

Przewodnik po T_EX Live, wydanie 5c

Sebastian Rahtz
sebastian.rahtz@oucs.ox.ac.uk
Michel Goossens
m.goossens@cern.ch

Czerwiec 2000

Spis treści

1. Wstęp	2
1.1. Rozszerzenia T _E X-a	2
2. Struktura i zawartość CD-ROM-u	3
2.1. Drzewo TDS	3
3. Instalacja i użytkowanie pod Unixem	4
3.1. Uruchamianie T _E X Live z CD-ROM-u	5
3.2. Instalacja T _E X Live na twardym dysku	6
3.3. Instalacja pojedynczych pakietów z T _E X Live na twardym dysku	9
3.4. Program texconfig	10
4. Instalacja i użytkowanie pod Windows	10
4.1. Uruchamianie T _E X Live z CD-ROM-u	10
4.2. Instalacja na twardym dysku	11
5. Budowa systemu na nowej platformie Unixowej	11
5.1. Warunki wstępne	11
5.2. Konfiguracja	11
5.3. Uruchamianie make	12
5.4. Końcowe kroki konfiguracyjne	13
6. Instrukcja obsługi systemu Web2c	13
6.1. Przeszukiwanie ścieżek przez Kpathsea	15
6.2. Bazy nazw plików	18
6.3. Parametry kontrolujące działanie programów	27
7. Historia i podziękowania	27
8. Przyszłe wersje	29

Spis tabel

1. Typy plików <code>Kpathsea</code>	20
--	----

1. Wstęp

Niniejsza dokumentacja opisuje CD-ROM **T_EX Live 5** – dystrybucję T_EX-a/L^AT_EX-a dla systemów Unix 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), który może być używany prawie z każdą instalacją T_EX-a.

Pakiet T_EX-owy zawiera programy w implementacji Web2c (w zmodyfikowanej wersji 7.3), co ma na celu jak największe ułatwienie T_EX-owania; wykorzystuje ponadto zalety wydajnej i konfigurowalnej biblioteki Kpathsea, autorstwa Karla Berry’ego i Olafa Webera. Można go 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), xdvi (przeglądarka dla X Window), dviIj (sterownik do 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 autorstwa Angus Duggana.

1.1. Rozszerzenia T_EX-a

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

1. ϵ -T_EX, który dodaje niewielki, lecz potężny zestaw nowych poleceń wbudowanych, oraz rozszerzenie T_EX-~~X_EL~~ 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 16-bitowymi, używając 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/doc2.tex](#).

ϵ -T_EX jest stabilny (wersja 2.1), chociaż nowsze wersje dadzą pewnie nieco więcej funkcjonalności. pdfT_EX (wersja 0.14f) i Ω (wersja 1.11) są ciągle rozwijane; wersje znajdujące się na tym CD-ROM-ie są najbardziej aktualne (stan z połowy marca 2000).

2. Struktura i zawartość CD-ROM-u

Poniżej wymienione są ważniejsze podkatalogi głównego katalogu CD-ROM.

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

tl doc dokumentacja \TeX Live;

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

info dokumentacja w formacie GNU „info” dla systemu \TeX ;

man dokumentacja w formacie Unixowych stron „man” dla systemu \TeX ;

source źródła wszystkich programów, włącznie z głównymi dystrybucjami Web2c \TeX -a i META-FONT-a. Są one spakowane w archiwum tar;

support różne programy związane z \TeX -em, które *nie* są domyślnie instalowane, takie jak Musix \TeX , programy pomocnicze, oraz kompletna dystrybucja Ghostscript – interpretera języka PostScript w wersji 5.50;

systems spakietowane systemy \TeX -owe, które są odseparowane od głównego \TeX Live. Podkatalogami są tutaj:

msdos pakiet emTeX dla DOS i OS/2;

os2 pakiet emTeX/TDS dla OS/2 (dostosowanie implementacji emTeX do struktury katalogów zgodnej z TDS).

texmf główne drzewo z makrami, czcionkami i dokumentacją;

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

Są także dwa skrypty instalacyjne dla systemów Unixowych, `install-cd.sh` i `install-pkg.sh`; będą one omówione w rozdziale 3 na str. 4.

2.1. Drzewo TDS

Drzewo `texmf` na \TeX Live składa się z różnych „zestawów”, z których każdy posiada zbiór „pakietów” (jest ich na CD-ROM-ie ponad 400). Normalna instalacja pozwala użytkownikowi skopiować z CD-ROM-u na twardy dysk cały zestaw, ale jest też możliwe zainstalowanie tylko jednego pakietu. Oto dostępne zestawy:

ams pakiety makr i fonty Amerykańskiego Towarzystwa Matematycznego;

bibtex Bib \TeX -owe style i bazy danych;

doc ogólne przewodniki i dokumentacje w różnych formatach, także w HTML i PDF;

dvips programy pomocnicze do sterownika DVI-to-PostScript T. Rokickiego;

etex programy pomocnicze do ϵ - \TeX -a;

fonts pliki źródłowe fontów, pliki metryczne, pliki PostScriptowe i bitmapowe;

formats Eplain, Rev \TeX , phyzzx, texsis, alateX, text1, lollipop itp.;

generic dodatkowe makra do użycia z dowolnym formatem;

graphics pakiety makr dla obsługi grafiki;

lang programy pomocnicze dla języków innych niż angielski;

latex \LaTeX , łącznie z oficjalnymi narzędziami i wszystkimi dostarczonymi pakietami do $\LaTeX 2_{\epsilon}$;

metapost pliki pomocnicze dla METAPOST-a;

omega pliki pomocnicze dla Ω ;

pdftex pliki pomocnicze dla pdf \TeX -a;

plain makra dla plain \TeX -a;

systems binaria dla różnych wersji Unixa (w tym Linux) i platform Win32;

texlive podstawowe materiały dystrybucji.

Każdy z zestawów podzielony jest na *podstawowe – basic* (1), *zalecane – recommended* (2) oraz *inne – other* (3). Wszystkie pakiety w zestawie latex1 są niezbędne aby rozpocząć pracę z \LaTeX -em, pakiety w latex2 są zalecane dla większości użytkowników, a latex3 zawiera pakiety opcjonalne. Katalog texmf/tpm zawiera spisy wszystkich plików w każdym z pakietów (spisy te wykorzystywane są przez programy instalacyjne).

3. Instalacja i użytkowanie pod Unixem

CD-ROM \TeX Live można używać na trzy sposoby:

1. można zamontować (mount) CD-ROM w systemie plików, zmodyfikować zmienną PATH, i uruchamiać wszystko z CD-ROM-u; zajmuje to bardzo mało miejsca na dysku i pozwala na natychmiastowy dostęp do wszystkiego na CD-ROM-ie. Chociaż wydajność nie będzie optymalna, jest to w pełni akceptowalne np. na komputerach PC z Linuxem;
2. można 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 10 megabajtów, albo 100 megabajtów w przypadku instalacji zalecanej);
3. można doinstalować wybrane pakiety do istniejącego systemu \TeX lub systemu \TeX Live instalowanego wcześniej.

Każda z tych metod jest dokładnie opisana w dalszych rozdziałach.

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.

3.1. Uruchamianie \TeX Live z CD-ROM-u

Struktura 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 katalogów.

DEC Alpha OSF/1 (4.0)	alphaev5-osf4.0d
HP9000 HPUX 10.20	hppa20-hpux10.20
Intel x86 z GNU/Linux	i386-linux
Intel x86 z FreeBSD ELF 3.4	i386-freebsd
SGI IRIX (6.5)	mips-irix6.5
IBM RS 6000 AIX 4.2*	rs6000-aix4.2.1.0
Sun Sparc Solaris (2.7)	sparc-solaris2.7
Windows 95/98/2000/NT (Intel)	win32

Wydawałoby się, że późniejsze wygenerowanie czcionek czy zmiana konfiguracji może spowodować problemy, ponieważ nie można zmieniać plików na CD-ROM-ie. Można jednak posiadać na dysku drugie, równoległe drzewo \TeX -owe, w którym można dokonywać zmian. Drzewo takie jest przeszukiwane przed głównym drzewem CD-ROM-u. Domyślną lokalizacją jest `texmf-localconfig` na CD-ROM-ie (katalog, który nie istnieje!), wobec tego *musisz* ją zmienić ustawiając zmienną środowiskową `VARTEXMF`.

Tak więc użytkownicy `sh` lub `bash` na Intel PC z systemem Linux mogą zamontować CD-ROM \TeX Live w katalogu `/cdrom` za pomocą polecenia:

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

Następnie należy uaktualnić zmienną PATH, dołączając do ścieżki przeszukiwania katalog zawierający binaria dla danej architektury.

```
PATH=/cdrom/bin/i386-linux:$PATH
export PATH
VARTEXMF=/usr/TeX.local
```

Dla wygody, powyższe linijki można dodać do skryptu `.profile`.

W razie wątpliwości najlepiej zwrócić się do lokalnego guru systemowego o pomoc przy instalowaniu CD-ROM-u i w wyborze katalogu dla systemu \TeX .

Odpowiednie pliki pomocnicze zostaną zainstalowane na twardym dysku za pierwszym razem, kiedy tylko będą potrzebne. Aby wszystko zainicjować i sprawdzić czy działa, warto od razu uruchomić skrypt `texconfig`.

3.2. Instalacja \TeX 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.
```

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

1. typu systemu operacyjnego, na którym chcesz dokonać instalacji;
2. zestawów, które mają być zainstalowane; na poziomie „podstawowym” (*basic*), „zalecanym” (*recommended*) albo „innym” (*other*);
3. miejsca na dysku przeznaczonego na pliki;
4. ustawienia trybu pracy niektórych programów.

Opcje wybiera się przez wpisanie litery lub liczby i potwierdzenie jej klawiszem `Enter`. Przykładowo: wykryty został system Linux ELF, wybrany został domyślny tryb *zalecany* do wszystkich zestawów, domyślny katalog instalacji to `/usr/local`; 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 10 megabajtów; można ją poszerzyć wybranymi pakietami, kiedy zajdzie taka potrzeba.

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

Pozycja `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.

W przypadku wybrania `<C>` – *collections*, wyświetlone zostaną dostępne zestawy, wybrany poziom instalacji oraz wymagane miejsce na dysku (rys. 2). Możliwe jest ustawienie oddzielnych poziomów instalacji dla każdego z zestawów, w zakresie od *none* (nic) do *all* (wszystko). Można wybrać zarówno wszystkie zestawy od razu, jak i konkretny zestaw i ustawić dla niego poziom instalacji (rys. 3).

```

===== TeX Live installation procedure =====

==> 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
<C> collections:      24 out of 35, disk space required: 193176 kB
<S> systems:         1 out of 8, disk space required: 8355 kB
                        total disk space required: 201531 kB
<L> install level (1: basic, 2: recommended, 3: all): 2
<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
<I> start installation, <H> help, <Q> quit

Enter command:

```

Rysunek 1: Główny ekran skryptu instalacyjnego

Po zakończeniu ustawień należy powrócić do głównego ekranu i rozpocząć instalację. Program instalacyjny, mając na uwadze wybrane zestawy i systemy, sprawdzi spisy plików na CD-ROM-ie, stworzy listę plików do skopiowania, po czym pliki zostaną skopiowane na twarde dyski. Następnie uruchamiana jest sekwencja inicjująca (tworzenie plików formatów itd.). Po zakończeniu wystarczy dodać do zmiennej PATH właściwy podkatalog bin instalacji TeX-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 A) i zmienić linię znajdującą się blisko początku pliku:

```
TEXMFMAIN = $SELFAUTOPARENT
```

na

```
TEXMFMAIN = $SELFAUTODIR
```

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

	name	selection	size
<1>	bibtex	[recommended]	7597 kB
<2>	doc	[recommended]	21152 kB
<3>	dvips	[recommended]	430 kB
<4>	etex	[recommended]	102 kB
<5>	fonts	[recommended]	51447 kB
<6>	formats	[recommended]	14651 kB
<7>	generic	[recommended]	459 kB
<8>	graphics	[recommended]	9674 kB
<9>	lang	[recommended]	19618 kB
<U>	latex	[recommended]	23429 kB
<V>	metapost	[recommended]	1443 kB
<W>	omega	[recommended]	4986 kB
<X>	pdftex	[recommended]	471 kB
<Y>	plain	[recommended]	1113 kB
<Z>	texlive	[recommended]	10155 kB
		SUM:	166829 kB

```

global commands: select <N>one / <B>asic / R<E>commended / <A>ll
                  for all collections
<R>  return to platform menu
<Q>  quit

```

Rysunek 2: Wybór zestawów

```

Collection: Fonts
=====
Fonts, including metrics, virtual fonts and sources
=====
<N> No packages
<B> Basic packages           [ 1023 kB]
<E> Basic + Recommended packages [ 51447 kB]
<A> All packages             [127417 kB]
=====
<R>  return to collection menu
<Q>  quit
Enter command:

```

Rysunek 3: Wybór zawartości zestawu

3.3. Instalacja pojedynczych pakietów z \TeX Live na twardym dysku

CD-ROM \TeX 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 ma dziewięć parametrów; pierwsze cztery pozwalają wybrać: pojedynczy pakiet do instalacji, cały zestaw (np. `ams2`), nazwę katalogu zamontowanego CD-ROM-u, oraz nazwę katalogu zawierającego pliki z listami (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 `mktextlsr`, który przebudowuje bazę danych o plikach, a czwarty powoduje jedynie wypisanie plików, które miałyby być zainstalowane:

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

Wreszcie można wyszczególnić, 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/latex3/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 \LaTeX -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 „inne” (*other*) Plain T_EX:

```
>> sh install-pkg.sh --collection=plain3
```

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

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

3.4. 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 instalacyjny powinien być używany dla wszelkich zmian, takich jak ustawianie drukarek czy przebudowywanie bazy danych o plikach. Obydwa tryby pracy mają tekst pomocy opisujący możliwości.

4. Instalacja i użytkowanie pod Windows

Rozdział ten odnosi się tylko do systemów Windows 9x lub NT. W przypadku użytkowania Windows 3.1, należy ręcznie zainstalować emTeX (z podkatalogu `systems` katalogu głównego CD-ROM).

Konieczne jest także sprawdzenie, czy Windows potrafi, przy czytaniu CD-ROM-ów, korzystać z rozszerzeń Microsoft Joliet. W aplikacji Explorer (Eksplorator) wystarczy sprawdzić, czy wyświetlana lista plików CD-ROM-u posiada „długie” nazwy, zapisane za pomocą małych i wielkich liter. Jeżeli tak nie jest, nie można używać systemu uruchamialnego 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. Uruchamianie z CD-ROM-u

Wszystkie programy T_EX-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 potrzebnego pliku formatu. Ponadto, w pomocniczym katalogu będą deponowane automatycznie generowane fonty. Wymienione kroki przygotowawcze wykonuje program TeXSetup.exe uruchamiany z katalogu `setupw32/` CD-ROM-u. Po uruchomieniu programu i wyborze katalogu pomocniczego należy wybrać opcję „Run CD”. Na zakończenie wymagany jest restart systemu. Można teraz uruchamiać programy z linii poleceń lub użyć edytora, w którym programy są uruchamiane za pomocą wygodnych menu.

4.2. Instalacja na twardym dysku

Instalację rozpoczyna się od uruchomienia w katalogu `setupw32/` programu `TeXSetup.exe`, który przeszukuje listę zestawów i pakietów znajdującą się na CD-ROM-ie. Program `TeXSetup.exe` pozwala na wybranie poziomu instalacji dla każdego z zestawów (opis zestawów i pakietów znajduje się w rozdziale 2.1), oraz pominięcie instalowania dokumentacji i/lub źródeł pakietów w wypadku posiadania ograniczonej przestrzeni dyskowej. Podczas instalacji program pyta, w których katalogach umieścić główną dystrybucję i lokalną konfigurację. Dodatkowo można zainstalować edytor dla \TeX -a (`TeXshell`, `PFE` lub `Emacs`) oraz PostScriptową przeglądarkę `Ghostscript`.

Należy być świadomym, że wybór rozmiaru klastra (*cluster*) na dysku DOS-owym może mieć radykalny wpływ na rozmiar instalacji \TeX -a. Drzewo z plikami \TeX -owymi zawiera setki małych plików i często zdarza się, że kompletna instalacja zajmuje do czterech razy więcej miejsca niż objętość samego CD-ROM-u.

Po zakończeniu instalacji należy ponownie uruchomić Windows, i można już uruchamiać programy \TeX -owe zarówno z linii poleceń, jak i z poziomu menu zainstalowanego edytora.

Po pierwszej instalacji (dowolnego poziomu) 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.

Szczegółowe informacje na temat konfiguracji oprogramowania w systemach Windows można znaleźć w texmf/doc/html/ftpex/ftpex.html.

5. 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).

5.1. 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.

5.2. 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`.

5.3. Uruchamianie `make`

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

```
>> make world
```

i wygodnie poczekać...

Możliwy 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
```

5.4. 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
```

6. Instrukcja obsługi systemu Web2c

Web2c to zestaw programów związanych z T_EX-em, tj. sam T_EX, METAFONT, METAPOST, BibT_EX itp. Oryginalna implementacja wykonana została przez Tomasa Rokickiego, który w roku 1987 stworzył pierwszy system T_EX-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 **T_EX Live**.

Web2c 7.3 działa na platformach systemowych takich jak Unix, Windows 3.1, 9x/NT, DOS, Amiga i innych. System wykorzystuje oryginalne źródła T_EX-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 T_EX-em. Podstawowymi składnikami systemu są:

bibtex tworzenie spisów bibliograficznych;

dmp konwersja troff do MPX (rysunki METAPOST-owe);

dvicopy 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 T_EX-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 T_EX-owego drzewa podkatalogów biblioteka ta używa kombinacji zmiennych środowiskowych oraz kilku plików konfiguracyjnych. Web2c w wersji 7.3 potrafi obsługiwać jednocześnie więcej niż jedno drzewo podkatalogów, co jest użyteczne w przypadku, gdy chce się przechowywać standardową dystrybucję T_EX-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.

6.1. 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 18) 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.

Jeżeli 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 T_EX-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 przeszukiwanie 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.

6.1.1. 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 24).

6.1.2. 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 A, który zaczyna się na stronie 30, 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; umożliwi 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;}//

```

6.1.3. 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.

6.1.4. Rozwijanie domyślne

Jeżeli ścieżka przeszukiwania największego uprzywilejowania (patrz „Źródła ścieżek” na stronie 15) zawiera *dotykowy 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
```

kończącą 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.

6.1.5. Rozwijanie nawiasów

Użyteczną cechą jest możliwość rozwijania nawiasów, co oznacza, że np. `v{a,b}` rozwija się do `vav:vbw`. Możliwe jest zagnieżdżanie nawiasów. Funkcja ta może być użyta do zaimplementowania różnych hierarchii \TeX -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 `TEX`-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.

6.1.6. 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.

6.1.7. 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.

6.2. 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.

6.2.1. 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.

6.2.2. `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 odgadywany 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 środowiskowe¹, 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) METAFONT-a	MFBASES, TEXMFINI	.base
bib	Źródła bibliograficzne Bib \TeX -a	BIBINPUTS, TEXBIB	.bib
bst	Pliki stylu Bib \TeX	BSTINPUTS	.bst
cnf	Pliki konfiguracyjne	TEXMFCNF	.cnf
dvips config	Pliki konfiguracyjne dvips-a, np. config.ps i psfonts.map	TEXCONFIG	.map
fmt	Formaty \TeX -a	TEXFORMATS, TEXMFINI	.fmt, .efmt, .efm
gf	fonty GF (bitmapowe)	FONTS, GFFONTS, GLYPHFONTS, TEXFONTS	.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
mp	Pliki źródłowe METAPOST-a	MPINPUTS	.mp
mppool	pliki napisów METAPOST-a	MPPPOOL, TEXMFINI	.pool
MetaPost support	pliki pomocnicze METAPOST-a	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	OTPINPUTS	.otp
ovf	Fonty wirtualne Ω	OVFFONTS, TEXFONTS	.ovf
ovp	Wirtualne listy właściwości Ω	OVPFONTS, TEXFONTS	.ovp
pk	Fonty spakowane (PK)	programFONTS (gdzie program to XDVI, etc.), PKFONTS, TEXPKS, GLYPHFONTS, TEXFONTS	.pk
PostScript header	Ładowalne programy PostScript-owe	TEXPSHEADERS, PSHEADERS	.pro, .enc
tex	Pliki źródłowe dla \TeX -a	TEXINPUTS	.tex, .cls, .sty, .clo, .def
TeX system documentation	Dokumentacja systemu \TeX	TEXDOCS	

¹Definicje zmiennych środowiskowych znaleźć można w pliku texmf.cnf (str. 30).

Typy plików Kpathsea *cd.*

Nazwa	Opis	Zmienne	Rozszerzenia
TeX system sources	Pliki źródłowe systemu T _E X	TEXSOURCES	
texpool	pliki napisów T _E X-a	TEXPOOL, TEXMFINI	.pool
tfm	Metryki fontów T _E X-owych	TFMFonts, TEXFonts	.tfm
Troff fonts	Fonty dla Troff, używane przez DMP	TRFonts	
truetype fonts	Fonty TrueType	TTFonts	.ttf, .ttc
type1 fonts	Fonty Type 1 (PostScript-owe)	T1Fonts, T1Inputs, TEXPSHeaders, DVIPSHeaders	.pfa, .pfb
type42 fonts	Fonty Type 42 (PostScript-owe)	T42Fonts	
vf	Fonty wirtualne	VFFonts, TEXFonts	.vf
web2c files	Pomocnicze pliki Web2c	WEB2C	
other text files	Pliki tekstowe użytkownika foo	FOOInputs	
other binary files	Pliki binarne użytkownika 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 odnaleźć 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*”.

6.2.3. 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 odnajdywane 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`).

Zwrócimy teraz naszą uwagę na pliki nagłówkowe i konfiguracyjne programu `dvips`. Najpierw szukamy pliku PostScript-owego prologu `tex.pro`, wykorzystywanego dla potrzeb T_EX-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.

6.2.4. 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 zmienną: `SET KPATHSEA_DEBUG_OUTPUT=err.log`.)

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) znajdują się na rys. 4. Program `dvips` zaczyna pracę od zlokalizowania potrzebnych mu plików. Najpierw znajduje plik `texmf.cnf`, który zawiera ścieżki przeszukiwania dla innych plików. Potem znajduje


```

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 4: Szukanie plików konfiguracyjnych

```

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 5: 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 6: Szukanie fontu

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 6.2.3).

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]

```

6.3. Parametry kontrolujące działanie programów

Inną użyteczną cechą Web2c jest możliwość ustalania wielu parametrów określających wielkość pamięci za pomocą pliku `texmf.cnf`. Załącznik A zaczynający się na stronie 30 zawiera listing pliku `texmf.cnf`. Ustawienia wszystkich parametrów znajdują się w części trzeciej pliku. Najważniejszymi zmiennymi są:

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

Dla każdego nowego ustawienia tej zmiennej należy wykonać nowy format. Przykładowo, możesz wygenerować „ogromną” wersję formatu T_EX 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 T_EX-a, takie jak: pudełka, kleje itd.; przydatna, zwłaszcza w przypadku korzystania z pakietu P_ACT_EX.

font_mem_size Wielkość pamięci przeznaczona przez T_EX-a na informacje o fontach. Jest to mniej więcej ogólna wielkość wczytywanych przez T_EX-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 odsyłacze. 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 T_EX-a i dlatego powyższe parametry stanowią praktyczny kompromis, pozwalając na pewną elastyczność.

7. Historia i podziękowania

Niniejszy CD-ROM jest rezultatem wspólnego przedsięwzięcia Międzynarodowej Grupy Użytkowników Systemu T_EX (*TEX Users Group*), Brytyjskiej Grupy Użytkowników Systemu T_EX (*UK TEX Users Group*), Francuskojęzycznej Grupy Użytkowników T_EX-a (*GUTenberg*), Niemieckiej Grupy Użytkowników T_EX-a (*DANTE e.V.*), z pomocą grup: czeskiej/słowackiej, holenderskiej, indyjskiej i polskiej. Dyskusje nad projektem rozpoczęły się pod koniec 1993 roku, kiedy Holenderska Grupa Użytkowników T_EX-a rozpoczęła prace nad swoim 4AllT_EX 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 4AllT_EX CD-ROM, a także spowodował powstanie Grupy Roboczej TUG ds. Standardu Katalogów T_EX-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 pakietu $\text{te}\TeX$ 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 pakietu $\text{te}\TeX$ przez Thomasa Essera. W związku z tym zdecydowaliśmy się oprzeć drugie wydanie CD-ROM na standardowej bibliotece Web2c, z dodaniem skryptu `texconfig` z pakietu $\text{te}\TeX$. Trzecie wydanie CD-ROM było oparte na Web2c 7.2, przygotowanej przez Olafa Webera. W tym samym czasie została przygotowana nowa wersja $\text{te}\TeX$ -a i **\TeX Live** udostępnił prawie wszystkie jego nowe funkcje. Czwarta edycja była przygotowana podobnie, wykorzystując nową wersję $\text{te}\TeX$ -a i nową wersję Web2c (7.3). Edycja piąta (marzec 2000) zawiera wiele poprawek i uzupełnień. Zaktualizowano setki pakietów, Omega i $\text{pdf}\TeX$ pojawiły się w nowych wersjach, dotyczy to także wielu programów pomocniczych (np. `xdvi`, `dvips` i `tex4ht`).

Główna zmiana w \TeX Live5 to 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.

Szczególne słowa podziękowania 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, na których można było kompilować \TeX -a, znajdujących się w Florida State University Supercomputer Research Institute. 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 $\text{te}\TeX$, bez którego ten CD-ROM z całą pewnością by nie powstał, a także którego stała pomoc umożliwiła uczynienie z tego CD 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 to wydanie w imieniu TUG;
- Petrowi Olšákowi, który koordynował i starannie sprawdzał przygotowanie czeskich i słowackich pakietów;
- Fabrice Popineau, który niezmordowanie przygotowywał pakiety dla systemów win32 (szczególnie setup!) i wniósł wielki wkład w formie pomysłów, porad i kodu;
- Staszce Wawrykiewiczowi, za wielkie wsparcie w sprawdzaniu dystrybucji oraz za koordynację i przygotowanie polskich pakietów;
- Olałowi Weberowi, za cierpliwe pielęgnowanie Web2c 7.3;
- Grahamowi Williamsowi, za pracę nad katalogiem pakietów.

Alain Rabaute, Pascal Quignon, Gerhard Wilhelms, Fabrice Popineau, Janka Chlebková, Staszek Wawrykiewicz, Erik Frambach i Ulrik Vieth, przetłumaczyli dokumentację na różne języki. Ponadto sprawdzili oryginalną dokumentację oraz zgłosili wiele cennych uwag i poprawek.

8. 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>.

A. 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,$TEXMFLOCAL,!!$VARTEXMF,!!$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 TEXMFDDBS = $TEXMF
91
92 % It may be convenient to define TEXMF like this:
93 %   TEXMF = {$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 TEXMFDDBS
96 % like this:
97 %   TEXMFDDBS = $HOMETEXMF;$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.pdflatex = .;$TEXMF/{pdfetex,tex}/{latex,generic,}//
147 TEXINPUTS.pdfetex = .;$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 = .;$TEXMF/{pdftex,tex}/{cslatex,csplain,latex,generic,}//
162 TEXINPUTS.pdfcsplain = .;$TEXMF/{pdftex,tex}/{csplain,plain,generic,}//
163
164 % Polish
165 TEXINPUTS.platex = .;$TEXMF/tex/{platex,latex,generic,}//
166 TEXINPUTS.pdfmex = .;$TEXMF/{pdftex,tex}/{mex,plain,generic,}//
167 TEXINPUTS.mex = .;$TEXMF/tex/{mex,plain,generic,}//
168 TEXINPUTS.pdfplatex = .;$TEXMF/{pdftex,tex}/{platex,latex,generic,}//
169
170 % french
171 TEXINPUTS.frtex = .;$TEXMF/{mltex,tex}/{plain,generic,}//
172 TEXINPUTS.frlatex = .;$TEXMF/{mltex,tex}/{frlatex,latex,generic,}//
173
174 % MLTeX
175 TEXINPUTS.mltex = .;$TEXMF/{mltex,tex}/{plain,generic,}//
176 TEXINPUTS.mllatex = .;$TEXMF/{mltex,tex}/{latex,generic,}//
177
178 % odd formats needing their own paths
179 TEXINPUTS.lollipop = .;$TEXMF/tex/{lollipop,generic,plain,}//
180 TEXINPUTS.lamstex = .;$TEXMF/tex/{lamstex,generic,plain,}//
181
182 % David Carlisle's xmltex
183 TEXINPUTS.xmltex = .;$TEXMF/tex/{xmltex,latex,generic,}//
184 TEXINPUTS.pdfxmltex = .;$TEXMF/{pdftex,tex}/{xmltex,latex,generic,}//
185
186 % Sebastian Rahtz' jadetex for DSSSL
187 TEXINPUTS.pdfjadetex = .;$TEXMF/{pdftex,tex}/{jadetex,generic,plain,}//
188 TEXINPUTS.jadetex = .;$TEXMF/tex/{jadetex,generic,plain,}//
189
190 % Earlier entries override later ones, so put this last.
191 TEXINPUTS = .;$TEXMF/tex/{generic,}//
192
193 % Metafont, MetaPost inputs.
194 MFINPUTS = .;$TEXMF/metafont//;{$TEXMF/fonts,$VARTEXFONTS}/source//
195 MPINPUTS = .;$TEXMF/metapost//
196
197 % Dump files (fmt/base/mem) for vir{tex,mf,mp} to read (see web2c/INSTALL),
198 % and string pools (.pool) for ini{tex,mf,mp}. It is silly that we have six

```

```

199 % paths and directories here (they all resolve to a single place by default),
200 % but historically ...
201 TEXFORMATS = .;$TEXMF/web2c
202 MFBASES = .;$TEXMF/web2c
203 MPMEMS = .;$TEXMF/web2c
204 TEXPOOL = .;$TEXMF/web2c
205 MFPOOL = .;$TEXMF/web2c
206 MPPPOOL = .;$TEXMF/web2c
207
208 % Device-independent font metric files.
209 VFFONTS = .;$TEXMF/fonts/vf//
210 TFMFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/tfm//
211
212 % The $MAKETEX_MODE below means the drivers will not use a cx font when
213 % the mode is ricoh. If no mode is explicitly specified, kpse_prog_init
214 % sets MAKETEX_MODE to /, so all subdirectories are searched. See the manual.
215 % The modeless part guarantees that bitmaps for PostScript fonts are found.
216 PKFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/pk/{$MAKETEX_MODE,modeless}//
217
218 % Similarly for the GF format, which only remains in existence because
219 % Metafont outputs it (and MF isn't going to change).
220 GFFONTS = .;$TEXMF/fonts/gf/$MAKETEX_MODE//
221
222 % A backup for PKFONTS and GFFONTS. Not used for anything.
223 GLYPHFONTS = .;$TEXMF/fonts
224
225 % For texfonts.map and included map files used by mktexpk.
226 % See ftp://ftp.tug.org/tex/fontname.tar.gz.
227 TEXFONTMAPS = .;$TEXMF/fontname
228
229 % BibTeX bibliographies and style files.
230 BIBINPUTS = .;$TEXMF/bibtex/{bib,}//
231 BSTINPUTS = .;$TEXMF/bibtex/{bst,}//
232
233 % PostScript headers, prologues (.pro), encodings (.enc) and fonts;
234 % this is also where pdftex finds included figures files!
235
236 TEXPSHEADERS.pdflatex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
237 TEXPSHEADERS.pdfelatex = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
238 TEXPSHEADERS.pdfTEXinfo = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
239 TEXPSHEADERS.pdfcsLATEX = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
240 TEXPSHEADERS.pdfcsPLAIN = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
241 TEXPSHEADERS.pdfETEX = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
242 TEXPSHEADERS.pdfJADETEX = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
243 TEXPSHEADERS.pdfPLATEX = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
244 TEXPSHEADERS.pdfXMLTEX = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
245 TEXPSHEADERS.pdfMEX = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
246 TEXPSHEADERS.pdfTEX = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
247 TEXPSHEADERS.pdfTEXinfo = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
248 TEXPSHEADERS.cont-de = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//

```

```

249 TEXPSHEADERS.cont-en = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
250 TEXPSHEADERS.cont-nl = .;$TEXMF/{tex,pdftex,dvips,fonts/type1}//
251 TEXPSHEADERS.context = .;$TEXMF/{etex,tex,pdftex,dvips,fonts/type1}//
252 TEXPSHEADERS = .;$TEXMF/{dvips,fonts/type1,pdftex}//
253
254 % PostScript Type 1 outline fonts.
255 T1FONTS = .;$TEXMF/fonts/type1//;$TEXMF/fonts/misc/hbf//
256
257 % PostScript AFM metric files.
258 AFMFONTS = .;$TEXMF/fonts/afm//
259
260 % TrueType outline fonts.
261 TTFONTS = .;$TEXMF/fonts/truetype//
262 TTF2TFMINPUTS = .;$TEXMF/ttf2pk//
263
264 % Type 42 outline fonts.
265 T42FONTS = .;$TEXMF/fonts/type42//
266
267 % A place to puth everything that doesn't fit the other font categories.
268 MISCFONTS = .;$TEXMF/fonts/misc//
269
270 % Dvips' config.* files (this name should not start with 'TEX!').
271 TEXCONFIG = .;$TEXMF/dvips//
272
273 % Makeindex style (.ist) files.
274 INDEXSTYLE = .;$TEXMF/makeindex//
275
276 % Used by DMP (ditroff-to-mpx), called by makempx -troff.
277 TRFONTS = /usr/lib/font/devpost
278 MPSUPPORT = .;$TEXMF/metapost/support
279
280 % For xdvi to find mime.types and .mailcap, if they do not exist in
281 % $HOME. These are single directories, not paths.
282 % (But the default mime.types, at least, may well suffice.)
283 MIMELIBDIR = $SELFAUTOPARENT/etc
284 MAILCAPLIBDIR = $SELFAUTOPARENT/etc
285
286 % TeX documentation and source files, for use with kpsewhich.
287 TEXDOCS = .;$TEXMF/doc//
288 TEXSOURCES = .;$TEXMF/source//
289
290 % allo for compressed files, and various extenions
291 TEXDOCSSUFFIX = .dvi:.ps:.html:.txt
292 TEXDOCSCOMPRESS = .gz:.bz2:.zip:.Z
293 TEXDOCEXT = {${TEXDOCSSUFFIX}-${TEXDOCSCOMPRESS}}
294
295 % Omega-related fonts and other files. The odd construction for OFMFONTS
296 % makes it behave in the face of a definition of TFMFONTS. Unfortunately
297 % no default substitution would take place for TFMFONTS, so an explicit
298 % path is retained.

```

```

299 OFMFFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/{ofm,tfm};//;$TFMFFONTS
300 OPLFFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/opl//
301 OVFFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/ovf//
302 OVPPFONTS = .;{$TEXMF/fonts,$VARTEXFONTS}/ovp//
303 OTPINPUTS = .;$TEXMF/omega/otp//
304 OCPINPUTS = .;$TEXMF/omega/ocp//
305
306 %% t4ht utility, sharing files with TeX4ht
307 TEX4HTFONTSET={alias,iso8859}
308 TEX4HTINPUTS = .;$TEXMF/tex4ht/base//;$TEXMF/tex4ht/ht-fonts/{$TEX4HTFONTSET}//
309 T4HTINPUTS= .;$TEXMF/tex4ht/base//
310 %% The mktex* scripts rely on KPSE_DOT. Do not set it in the environment.
311 KPSE_DOT = .
312
313 % This definition isn't used from this .cnf file itself (that would be
314 % paradoxical), but the compile-time default in paths.h is built from it.
315 % The SELFAUTO* variables are set automatically from the location of
316 % argv[0], in kpse_set_progname.
317 %
318 % About the /. construction:
319 % 1) if the variable is undefined, we'd otherwise have an empty path
320 % element in the compile-time path. This is not meaningful.
321 % 2) if we used /$VARIABLE, we'd end up with // if VARIABLE is defined,
322 % which would search the entire world.
323 %
324 % The TETEXDIR stuff isn't likely to be relevant unless you're using teTeX,
325 % but it doesn't hurt.
326 %
327 TEXMFCNF = .;{$SELFAUTOLOC,$SELFAUTODIR,$SELFAUTOPARENT}\
328 {,{/share,}/texmf{.local,}/web2c};c:/TeX/texmf/web2c
329
330
331 % Suggestions for editor settings under Windows. Uncomment your
332 % preferred option. The corresponding MFEDIT can also be set for use with
333 % Metafont.
334 %
335 % Winedt:
336 % TEXEDIT=C:\WinEdt\WinEdt.exe "[Open('%s');Selline(%d,7)]
337 % Textpad:
338 % TEXEDIT = c:\Progra~1\TextPad\System\Ddeopn32 TextPad %s(%d)
339 % UltraEdit (newer Win32 versions):
340 % TEXEDIT = uedit32 %s/%d/1
341 % WinTeXShell32:
342 % TEXEDIT = texshell.exe /l=%d %s
343 % vi, vim, gvim. here we show Windows gvim.exe:
344 % TEXEDIT = gvim.exe %s +%d
345 % PFE:
346 % TEXEDIT=pfe32/g%d %s
347 % MED:
348 % TEXEDIT=med.exe "%s" %d

```

```

349 % TSE:
350 % TEXEDIT=e32.exe "%s" -n%d
351 % Epsilon (Lugaru)          http://www.lugaru.com/
352 % TEXEDIT="c:\Program Files\eps90\bin\e32.exe" +%d %s
353
354 % For unix
355 %
356 % vi, vim, NEdit, (X)Emacs, pico, jed
357 % TEXEDIT = vi      +%d %s
358 % TEXEDIT = vim     +%d %s
359 % TEXEDIT = nedit   +%d %s
360 % TEXEDIT = xemacs  +%d %s
361
362 %(x)fte:
363 % TEXEDIT = xfte    -l%d %s
364
365
366 %-----
367 % Write .log/.dvi/etc. files here, if the current directory is unwritable.
368 % TEXMFOUTPUT = /tmp
369
370 % If a dynamic file creation fails, log the command to this file, in
371 % either the current directory or TEXMFOUTPUT. Set to the
372 % empty string or 0 to avoid logging.
373 MISSFONT_LOG = missfont.log
374
375 % Set to a colon-separated list of words specifying warnings to suppress.
376 % To suppress everything, use TEX_HUSH = all; this is equivalent to
377 % TEX_HUSH = checksum:lostchar:readable:special
378 TEX_HUSH = none
379
380 % Enable system commands via \write18{...}?
381 shell_escape = f
382
383 % Allow TeX \openout/\openin on filenames starting with '.' (e.g., .rhosts)?
384 % a (any)      : any file can be opened.
385 % r (restricted) : disallow opening "dotfiles".
386 % p (paranoid)  : as 'r' and disallow going to parent directories, and
387 %               restrict absolute paths to be under $TEXMFOUTPUT.
388 openout_any = p
389 openin_any = a
390 % Allow TeX, MF, and MP to parse the first line of an input file for
391 % the %&format construct.
392 parse_first_line = t
393
394 % Enable the mktex... scripts by default? These must be set to 0 or 1.
395 % Particular programs can and do override these settings, for example
396 % dvips's -M option. Your first chance to specify whether the scripts
397 % are invoked by default is at configure time.
398 %

```

```

399 % These values are ignored if the script names are changed; e.g., if you
400 % set DVIPSMMAKEPK to 'foo', what counts is the value of the environment
401 % variable/config value 'FOO', not the 'MKTEXPK' value.
402 %
403 % MKTEXTEX = 0
404 % MKTEXPK = 0
405 % MKTEXMF = 0
406 % MKTEXTFM = 0
407 % MKOCP = 0
408 % MKOFM = 0
409
410 % What MetaPost runs to make MPX files. This is passed an option -troff
411 % if MP is in troff mode. Set to '0' to disable this feature.
412 MPXCOMMAND = makempx
413
414
415 % Part 3: Array and other sizes for TeX (and Metafont and MetaPost).
416 %
417 % If you want to change some of these sizes only for a certain TeX
418 % variant, the usual dot notation works, e.g.,
419 % main_memory.hugetex = 20000000
420 %
421 % If a change here appears to be ignored, try redumping the format file.
422
423 % Memory. Must be less than 8,000,000 total.
424 %
425 % main_memory is relevant only to initex, extra_mem_* only to non-ini.
426 % Thus, have to redump the .fmt file after changing main_memory; to add
427 % to existing fmt files, increase extra_mem_*. (To get an idea of how
428 % much, try \tracingstats=2 in your TeX source file;
429 % web2c/tests/memtest.tex might also be interesting.)
430 %
431 % To increase space for boxes (as might be needed by, e.g., PiCTeX),
432 % increase extra_mem_bot.
433 %
434 % For some xy-pic samples, you may need as much as 700000 words of memory.
435 % For the vast majority of documents, 60000 or less will do.
436 %
437 main_memory = 263000 % words of inmemory available; also applies to inimf&mp
438 extra_mem_top = 0 % extra high memory for chars, tokens, etc.
439 extra_mem_bot = 0 % extra low memory for boxes, glue, breakpoints, etc.
440
441 % Words of font info for TeX (total size of all TFM files, approximately).
442 font_mem_size = 200000
443
444 % Total number of fonts. Must be >= 50 and <= 2000 (without tex.ch changes).
445 font_max = 1000
446
447 % Extra space for the hash table of control sequences (which allows 10K
448 % names as distributed).

```

```

449 hash_extra = 0
450
451 % Max number of characters in all strings, including all error messages,
452 % help texts, font names, file names, control sequences.
453 % These values apply to TeX and MP.
454 pool_size = 125000
455
456 % Minimum pool space after TeX/MP's own strings; must be at least
457 % 25000 less than pool_size, but doesn't need to be nearly that large.
458 string_vacancies = 25000
459 max_strings = 15000           % max number of strings
460 pool_free = 5000             % min pool space left after loading .fmt
461
462 % Hyphenation trie. As distributed, the maximum is 65535; this should
463 % work unless 'unsigned short' is not supported or is smaller than 16
464 % bits. This value should suffice for UK English, US English, French,
465 % and German (for example). To increase, you must change
466 % 'ssup_trie_opcode' and 'ssup_trie_size' in tex.ch (and rebuild TeX);
467 % the trie will then consume four bytes per entry, instead of two.
468 %
469 % US English, German, and Portuguese: 30000.
470 % German: 14000.
471 % US English: 10000.
472 %
473 trie_size = 64000
474
475 % Buffer size. TeX uses the buffer to contain input lines, but macro
476 % expansion works by writing material into the buffer and reparsing the
477 % line. As a consequence, certain constructs require the buffer to be
478 % very large. As distributed, the size is 50000; most documents can be
479 % handled within a tenth of this size.
480 buf_size = 200000
481
482 % These are Omega-specific.
483 ocp_buf_size = 20000         % character buffers for ocp filters.
484 ocp_stack_size = 10000      % stacks for ocp computations.
485 ocp_list_size = 1000       % control for multiple ocps.
486
487 % These work best if they are the same as the I/O buffer size, but it
488 % doesn't matter much. Must be a multiple of 8.
489 dvi_buf_size = 16384       % TeX
490 gf_buf_size = 16384       % MF
491
492 % It's probably inadvisable to change these. At any rate, we must have:
493 % 45 < error_line < 255;
494 % 30 < half_error_line < error_line - 15;
495 % 60 <= max_print_line;
496 % These apply to Metafont and MetaPost as well.
497 error_line = 79
498 half_error_line = 50

```

```

499 max_print_line = 79
500 stack_size = 300      % simultaneous input sources
501 save_size = 4000     % for saving values outside current group
502 param_size = 500     % simultaneous macro parameters
503 max_in_open = 15     % simultaneous input files and error insertions
504 hyph_size = 1000     % number of hyphenation exceptions, >610 and <32767
505 nest_size = 100      % simultaneous semantic levels (e.g., groups)
506 obj_tab_size = 200000 % PDF objects
507
508
509 main_memory.mpost = 1000000
510
511 main_memory.context = 1500000
512 hash_extra.context = 25000
513 pool_size.context = 750000
514 string_vacancies.context = 45000
515 max_strings.context = 55000
516 pool_free.context = 47500
517 nest_size.context = 500
518 param_size.context = 1500
519 save_size.context = 5000
520 stack_size.context = 1500
521 obj_tab_size.context = 256000
522
523 main_memory.hugetex = 1100000
524 param_size.hugetex = 1500
525 stack_size.hugetex = 1500
526 hash_extra.hugetex = 15000
527 string_vacancies.hugetex = 45000
528 pool_free.hugetex = 47500
529 nest_size.hugetex = 500
530 save_size.hugetex = 5000
531 pool_size.hugetex = 500000
532 max_strings.hugetex = 55000
533
534 main_memory.cslatex = 1100000
535 param_size.cslatex = 1500
536 stack_size.cslatex = 1500
537 hash_extra.cslatex = 15000
538 string_vacancies.cslatex = 45000
539 pool_free.cslatex = 47500
540 nest_size.cslatex = 500
541 save_size.cslatex = 5000
542 pool_size.cslatex = 500000
543 max_strings.cslatex = 55000
544 font_mem_size.cslatex= 400000
545
546 main_memory.hugelatex = 1100000
547 param_size.hugelatex = 1500
548 stack_size.hugelatex = 1500

```



```
549 hash_extra.hugelatex = 15000
550 string_vacancies.hugelatex = 45000
551 pool_free.hugelatex = 47500
552 nest_size.hugelatex = 500
553 save_size.hugelatex = 5000
554 pool_size.hugelatex = 500000
555 max_strings.hugelatex = 55000
556 font_mem_size.hugelatex= 400000
557
558 main_memory.jadetex = 1500000
559 param_size.jadetex = 1500
560 stack_size.jadetex = 1500
561 hash_extra.jadetex = 50000
562 string_vacancies.jadetex = 45000
563 pool_free.jadetex = 47500
564 nest_size.jadetex = 500
565 save_size.jadetex = 5000
566 pool_size.jadetex = 500000
567 max_strings.jadetex = 55000
568
569 main_memory.pdfjadetex = 2500000
570 param_size.pdfjadetex = 1500
571 stack_size.pdfjadetex = 1500
572 hash_extra.pdfjadetex = 50000
573 string_vacancies.pdfjadetex = 45000
574 pool_free.pdfjadetex = 47500
575 nest_size.pdfjadetex = 500
576 save_size.pdfjadetex = 5000
577 pool_size.pdfjadetex = 500000
578 max_strings.pdfjadetex = 55000
579
580 main_memory.xmltex = 1500000
581 param_size.xmltex = 1500
582 stack_size.xmltex = 1500
583 hash_extra.xmltex = 50000
584 string_vacancies.xmltex = 45000
585 pool_free.xmltex = 47500
586 nest_size.xmltex = 500
587 save_size.xmltex = 5000
588 pool_size.xmltex = 500000
589 max_strings.xmltex = 55000
590
591 main_memory.pdfxmltex = 2500000
592 param_size.pdfxmltex = 1500
593 stack_size.pdfxmltex = 1500
594 hash_extra.pdfxmltex = 50000
595 string_vacancies.pdfxmltex = 45000
596 pool_free.pdfxmltex = 47500
597 nest_size.pdfxmltex = 500
598 save_size.pdfxmltex = 5000
```

```
599 pool_size.pdfxmltex = 500000
600 max_strings.pdfxmltex = 55000
601
602 font_mem_size.pdflatex = 210000
603 main_memory.pdflatex = 1500000
604 param_size.pdflatex = 3000
605 stack_size.pdflatex = 3000
606 hash_extra.pdflatex = 15000
607 string_vacancies.pdflatex = 45000
608 pool_free.pdflatex = 47500
609 nest_size.pdflatex = 500
610 pool_size.pdflatex = 500000
611 save_size.pdflatex = 5000
612 max_strings.pdflatex = 55000
613
614 main_memory.pdfelatex = 1500000
615 param_size.pdfelatex = 1500
616 stack_size.pdfelatex = 1500
617 hash_extra.pdfelatex = 15000
618 string_vacancies.pdfelatex = 45000
619 pool_free.pdfelatex = 47500
620 nest_size.pdfelatex = 500
621 pool_size.pdfelatex = 500000
622 save_size.pdfelatex = 5000
623 max_strings.pdfelatex = 55000
624
```