadetex = 1250000
632 max_strings.jadetex = 55000
633 font_mem_size.jadetex= 400000
634
635
636 main_memory.pdfjadetex = 2500000
637 param_size.pdfjadetex = 1500
638 stack_size.pdfjadetex = 1500
639 hash_extra.pdfjadetex = 50000
640 string_vacancies.pdfjadetex = 55000
641 pool_free.pdfjadetex = 47500
642 nest_size.pdfjadetex = 500
643 save_size.pdfjadetex = 5000
644 pool_size.pdfjadetex = 1250000
645 max_strings.pdfjadetex = 55000
646
647 main_memory.xmltex = 1500000
648 param_size.xmltex = 1500
649 stack_size.xmltex = 1500
650 hash_extra.xmltex = 50000
651 string_vacancies.xmltex = 45000
652 pool_free.xmltex = 47500
653 nest_size.xmltex = 500
654 save_size.xmltex = 50000
655 pool_size.xmltex = 1250000
656 max_strings.xmltex = 55000
657
658 main_memory.pdfxmltex = 2500000
659 param_size.pdfxmltex = 1500
660 stack_size.pdfxmltex = 1500

```

```

661 hash_extra.pdfxmltex = 50000
662 string_vacancies.pdfxmltex = 45000
663 pool_free.pdfxmltex = 47500
664 nest_size.pdfxmltex = 500
665 save_size.pdfxmltex = 50000
666 pool_size.pdfxmltex = 1250000
667 max_strings.pdfxmltex = 55000
668
669 font_mem_size.pdflatex = 210000
670 main_memory.pdflatex = 1500000
671 param_size.pdflatex = 3000
672 stack_size.pdflatex = 3000
673 hash_extra.pdflatex = 15000
674 string_vacancies.pdflatex = 55000
675 pool_free.pdflatex = 47500
676 nest_size.pdflatex = 500
677 pool_size.pdflatex = 1250000
678 save_size.pdflatex = 50000
679 max_strings.pdflatex = 55000
680
681 main_memory.pdfelatex = 1500000
682 param_size.pdfelatex = 1500
683 stack_size.pdfelatex = 1500
684 hash_extra.pdfelatex = 15000
685 string_vacancies.pdfelatex = 45000
686 pool_free.pdfelatex = 47500
687 nest_size.pdfelatex = 500
688 pool_size.pdfelatex = 1250000
689 save_size.pdfelatex = 50000
690 max_strings.pdfelatex = 55000
691
692 main_memory.pdfetex = 1500000 % 1000000 bot/top
693 hash_extra.pdfetex = 50000
694 pool_size.pdfetex = 1250000
695 string_vacancies.pdfetex = 90000
696 max_strings.pdfetex = 100000
697 pool_free.pdfetex = 47500
698 nest_size.pdfetex = 500
699 param_size.pdfetex = 5000
700 save_size.pdfetex = 50000
701 stack_size.pdfetex = 5000
702 obj_tab_size.pdfetex = 256000

```