



# OBSAH

Petr Olšák: Jak jsem dělal knihu klikací . . . . .	1
Josef Barák: Iniciály . . . . .	18
Igor Podlubný: 4 $\text{T}_{\text{E}}\text{X}$ — $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ — Slovenčina . . . . .	20
Petr Sojka: Pár poznámek pro $\text{T}_{\text{E}}\text{X}$ ové novice aneb o účelovosti, Occamově břitvě a černobílosti . . . . .	26
Hierarchie $\text{T}_{\text{E}}\text{X}$ istů . . . . .	30
Sebastian Rahtz, Michel Goossens: The $\text{T}_{\text{E}}\text{X}$ Live Guide, version 2 . . .	34
Petr Olšák: Proč nerad používám $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . . . . .	89
Josef Barák: Příprava a tisk jednoduché brožury s případným využitím stylu <i>fancyheadings</i> . . . . .	99
Announcement of the 18th Annual Meeting of the TeX Users Group . . .	102
Jan Pazdziora: Recenze knihy <b><math>\text{T}_{\text{E}}\text{X}</math>book naruby</b> . . . . .	103
Závazná objednávka knihy $\text{T}_{\text{E}}\text{X}$ book naruby . . . . .	106
TUGboat 17 (4) December 1996 . . . . .	107

Zpravodaj Československého sdružení uživatelů  $\text{T}_{\text{E}}\text{X}$ u je vydáván v tištěné podobě a distribuován zdarma členům sdružení. Po uplynutí dvanácti měsíců od tištěného vydání je poskytován v elektronické podobě (PDF) ve veřejně přístupném archívu dostupném přes <http://www.cstug.cz>.

Své příspěvky do Zpravodaje můžete zasílat v elektronické podobě anonymním ftp na <ftp.icpf.cas.cz> do adresáře `/wagner/incoming/`, nejlépe jako jeden archivní soubor (`.zip`, `.arj`, `.tar.gz`). Současně zašlete elektronickou poštou upozornění na <mailto:bulletin@cstug.cz>. Uvedený adresář je pro vás „write/only“. Pokud nemáte přístup na Internet, můžete zaslat příspěvek na disketě na adresu:

Zdeněk Wagner  
Vinohradská 114  
130 00 Praha 3

Disketu formátujte nejlépe pro DOS, formát Macintosh 1.44 MB je též přijatelný. Nezapomeňte přiložit všechny soubory, které dokument načítá (s výjimkou standardních součástí  $\zeta\text{T}_{\text{E}}\text{X}$ u), zejména v případě, kdy vás nelze kontaktovat e-mailem.

V tomto článku popíšu postupy a makra, pomocí nichž jsem převedl svou knihu *TeXbook naruby* [1] do klikací podoby formátu pdf. Použil jsem modifikaci  $\TeX$ u nazvanou `tex2pdf` od pana Han The Thanh. Proto bude řeč hlavně o tomto programu. Každý uživatel Internetu a čtecího programu Acrobat Reader se může podívat na výsledek mého snažení například na `ftp://math.feld.cvut.cz/pub/olsak/tbn`.

Český jazyk nám umožňuje pořadím slov nepatrně měnit význam řečeného. V názvu tohoto článku jsem použil spojení „knihu klikací“ místo „klikací knihu“. Tím jsem chtěl naznačit, že nejprve jsem vytvořil knihu a potom (jako vedlejší efekt) jsem se už danou knihu snažil převést do klikací podoby. Cílem díla tedy od začátku bylo vytvořit knihu, která má existovat především v *papírové podobě*, pokud možno kvalitně vysázená a svázaná. Vůbec jsem nechtěl dělat příruční nápovědu k  $\TeX$ u do počítače. Celé dílo bylo tedy především podřízeno koncepci formátu knihy, nikoli filosofii elektronického hypertextového dokumentu. Kdybych měl dělat elektronickou nápovědu pro  $\TeX$ , udělal bych ji zcela jinak. Jistě mnozí čtenáři vědí, že jsem s počítačovou myší zcela nesrostl, ba právě naopak. Takže asi bych elektronickou nápovědu nedělal vůbec.

Zdrojový text knihy jsem napsal s hojným počtem symbolických odkazů typu `\ref{značka}` podstatně dříve, než jsem začal vůbec koketovat s možností pořízení hypertextové verze. Jakým konkrétním dialektem byl text dokumentu napsán se může každý podívat – v elektronické distribuci knihy je ukázkový soubor `tabulky.tex` obsahující jednu kapitolu knihy. Při zavedení hypertextových odkazů tedy nebylo potřeba dělat vůbec žádnou změnu ve zdrojovém textu. Pozměnil jsem pouze pár maker pro formátování knihy a výsledkem byl klikací dokument, který obsahuje přes 10 tisíc hypertextových odkazů.

## O programu `tex2pdf`

V [3] zveřejnil autor programu `tex2pdf` své dílo. Rozhodl jsem se tuto práci použít. Popíšu nejprve, jak vypadala instalace<sup>1</sup>. Balík `tex2pdf` je veřejně přístupný na `ftp://ftp.muni.cz/pub/tex/local/cstug/thanh/tex2pdf` a je snadno instalovatelný na UNIXových systémech, které používají  $\TeX$  podle

<sup>1</sup> Přiznám se, že mě nebaví číst recenze typu, jak jsem „To“ vybalil z krabice, jak jsem „To“ instaloval, jak jsem „To“ rozchodil či nerozchodil apod. V tomto případě mi ale připadá popis instalace poměrně důležitý.

Berryho instalace `web2c`. V mém případě šlo o SUN OS 4.1.3 a `web2c` jsem tam měl, takže půda byla připravena. Stačilo postupovat podle návodu v `README`. Tam se například můžeme dočíst, že je nejprve potřeba překontrolovat, zda je přítomný `web2c/tex` společně s knihovnou `kpathsea-2.6`. To bohužel není zcela samozřejmé, protože Karl Berry nechal `web2c/tex` už dva roky netknutý, se starou knihovnou `kpathsea-1.8`, která obsahuje chyby a se kterou nejde `tex2pdf` přeložit. Je tedy nejprve nutné postupovat podle záplatovacího souboru `web2c.kpathsea-2.6.help`, který je přítomen v archívech CTANu v místě vlastního `web2c TEXu2`.

Pokud používáme originální Knuthův T<sub>E</sub>X a nechceme o něj přijít, pak překontrolujeme, zda máme v `/usr/local/bin` kopie binárních souborů `virtex` a `initex` a nikoli jen jejich linky ukazující do `web2c/tex`. V adresáři `web2c/tex` totiž vzniknou nové spustitelné programy `virtex` a `initex` (zahrnující vlastnosti výstupu do pdf) a původní programy nenávratně zmizí.

Před kompilací `tex2pdf` je rovněž nutné nejprve přeložit knihovny `zlib` a `libpng`. Příznám se, že s tou druhou jsem měl jisté problémy, protože tam někdo nechal v `make` souboru místo povelu `ranlib` povel `echo`.

Konečně jsem pozměnil změnový soubor `tex1.ch` tak, aby výsledný T<sub>E</sub>X zapisoval do logu a dalších pracovních souborů česky a nikoli `^~e8esky`. Na odpovídající místo (podle pořadí sekce) jsem proto přidal<sup>3</sup>:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% [4.49] Make characters 128--255 printable
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
@x
(k<" ")or(k>"~")
@y
(k<" ")or(k=invalid_code)
@z
```

Vyhledávací místa (cesty) jsem nechal v původním stavu, tj. programy si najdou co potřebují v košatém stromu adresářů v `/usr/local/lib/texmf` podle dohodnutého schématu. Komu se to nehodí, přečte si dokumentaci ke `kpathsea` a upraví soubor `texmf.cnf`.

Nyní už stačí spustit podle návodu `install.sh`. Nemusíme se bát – nic se neinstaluje. Název tedy neberme vážně. V adresáři `web2c/tex` se provede změna změnových souborů a kompilují se tam nové programy `virtex` a `initex`. Vlastní instalace typu „napišu `install` a pak program používám“ není zatím v této verzi

---

<sup>2</sup>Pozn. recenzenta: Knihovna dosáhla v posledních letech rychlého rozvoje a je standardem v dnešním `web2c` ve verzi 3.0 a tuto verzi standardně používá i poslední verze `tex2pdf` a binárky na CD-ROM T<sub>E</sub>X Live (viz článek v tomto čísle).

<sup>3</sup>Pozn. recenzenta: Uvedená úprava je standardní součástí záplaty dostupné na `ftp://ftp.cstug.cz/pub/tex/local/unixtexpatch`; stačí tedy aplikovat tuto záplatu.

distribuce zahrnuta. Možná je to dobře, protože každý se může sám rozhodnout, jak zařídí koexistenci nových programů s původními originálními Knuthovými. Já jsem například instalaci provedl takto:

```
# cd /usr/local/bin
# mv virtex virtex-knuth
# mv initex initex-knuth
# cd /usr/local/lib/texmf/ini
# mv tex.pool tex.pool-knuth
# mv csplain.fmt csplain.fmt-knuth
# cd /usr/src/kpathsea-2.6/web2c/tex
# cp virtex initex /usr/local/bin
# cp tex.pool /usr/local/lib/texmf/ini
# cd /usr/local/lib/texmf/ini
# initex csplain.ini
```

Vidíte, že jsem si nejprve udělal zálohy `*-knuth`, ke kterým se po dokončení práce s formátem pdf vrátím<sup>4</sup>. Asi lepší řešení by bylo pozměnit názvy formátů (například `csplain-pdf.fmt`) a volat při požadavku na pdf výstup místo `csplain csplain-pdf`. Protože ale pdf varianta T<sub>E</sub>Xu se od původní liší jen přidáním některých primitivů a bez dodatečných zásahů generuje běžné dvi, dá se nějakou dobu celkem dobře existovat i s tímto pozměněným T<sub>E</sub>Xem.

V distribuci `tex2pdf` je názorná ukázka `example.tex`, ze které je celkem přehledně vidět použití jednotlivých nových primitivů. Když jsem si pak výsledek mohl vyzkoušet v Acrobat Readeru a viděl jsem, co to dělá, získal jsem první představu, jak formát pdf a implementace `tex2pdf` fungují. Přiznám se, že o pdf nemám žádné podrobnější informace (nevlastním žádnou dokumentaci), takže ono `example.tex` bylo pro mě doslova darem.

Při pokusu zpracovat ukázkou `example.tex` celou (včetně obrázku na konci) se mi současná verze programu zborčila. To je ovšem možné přisoudit některým chybám v systémových knihovnách nebo v knihovnách funkcí na zpracování obrázků. Můj záměr udělat knihu klikací to neovlivnilo, protože nemám v knize jediný obrázek. Ptáte se, proč tam nejsou obrázky? Protože kniha je manuálem k T<sub>E</sub>Xu a nikoli obrázkovou publikací pro děti.

## Použité fonty v knize

V knize jsem použil  $\mathcal{G}$ -fonty, protože text knihy je hlavně o plainu a csplainu. Druhý zmíněný formát tyto fonty implicitně používá. Zdálo se mi neférové použít jiné fonty než o kterých mluvím v ukázkách.

---

<sup>4</sup>Pozn. recenzenta: Vracet se není třeba, neboť nová verze obsahuje „starý“ T<sub>E</sub>X jako podmnožinu a nahrazuje tedy původní.

Formát pdf ovšem nepodporuje fonty ve formátu METAFONTu. Podporuje pouze fonty Type1 nebo TrueType. Poslední jmenovaný formát jsem okamžitě zavrhl a začal pátrat po řešení vedoucí k Type1 formátu fontů. Bohužel,  $\mathcal{C}$ -fonty v tomto formátu zatím nemáme. Přesto se nakonec ukázalo, že existuje řešení prostinké jak koloběžka. Na Síti jsou totiž volně dostupné tzv. BaKoMa Type1 fonty zahrnující jednak všechny Computer Modern fonty a jednak všechny DC fonty. Přitom `tex2pdf` pracuje při závěrečném přibalení fontů do výstupu se souborem `texpdf.map`, který má velmi podobné vlastnosti, jako soubor `psfonts.map`, jenž důvěrně známe z `dvips`. Například jsou zde informace o tom, jak naložit s fontem `cmmi10`:

```
cmmi10 cmmi10 4 <cmmi10.pfb CMmathit.enc
```

První slovo označuje název fontu, který je použit přímo v  $\TeX$ u nebo na který odkazuje použitý virtuální font. Program `tex2pdf` totiž umí (při vytváření pdf výstupu) číst i virtuální fonty a provádí expanzi virtuálních skriptů až na úroveň „skutečně použitého“ fontu. Jeho název pak hledá v `texpdf.map`. Druhé slovo je názvem fontu ve světě PostScriptu. V případě BaKoMa fontů má `cmmi10.pfb` skutečně název `cmmi10`, takže toto slovo je shodné s prvním slovem. Za jiných okolností by tam bylo třeba napsáno `Palatino-BoldItalic` nebo něco podobného. Pak následuje tajemná konstanta 4, jejíž význam jsem příliš nepochopil. Souvisí to nějak s různými variantami formátů fontů v pdf. Pokorně tu čtverku obkresluji a funguje to.

Za menšítkem je vlastní název souboru s fontem (`*.pfb`). Poslední slovo označuje název souboru, který popisuje kódování použitého fontu z pohledu  $\TeX$ u v terminologii PostScriptu. Jedná se o známý kódovací vektor, který se používá i v `dvips`. Pokud je například na šesté pozici v tomto vektoru napsáno `/Sigma` a  $\TeX$  sází fontem `cmmi10` šestou pozici, algoritmus vybere z fontu `cmmi10.pfb` znak `/Sigma` bez závislosti na tom, na jaké pozici je tento znak v PostScriptovém fontu zrovna implementován.

Dopíšme do `texpdf.map` například řádky:

```
csr10 dcr10 4 <dcr10.pfb il2.enc  
cstt10 dctt10 4 <dctt10.pfb it2.enc
```

Tím jsme programu `tex2pdf` prozradili, odkud má brát jednotlivé znaky pro realizaci fontu `csr10` a `cstt10`. Kódovací vektory  $\mathcal{C}$ -fontů (soubory `il2.enc` a `it2.enc`) jsem převzal z distribuce balíku `a2ac` skoro beze změny. Přesněji, provedl jsem jen dvě úpravy: 1. V souboru `it2.enc` jsem zaměnil `/.notdef` na pozici 32 za `/visibleSpace`, abych nepřišel o možnost tisku vaničky pro vyznačení mezery ( $\sqcup$ ). 2. Slova `/dcaron` a `/tcaron`, pro jejichž zavedení jsem měl dobré důvody (viz [2]), nejsou respektována v použitých fontech BaKoMa. Proto jsem je byl nucen nahradit slovy `/dquoteright` a `/tquoteright`.

Vidíme, že není potřeba použít žádné virtuální fonty. Za tuto jednoduchost

ovšem platíme. Přišli jsme o znak promile, škrťátko pro polské l a hlavně o skupinu jedenácti znaků ze začátku kódové tabulky CM fontů, obsahující velká písmena řecké abecedy ( $\Gamma$ ,  $\Delta$ , atd.). Poslední problém lze vyřešit dodatečným zavedením fontu pro matematiku:

```
\font\cmtenrm=cmr10 \textfont0=\cmtenrm
\font\cmsevenrm=cmr7 \scriptfont0=\cmsevenrm
\font\cmfiverm=cmr5 \scriptscriptfont0=\cmfiverm
```

Font `\tenrm` zůstává zaveden jako `csr10`, takže v textovém režimu nepřícházíme o češtinu a v matematickém jsme ji nikdy neměli. Zůstává nám tedy nevyřešeno promile a škrťátko pro polské l. Nad tímto detailem jsem mávl rukou a vytvořil pdf formát knihy s vědomím, že u hesla `\promile` a `\l` v části B knihy budou na určitých místech bílá místa. Kdyby to náhodou někomu nedalo spát, je potřeba zdůraznit, že technické prostředky na řešení zmíněných dvou znaků existují. Pomocí virtuálního fontu sestavíme znak promile ze znaku procento a maličké nuly, která v DC fontech je. Pro škrťátko se odkážeme ve virtuálním skriptu například do fontu `cmr10`, takže se škrťátko vykreslí z fontu `cmr10.pfb`. Já osobně ale považuji pdf formát jen za klikací atraktivitu, a ne za solidní formát určený k profesionálnímu tisku<sup>5</sup>. Proto také oželím ztrátu kvality kresby některých akcentů způsobenou použitím kreseb DC fontů místo původních  $\zeta$ -fontů.

V souboru `texpdf.map` jsou již všechny fonty `cm*` zaneseny tak, že se čtou soubory `cm*.pfb` a odpovídající kódovací vektory. Bohužel, soubory s těmito kódovacími vektory (například `CMmathit.enc`) přibaleny nejsou. Povedlo se mi sice na CTANu soubory stejných názvů najít, ale tyto soubory používají na hojných místech zcela jiné názvy pro jednotlivé znaky, než používají BaKoMa fonty. Výsledkem potom je, že v matematických vzorcích máme až příliš mnoho nevytištěných znaků. Nezbylo mi tedy nic jiného než vzít odpovídající BaKoMa fonty ve formátu `afm`, prohlédnout, jaké jsou tam použity názvy znaků, a v kódovacích vektorech to opravit. V původním `texpdf.map` také nebylo rozlišeno kódování pro font `cmr10` a třeba `cmtt10`. V obou případech bylo použito `CMtext.enc`. Pak ale nevytisknete backslash z `cmtt10` ani kroucené závorky. Provedl jsem některé opravy a domluvím se s panem Thanhem, aby tuto „fontovou podporu pro BaKoMa fonty“ zařadil do své distribuce programu `tex2pdf`.

Na závěr tohoto odstavce se zmíním ještě o dalším problému. V knize jsem použil font `logo10` pro logo METAFONT a dále balík fontů `bbold` pro ukázkou zařazení nového matematického fontu do  $\TeX$ u. Font `logo10` není do BaKoMa fontů zahrnut a z balíku `paradissa` mi tento font nějak nefungoval. Acrobat Reader ho odmítal zobrazit. Fonty `bbold` v Type1 formátu pravděpodobně

---

<sup>5</sup>Pozn. recenzenta: Názor Adobe je přesně opačný a snahou firmy je zmenšit velikost souborů přenášených k osvitové jednotce přechodem od formátu PostScript k PDF.

vůbec neexistují. Podruhé jsem mávl nad pdf formátem rukou a napsal jsem do zdrojového souboru `tbn.tex` náhradní a velmi nouzové řešení:

```
\let\mflogo=\rm
\font\bbtext=cmmib10
\let\bbex=\tenex
```

## Původní struktura odkazů v knize

Kniha *TeXbook naruby* má obrovské množství odkazů nejrůznějších typů. Jejich převodem do hypertextové podoby vznikne poměrně pořádná klikací atrakce. Popíšu nejprve strukturu odkazů v takovém stavu, v jakém je použita v tištěné verzi knihy (tj. bez hyperlinků). Kdo má knihu před sebou, může tuto část číst rychleji, protože ví, o čem hovořím.

Obsah knihy je členěn na kapitoly. Každá kapitola má ještě několik sekcí. Takové členění je v části A knihy. Dále v části B knihy jsou dodatky. Některé krátké dodatky nejsou členěny vůbec a rozsáhlejší dodatky (slovník syntaktických pravidel a slovník primitivů a maker) je dále členěn na hesla řazená v abecedním pořadí.

V knize je velké množství ukázek maker ve verbatim prostředí. Tyto ukázky mají číslované řádky a v textu se na některá čísla řádků odkazují. Ukázky jsou při zpracování knihy automaticky extrahovány do sumárního souboru všech ukázek maker `tbn.mac`. V tomto souboru se odkazuje zpětně na strany, na kterých se ukázky v knize vyskytují.

V textu knihy se odkazují na jednotlivé sekce v části A a používám hesla, která lze vyhledat v části B. Dále se na mnoha místech odkazují na čísla stran. V části B je u každého hesla sumární odkaz na všechna čísla stran a čísla řádků v ukázkách, ve kterých bylo heslo v knize použito. Jakými makry jsem tohoto efektu docílil, popisují v sekci 2.4 knihy.

Konečně v rejstříku odkazujeme zpětně na čísla stran a stejně tak činíme v seznamu všech použitých příkladů v knize. Na seznam literatury odkazujeme v textu běžným číslem (například [1]) a v seznamu literatury je uveden zpětně seznam všech stránek v knize, kde byl titul zmíněn.

Knihu považují svým způsobem též za ukádku, co všechno lze v oblasti křížových referencí naprogramovat makrojazykem  $\text{T}_{\text{E}}\text{X}$ u. Při zpracování knihy se vytvářejí kromě `dvi` (nebo `pdf`) automaticky tři soubory. Již zmíněný soubor všech ukázek maker `tbn.mac`, dále soubor pro pozdější vygenerování obsahu `tbn.toc` a konečně soubor všech podkladů pro křížové reference `tbn.ref`. Tento soubor obsahuje kolem dvanácti tisíc řádků plných údajů o odkazech. Z toho je možné soudit, že různými křížovými referencemi všeho druhu je kniha skutečně naplněna až po okraj.



Teprve při čtvrtém zpracování knihy  $\TeX$ em se všechny křížové reference použijí jak mají. Z důvodu šetření paměti  $\TeX$ u jsou totiž do `tbm.ref` zanášeny jen takové údaje, které skutečně potřebují být někde odkazovány. Probereme si jednotlivé průchody:

1. průchod: Soubor `tbm.ref` neexistuje.  $\TeX$  se proto domnívá, že neexistuje žádné heslo v části B, na které je nutno odkazovat a z něhož je nutno odkazovat zpět na stránky, kde bylo heslo použito. Proto do `tbm.ref` nezapisuje výskyt žádného hesla, ale po přečtení části B tam zapíše informaci o existenci všech hesel.

2. průchod: Ze souboru `tbm.ref` si  $\TeX$  přečte všechna použitá hesla. Nyní pro každé takové heslo zapíše do `tbm.ref` záznam o jeho poloze na stránce a případně na řádku v ukázce. Při tisku části B jsou ale zatím sumární informace s odkazy na stránky s výskyty odpovídajícího hesla prázdné. Proto část B vyjde na méně stránek, než odpovídá definitivnímu rozměru knihy.

3. průchod: Soubor `tbm.ref` už je zcela zaplněn, jen možná pro některá hesla nemá správný údaj o straně. Je to z toho důvodu, že v předchozím průchodu vyšla část B na méně stránek, než je definitivní rozměr knihy. Nyní už bude část B obsahovat plně zpětné odkazy a zaplní správný počet stránek. Tím se vytvoří definitivně `tbm.ref`, kde je vše zaneseno v pořádku.

4. průchod: Načte se správný `tbm.ref` a vytvoří se definitivní sazba knihy. Pomocí UNIXového `diff` lze ověřit, že nově vytvořený `tbm.ref` se od předchozí verze z 3. průchodu neliší.

Pro zajímavost uvedu ještě velikosti jednotlivých elektronických formátů. Zdrojové soubory `*.tex` mají dohromady velikost 1,25 MB. Ty byly napsány lidskou rukou. Ostatní formáty jsou generovány automaticky. Soubor `tbm.dvi` má velikost 1,7 MB, což vzhledem k množství vstupních informací a vzhledem k tomu, že soubor obsahuje údaje o naprosto přesné sazbě textu, není nijak velký objem. Vidíme tedy, že formát `dvi` byl velmi dobře navržen (podrobněji se o něm můžete dozvědět přímo v knize, kde mimo jiné popisují  $\TeX$ ovské formáty `dvi` a `tfm`). PostScriptový výstup, určený pro tisk, má pro srovnání 3,2 MB (fonty tam jsou v rozlišení 600 dpi bitmapové). Konečně formát `pdf` v nekomprimované podobě má přes 9 MB, což je nepoužitelná obluda. Proto jsem volil v `tex2pdf` kladný `\compresslevel` a dosáhl jsem výstupního `pdf` ve velikosti 4,8 MB. Bohužel komprimovaná data v `pdf` umí číst až Acrobat Reader verze 3.0<sup>6</sup>, takže pro uživatele verze 2 je `tbm.pdf` nečitelný.

---

<sup>6</sup>Pozn. recenzenta: Komprimovaná data umí číst Acrobat od prvopočátku, avšak `tex2pdf` používá kompresní metodu, která je podporována až od verze 3.0.

## Přidáváme hyperlinky

Hyperlink se v dokumentu skládá aspoň ze dvou částí: aktivní odkaz a cíl. Kliknutím na plochu aktivního odkazu se zobrazí na obrazovce ta část dokumentu, která obsahuje cíl. Na společný cíl může směřovat více aktivních odkazů. Všechny tyto aktivní odkazy a jejich společný cíl mají stejné přirozené číslo (tzv. číselný klíč), kterým jsou vzájemně propojeny. Toto číslo uživatel nevidí, ale při implementaci maker pro pdf formát nás bude toto číslo samozřejmě zajímat.

Aktivní odkaz je určen plochou obdélníkového tvaru umístěnou někde v dokumentu. Jakmile kurzor myši posuneme nad tuto plochu, změní se tvar kurzoru (například z packy se stane vztyčený prst). Navíc ještě může být toto místo ohraničeno viditelným rámečkem a text v něm (pro přehlednost) volíme obvykle v jiné barvě. O volbu odlišné barvy textu hyperlinku se ovšem musíme postarat sami – to není samozřejmá vlastnost hyperlinku.

Pro vymezení plochy aktivního odkazu jsou v `tex2pdf` použity značkovací primitivy `\pdfannotlink` a `\pdfendlink`. Jejich přítomnost v textu nemá žádný vliv na umístění ostatní sazby nebo velikosti boxů. První značka označuje začátek (levý okraj) aktivní plochy a druhá pravý okraj. Nad účařím bude plocha tak veliká, jako je výška boxu, ve kterém se obě značky vyskytují současně. Analogicky hloubka tohoto boxu určuje přesah aktivní plochy pod účaří. Tyto údaje lze též zadat explicitně pomocí klíčových slov `height` a `depth` u primitivu `\pdfannotlink`. V takovém případě se při výpočtu aktivní plochy ignorují rozměry boxu. Značky počátku a konce sice musí být ve společném boxu (jinak se `tex2pdf` tak prudce rozčílí, že ukončí celou činnost), ale pokud jsou dodatečně algoritmem řádkového zlomu od sebe odtrženy do více boxů, pak jsou značky začátku a konce automaticky přidány do míst roztržení. Tím je umožněno nechat aktivní celý víceřádkový úsek odstavce. To jsem ve své knize ale nikde nepoužil.

Cíl hyperlinku je dán pouze bezrozměrným místem v dokumentu. Rozlišujeme několik kategorií cílů: vertikální, horizontální a jiné. Kategorie se liší pouze v tom, jak se prohlížeč pdf vyrovná s celkovým umístěním strany v místě cíle po aktivaci hyperlinku. Buď bude vidět cíl společně s plnou šířkou strany, nebo bude vidět cíl společně s plnou výškou strany, nebo neví. Přiznám se, že jsem tyto vlastnosti zatím příliš neprokoukl a používám osvědčený `\pdfdestfith`<sup>7</sup>, což je zřejmě cíl, který ponechá velikost stránky na plnou její šířku.

Užitečné je vědět, že pokud existuje aktivní odkaz, který nemá odpovídající cíl, přesto se pdf vytvoří a uživatel Acrobat Readeru pak kliká naprázdno. Pokud zpracováváme pokusně jen část knihy, je tento jev velmi častý. Také je dobré vědět, že hyperlinky hned po prvním průchodu fungují dopředu i zpět, což ve srovnání s generováním tištěných odkazů nemá obdobu. V pdf formátu nevadí ani přítomnost osířelých cílů, které nemají na sebe žádný odkaz.

---

<sup>7</sup>V nové verzi `tex2pdf` je místo `\pdfdestfith` použit primitiv `\pdfdest` s různými parametry `fitph`, `fitpv`, `fitp`, `fitbh` atd.

Nejprve jsem navrhl makra, která tvoří rozhraní mezi makry již použitými v knize a novými primitivy z `tex2pdf`. Pokud nepoužiji výstup do `pdf`, chtěl bych, aby makra pracovala „naprázdno“, tj. třeba takto:

```
\def\beglink#1{ % Začátek odkazu,
                % #1 je identifikátor klíče
\def\endlink{ % Konec odkazu
\def\aimlink#1{ % Cíl odkazu,
                % #1 je identifikátor klíče
\def\pglink#1{#1} % \pglink{28} tiskne číslo 28 jako
                  % aktivní odkaz na stranu 28
```

Použití uvedených čtyř maker jsem zařadil do stávajících maker knihy na hojná místa a překontroloval, zda tyto zásahy nezměnily dosavadní zpracování knihy. Například pro tisk řádku obsahu s názvem sekce jsem pozměnil makro `\subtocline` následovně:

```
% v tbn.toc je třeba:
% \subtocline{1.3}{Token procesor}{19}{token}
\def\subtocline #1#2#3#4{%
  \line{\qqquad #1\quad
        \hbox{\beglink{sec/#4}#2\endlink}%
        \puntiky\hbox to2em{\hfil\pglink{#3}}}\cvack}
```

V obsahu tedy budeme mít aktivní jednak názvy sekcí a jednak čísla stran. Názvy sekcí odkazují přímo na začátek sekce, zatímco čísla stran odkazují na začátek strany, na které sekce začíná.

V makru `\sub`, které je použito v místě začátku sekce, píšou:

```
% použití: \sub Token procesor [token]
\def\sub #1 [#2]{\par\goodbreak
  \advance\secnum by1
  \noindent{\bbf
    \the\chapnum.\the\secnum. #1}\par\nobreak
  \aimlink{sec/#2} % <<< --- toto souvisí
                    % s hypertextovým cílem
  ... zapiš odpovídající informace do tbn.toc a~tbn.ref}
```

Z důvodu přehlednosti jsem si tuto ukázkou lehce zestručnil a nerozepisují, jak se zapisují informace do `tbn.toc` a `tbn.ref`. To není věc, kterou nyní sledujeme.

Nastal čas vytvořit klikací folklór. K tomu stačí předefinovat zmíněná makra například takto:

```
\newcount\numlink \newcount\sumlink % Číslo PDF odkazu
\pdfoutput=1
\def\pdfsetcmykcolor#1{\special{PDF:#1 k}}
```

```

\def\CadetBlue{\pdfsetcmykcolor{0.62 0.57 0.23 0}}
\def\Black{\pdfsetcmykcolor{0 0 0 1}} % Barva
                                     % pro hyperlinky
\def\beglink#1{\convertkey{#1}\CadetBlue
  \pdfannotlink height8pt depth3pt
  attr{/Border [0 0 0]} \numlink \relax}
\def\endlink{\pdfendlink\Black}
\def\aimlink#1{\convertkey{#1}\pdfdestfith
  \numlink \relax}
\def\convertkey#1{{% Konvertuje textový klíč
  % na číslo do \numlink
  \expandafter\ifx \csname #1\endcsname \relax
  \global\advance\sumlink by1
  \expandafter\xdef \csname #1\endcsname
  {\the\sumlink}%
  \global\numlink=\sumlink
  \else \global\numlink=\csname #1\endcsname\relax
  \fi}}

```

Vidíme, že pomocí příslušného `\special` jsme schopni „namíchat jakoukoli barvu“ pro běžný text i pro text aktivních odkazů. Zde byla pro texty odkazů použita barva modrá, soudě podle názvu jde o barvu uniform vojenského námořnictva či letectva. V distribuci `tex2pdf` je soubor `colorpdf.tex`, kde je předdefinováno velké množství různých barviček. Vidíme, že se barva deklaruje obvyklým způsobem po jednotlivých složkách v CMYK barevném plánu.

Parametrem `attr`, který přísluší primitivě `\pdfannotlink`, je možno dát formátu `pdf` vzkaz v jeho řeči o vzhledu aktivního odkazu. Já tu řeč moc neumím, pouze vím, že nenapíše-li se `/Border [0 0 0]`, budou mít aktivní odkazy černé rámečky, které v textu knihy poměrně dost ruší.

Těžiště našeho makra spočívá v pomocném makru `\convertkey`, které konvertuje textový klíč na přirozené číslo. Přitom tato konverze musí být jednoznačná. Například po `\convertkey{sec/token}` obdržíme v registru `\numlink` číslo (řekněme) 457 a příště, až se budeme znovu ptát na „`sec/token`“, dostaneme totéž číslo. Položíme-li dotaz s jiným identifikátorem klíče, vrátí nám makro do `\numlink` jiné číslo než 457. Primitivy implementující hyperlinky (`\pdfannotlink` a `\pdfdestfith`) vyžadují totiž ke své práci numerický klíč a ne mnemotechnické slovo<sup>8</sup>.

Zbývá nám ještě ukázat kód makra `\pglink`, které vsází svůj číselný parametr jako hypertextový odkaz na stránku odpovídajícího čísla.

---

<sup>8</sup>Poznamenejme, že nová verze `tex2pdf`, kterou jsem ale při formátování knihy nepoužil, zahrnuje možnost čtení textových klíčů přímo na úrovni primitiv `\pdfannotlink`, `\pdfdest` a `\pdfoutline`. V takovém případě je zřejmě makro `\convertkey` zbytečné.

```

\def\pglink#1{\hbox{\beglink{pg/#1}#1\endlink}}
\let\orishipout=\shipout
\def\shipout#1#2{\orishipout#1{\aimlink
      {pg/\the\pageno}#2}}

```

Zde jsme sáhli na primitiv `\shipout`, který jsme předefinovali tak, aby vkládal do výstupního seznamu i `\aimlink` s odpovídajícím číslem strany.

Zajímavé je též makro, které rozhoduje, zda vytištěná řídicí sekvence v ukázce maker „zmodrá“ (přesněji stane se aktivním odkazem), nebo zůstane černá. Kód tohoto makra zde nebudu uvádět, protože souvisí s dalšími záležitostmi a celé je to velmi rozsáhlé. Zmíním pouze princip tohoto makra.

Jak již bylo řečeno, při zpracování části B se pro každé heslo uloží do souboru `tbn.ref` informace o výskytu tohoto hesla. Při druhém zpracování si  $\text{\TeX}$  z tohoto souboru přečte všechny názvy použitých hesel a udělá si jasno o tom, které heslo je v části B použito a které ne. Když pak například ve verbatim ukázce zpracovává heslo `\def`, ověří si, že toto heslo je v části B uvedeno, a zpracuje je jako hyperlinkový odkaz. Když ale zpracovává třeba heslo `\memakro`, zjistí, že v části B nic takového není, a ponechá text `\memakro` černý. Výsledkem je strakatá modročerná ukázka, kde vše, co je modré, je primitivem nebo makrem plainu vysvětleným v části B, a vše, co je černé, je jen pomocným nebo nově definovaným makrem. V části B u jednotlivých hesel jsou zpětně odkazy na všechny výskyty hesla v knize a používají se tam další hesla vyhledatelná jinde v části B. Čtenář-uživatel se tedy může proklikat až do bohnické léčebny<sup>9</sup>.

Aby bylo takové zpracování modročerných ukázek možné, je samozřejmě v tomto „verbatim“ režimu backslash aktivní. Aby to všechno fungovalo včetně extrahování ukázek do souboru `tbn.mac`, je to navíc poměrně komplikované. Kdo má zájem se do maker ponořit, může. Soubor maker `tbn.tex`, kterými byla kniha zpracována, je volně k dispozici v rámci elektronické distribuce knihy.

## Přidáváme záložky dokumentu

Formát `pdf` má ještě jednu užitečnou vlastnost, kterou jsem nechtěl nechat stranou. Lze v něm deklarovat tzv. záložky. Jedná se o hypertextové odkazy, které nejsou umístěny v dokumentu, ale na okraji v samostatném rámečku. Tyto odkazy jsou tedy k dispozici pořád bez závislosti na zrovna zobrazené stránce dokumentu a váží se na cíl uvnitř dokumentu. Klikne-li uživatel na text záložky, zobrazí se mu stránka s odpovídajícím cílem.

Odkazy typu záložka lze navíc uspořádat v postranním rámečku do stromové struktury. Každá záložka může mít určité množství potomků (to jsou

---

<sup>9</sup>Pozn. recenzenta: Patříčné URL do Bohnic se v knize nepodařilo najít, jde patrně o omyl autora článku.

zase záložky) a navíc je možné předem definovat, zda implicitně uživatel ono potomstvo záložky vidí (otevřená záložka), nebo se mu objeví až po kliknutí na určitý symbol (zatím zavřená záložka).

Každého okamžitě napadne, že je nanejvýš žádoucí převést do záložek stromovou strukturu obsahu knihy. Tím má uživatel obsah neustále před sebou a nemusí se k němu vracet na konkrétní stránku dokumentu.

Záložky musí být vloženy v pořadí, v jakém se objeví v postranním rámečku a musí obsahovat text záložky a číselný klíč cíle (ono přirozené číslo, o kterém jsme už mluvili v souvislosti s hyperlinky). Navíc u záložky musí být řečeno, kolik má záložka potomků a zda se má zobrazit otevřená nebo zavřená. Protože musíme vložit počet potomků do záložky dříve, než projdeme všechny následující potomky v dokumentu, je zřejmé, že tuto vlastnost lze programovat pomocí aspoň dvou průchodů  $\TeX$ u.

K implementaci stromové struktury záložek obsahu knihy ovšem nepotřebujeme zakládat nový pracovní soubor. V souboru s podklady pro obsah `tbm.toc` totiž už od druhého průchodu můžeme zjistit celou strukturu stromu záložek a pak už jen záložky vkládat. Záložky lze vkládat kdekoli v dokumentu (to nemá žádný vliv na jejich chování). Logicky ale budeme řešit záložku v době, kdy budeme vytvářet její cíl.

Rozhodl jsem se mezi záložky vložit též všechna hesla v části B. Budeme mít tedy čtyři druhy záložek. 1. Odkazující na kapitoly (budou mít potomky). 2. Odkazující na sekce (bez potomků). 3. Odkazující na dodatky (s výjimkou dvou dodatků s hesly bez potomků). 4. Odkazující na hesla v dodatcích (bez potomků). Všechny strukturované záložky budou implicitně zavřeny. Uživatel tedy nejprve uvidí jen názvy kapitol a dodatků. Klikne-li na příslušný symbol u názvu kapitoly, objeví se mu všechny sekce dané kapitoly. Klikne-li na vybranou sekci, může začít číst. Podobně po otevření záložky dodatku se všemi primitivy a makry se uživateli v prostoru záložek všechny tyto primitivy a makra objeví v abecedním pořádku. Je jich kolem tisíce. Pak uživateli stačí jen vybrat to heslo, které ho zajímá.

Nejprve jsem navrhl makro `\outlink`, které tvoří rozhraní mezi ostatními makry pro formátování knihy a použitými primitivy pro hyperlinky. Makro má tři parametry. Prvním parametrem je mnemotechnický klíč shodný s klíčem cíle, druhým parametrem je počet potomků (které budou následovat) a třetím parametrem je vlastní text záložky. Prázdná varianta makra (pro případ výstupu do dvi a nikoli do pdf) vypadá tedy takto:

```
\def\outlink #1#2#3{}
```

Varianta makra v případě výstupu do pdf je následující:

```
\def\outlink #1#2#3{%
  \def\tt{}\def~{ }\def\char‘{\string}\def\TeX##1{TeX}%
  \convertkey{#1}% klíč cíle
```

```
\pdfoutline \numlink count -#2 % záložka je zavřená
  {#3}} % vlastní text záložky
```

Za vysvětlení stojí zřejmě druhý řádek ukázky, kde jsou některé sekvence předefinovány. Do textu záložky se totiž dostane vše, i to, co bychom nechtěli. Například v nadpisu sekce používám zápis:

```
\sub Tabulky pomocí \tt \char'\halign [halign]
```

Protože jsem v makru `\sub` zařídil (zde neuvádím jak), že v okamžiku zápisu do `tbm.toc` mají sekvence `\halign` a `\tt` význam `\relax` a sekvence `\char` je primitivní, zapíše se do `tbm.toc` skutečně text „Tabulky pomocí `\tt \char '\halign`“. V době tisku obsahu se to správně přečte a vytiskne. Ovšem nic takového nechceme zapisovat do záložky. Proto jsou odpovídající sekvence na chvíli předefinovány. Náš příklad se tedy expanduje postupně na „Tabulky pomocí `\string\halign`“ a do textu záložky se zapíše „Tabulky pomocí `\halign`“. Protože je text záložky Acrobat Readerem interpretován jako string s céčkovou konvencí, změní se nakonec dvojité zpětné lomítko na jednoduché.

Zabýváme se nyní výpočtem struktury záložek. Rozhodl jsem se pracovat s těmito řídicími sekvencemi: „`\kap:⟨číslo kapitoly⟩`“ — expanduje na počet sekcí dané kapitoly. Dále „`\app:⟨číslo dodatku⟩`“ — expanduje na počet hesel v daném dodatku.

Nejprve uvedu pomocná makra, která s těmito řídicími sekvencemi pracují:

```
\def\expnumber#1{\expandafter \ifx\curname#1\endcurname
  \relax 0%
  \else \curname#1\endcurname \fi}
\def\advancenumber#1{\tempnum=\expnumber{#1}\relax
  \advance\tempnum by1
  \expandafter\xdef\curname#1\endcurname{\the\tempnum}}
\def\extractkapnum#1.#2:{#1}
```

V souboru `tbm.toc` se vytvoří podklady pro obsah v tomto tvaru:

```
\kaptocline {1}{Vstupní části \TeX {u}}{10}
\subtocline {1.1}{Koncept vstupní brány
  \TeX {u}}{10}{koncept}
\subtocline {1.2}{Input procesor}{12}{input}
\subtocline {1.3}{Token procesor}{19}{token}
\kaptocline {2}{Expand procesor}{31}
\subtocline {2.1}{Definování maker}{31}{def}
...
```

Načtením takto členěného souboru `tbm.toc` získáme informace o počtech sekcí každé kapitoly:

```

\def\subtocline #1#2#3#4{%
  \advancenumbers{\kap:\extractkapnum#1:}%
  \line{... dále sázíme řádek obsahu
        stejně, jako prve ...}}
\kaptocline #1#2#3{\line{... sazba řádku
                        obsahu beze změn ...}}

```

Podobně spočítáme počet hesel v části B při čtení souboru `tbm.ref`. Nyní stačí záložky prostě vložit na požadované místo. Třeba v definici `\kap` a `\sub` upravíme kód následujícím způsobem:

```

\def\kap#1 \par{\vfill\break \linenum=0 \secnum=0
  \global\advance\chapnum by1
  \xdef\chapmark{\the\chapnum. #1}
  \headline={\hfil\starthead}
  {\bigbf \the\chapnum. #1}\par
  \aimlink{kp/\the\kapnum}
  \outlink{kp/\the\kapnum}%
  {\expnumber{\kap:\the\kapnum}}{#1}
  ... a~dále zápis do tbm.toc}
\def\sub #1 [#2] \par{\par\goodbreak
  \advance\secnum by1
  \noindent{\bbf
    \the\kapnum.\the\secnum. #1}\par\nobreak
  \aimlink{sec/#2}
  \outlink{sec/#2}0{#1}
  ... a~dále zápis do tbm.ref a~tbm.toc}

```

Práce se záložkami má svá úskalí. Nelze vybrat font pro text záložek, protože tento font závisí na konfiguraci Acrobat Readeru v použitém operačním systému. Jsou-li tedy texty záložek české (v csplainu a v Unixovém  $\TeX$ u je použito ISO-8859-2), pak se tyto texty správně zobrazí jen tehdy, pokud použitý operační systém pracuje s fonty ve stejném kódování a Acrobat Reader je má nakonfigurovány pro zobrazování textů záložek. Z toho důvodu „Portable Document Format“ není zas tak úplně „Portable“, ale nad tím jsem jen mávl (do třetice) rukou. Mohl jsem sice vytvořit v  $\TeX$ u makra, která by konvertovala názvy kapitol a sekcí na nehackovanou a necarkovanou variantu textu, ale nač si přidělovat zbytečnou práci, že?

Zde vidíme, že změna změnového souboru, zmíněná na začátku tohoto článku, má pro nás velký význam. Kdybychom ji neudělali, dostali bychom v českých textech do záložek `p8edb9ernou zmecbb stf8edb9ek10`.

---

<sup>10</sup>Zde bylo v ISO-8859-2 řečeno: „příšernou změň stříšek“.



## Větvení výpočtu

V předchozích odstavcích jsme naznačili, že makra `\beglink`, `\endlink`, `\aimlink`, `\pglink` a `\outlink` necháme prázdná, pokud zpracováváme knihu do `dvi`. Naopak, nastavíme je na neprázdnou hodnotu, pokud generujeme `pdf`. K tomu se mi osvědčilo velmi prosté větvení:

```
% \let\pdfoutput=\undefined
%% Pokud chceme výstup jen do DVI
\ifx\pdfoutput\undefined
... definice prázdných maker:
    \beglink, \endlink, \aimlink, \pglink, \outlink
\else
\pdfoutput=1
\fontsubset=1
... definice hyperlinkových maker:
    \beglink, \endlink, \aimlink, \pglink, \outlink
... řešení náhradních fontů
    \fam0, \mflogo, \bbtext, \bbex
... zmenšení okrajů elektronického "papíru"
\fi
```

Jestliže nyní budeme zpracovávat knihu na Thanhově  $\TeX$ u, bude se generovat `pdf` soubor. Pokud ji budeme zpracovávat klasickým  $\TeX$ em, dostaneme staré známé `dvi`. Navíc, pokud chceme na Thanhově  $\TeX$ u vytvořit výstupní `dvi`, stačí zrušit počáteční procento na prvním řádku ukázky.

Větvení jsem použil ještě v protitulu knihy. Tam mám v klasické variantě zrcadlový obrázek, zatímco v `pdf` variantě je vysvětlující text. K tomu stačilo psát:

```
%% Protitul
\ifx\pdfoutput\undefined
\input epsf
\vglue 7mm
\centerline{\epsfbox{protitul.eps}}
\else
\vglue 2cm
Toto je text knihy v~\uv{klikací} verzi ve formátu PDF.
Jsou zde použity náhradní fonty ve snížené kvalitě.
Proto nedoporučuji z~tohoto formátu tisknout.
Pro tisk použijte formát \texttt{dvi} nebo
\texttt{ps}, viz též soubor \texttt{tbn-tisk.txt}.
\fi
\vfil\break
```

## Závěrem

Chtěl bych poděkovat panu Thanhovi za jeho velmi dobře udělaný program `tex2pdf`. Hodně si cením i jeho ochoty pomoci mi v době, kdy jsem měl s tímto programem problémy. Pokud jsem v programu objevil nějakou závadu, měl jsem ji během pár hodin opravenou. Takový servis se hned tak nevidí.

Dále bych chtěl poznamenat, že mě ani po hraní s `pdf` formátem v Acrobat Readeru tato filosofie přenosu informací nenadchla a že zůstávám u starého známého tvrzení: *knih je kniha*. S knihou si mohu vlézt do postele nebo najít jinou fyziologicky příjemnou polohu, zvolit osvětlení příjemné pro oči a plně a nerušeně se soustředit na informace v knize obsažené. Sebelepší počítač, sebelepší klikací vylomeniny a sebelepší obrazovka tuto pohodu nenahradí.

## Tabulka použitých primitivů z `tex2pdf` implementace $\TeX$ u

Uvádím jen primitivy ze starší verze `tex2pdf`, které jsem použil. Program `tex2pdf` je stále ve vývoji a syntaxe některých primitivů už doznala změny. Proto doporučuji prostudovat nejprve dokumentaci k `tex2pdf` a `example.tex`.

Já jsem použil starší verzi `tex2pdf` hlavně proto, že jsem v ní měl už text knihy odladěný a nová verze (s bohatším jazykem primitivů a s novými klikacími možnostmi) byla v době závěrečného zpracování knihy teprve ve fázi testování. Až bude otestována, pak mi bude stačit během pár vteřin pozměnit použití primitivů v makrech `\beglink`, `\endlink`, `\aimlink` a `\outlink` a přechod na novou verzi bude proveden.

<code>\pdfoutput</code>	Je registr typu $\langle number \rangle$ . Je-li kladný, bude výstup do <code>pdf</code> . Jinak do <code>dvi</code> . Implicitně 0. Záleží na jeho hodnotě při prvním <code>\shipout</code> v dokumentu.
<code>\pdfpagewidth</code>	Registr typu $\langle dimen \rangle$ . Určuje šířku strany v <code>pdf</code> dokumentu. Není-li specifikován, bude počítán z registru <code>\hsize</code> plus okraj <code>(lin+\hoffset)</code> po obou stranách.
<code>\pdfpageheight</code>	Podobně jako <code>\pdfpagewidth</code> .
<code>\compresslevel</code>	Registr typu $\langle number \rangle$ . Hodnota komprese textu ve výstupním <code>pdf</code> . Implicitně 0 (žádná komprese), maximálně 9 (největší komprese). Pozor, je-li <code>\compresslevel &gt; 0</code> , je výstup čitelný jen v Acrobat Readeru verze $\geq 3.0$ .
<code>\fontsubset</code>	Registr typu $\langle number \rangle$ . Je-li kladný, budou do <code>pdf</code> zaváděny jen ty znaky fontů, které jsou použity. Nikoli tedy celé fonty.
<code>\pdfannottext width<math>\langle dimen \rangle</math> height<math>\langle dimen \rangle</math> [open] <math>\{ \langle balanced text \rangle \}</math></code>	Vloží do dokumentu značku. Klikne-li uživatel na tuto značku, otevře se text uvedený (po expanzi) v $\langle balanced text \rangle$ . Je-li uvedeno nepovinné klíčové slovo <code>open</code> , inicializuje se

značka jako otevřená.

`\pdfdestfith`  $\langle number \rangle$

Cíl hyperlinku nebo záložky. Parametr  $\langle number \rangle$  je klíč cíle.

`\pdfannotlink` [`height` $\langle dimen \rangle$ ] [`depth` $\langle dimen \rangle$ ] [`attr`{ $\langle balanced text \rangle$ }]  
 $\langle number \rangle$

Zahajovací značka odkazu hyperlinku. Parametr  $\langle number \rangle$  je klíč shodný s klíčem cíle. Nepovinný  $\langle balanced text \rangle$  udává parametry značky a `height` a `depth` její rozměry.

`\pdfendlink` Ukončující značka hyperlinku. Musí následovat ve stejném boxu, jako zahajující `\pdfannotlink`.

`\pdfoutline`  $\langle number \rangle$  [`count` $\langle number \rangle$ ] { $\langle balanced text \rangle$ }

Záložka. Povinný parametr  $\langle number \rangle$  je klíč odpovídající klíči cíle a  $\langle balanced text \rangle$  je (po expanzi) text záložky. Nepovinný `count` $\langle number \rangle$  specifikuje počet potomků záložky, které budou za touto záložkou následovat. Implicitně 0.

`\pdfinfo` { $\langle balanced text \rangle$ }

Vloží do výstupního souboru údaje uložené v  $\langle balanced text \rangle$ . Jedná se o informace speciálních typů `/Author`, `/Title`, `/Subject`, `/Keywords`, `/CreationDate`, `/ModDate`, `/Creator` a `/Producer`, které se zobrazí při vyvolání „Document info“ ve čtecím programu.

`\special` {PDF: $\langle balanced text \rangle$ }

To je samozřejmě starý známý primitiv, ovšem za zmínku stojí, že při výstupu do pdf je `tex2pdf` schopen interpretovat jediné `\special` začínající na text „PDF:“. Jaké vzkazy se dají do výstupu prostřednictvím  $\langle balanced text \rangle$  vkládat vyplývá zřejmě z dokumentace k pdf. Pro práci s obrázky ve formátu png je v `tex2pdf` implementován primitiv `\pdfimage`, který jsem ovšem v knize nepoužil.

## Literatura

- [1] Petr Olšák: *TEXbook naruby*, Praha 1996. Plný text knihy je k dispozici v elektronické podobě na <ftp://math.feld.cvut.cz/pub/olsak/tbn>. 468 stran.
- [2] Petr Olšák: Program `a2ac` – manipulace s fontem na úrovni PostScriptu. Zpravodaj Československého sdružení uživatelů `TEXu`, **6** (1), 23–30 (1996).
- [3] Han The Thanh: Alternativní výstup programu `TEX` – PDF. Zpravodaj Československého sdružení uživatelů `TEXu`, **6** (2), 69–85 (1996).

**N**ři vytváření svých textů v  $\text{T}_{\text{E}}\text{X}$ u můžeme využívat nepřeberného množství fontů různých typů a velikostí. Ve svém příspěvku bych se chtěl zmínit o možnosti začínat kapitoly (odstavce) velkými ozdobnými písmeny, tzv. iniciálami. Protože používám  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}^1$ , zaměřím svůj popis právě na tento formát  $\text{T}_{\text{E}}\text{X}$ u.

**N**ejprve je třeba si opatřit patřičný font. Velmi pěkný font, používaný i v tomto článku, nalezneme na CTANu<sup>2</sup>. Je dílem Yannise Haralambouse. V adresáři YINIT nalezneme několik malých souborů, většinou s příponou *mf*. V souborech *yinita.mf* až *yinitz.mf* jsou vlastní definice každého znaku. Dále se zde nachází soubor *yintbase.mf* a pro nás nejdůležitější *yinitas.mf*. Právě tento soubor je totiž nutné zpracovat METAFONTEM.

**S**romě souborů *.mf* se v adresáři YINIT vyskytuje soubor *initials.sty* od Andrease Schrella, který má napomoci snadnějšímu použití iniciál ve vlastním textu. Použití tohoto stylu je následující:

- soubor *initials.sty* umístíme do vhodného adresáře s dalšími styly (já osobně jsem jej umístil do `emtex\textinput\latex2e`)
- před `\begin{document}` uvedeme:

```
\usepackage{initials} (možno kombinovat s dalšími styly)
\font\initialfont=yinitas
```

Nyní již můžeme využít nových příkazů `\goodlineskip` a `\initial`. Příkaz `\goodlineskip` slouží pro nastavení správného řádkování (vzhledem k iniciálám); stačí jej uvést pouze jednou, za `\begin{document}`. Příkaz `\initial` slouží k umístění a zobrazení vlastní iniciály (např. `\initial{A}`). Vzhledem k tomu, že iniciála zahajuje určitou partii textu, nemá příkaz jako například `\initial{a}` smysl, a tedy není podporován (ani ve vlastním fontu nejsou definovány minusky, tj. malá písmena abecedy). Další text může bezprostředně navazovat za příkaz `\initial`, takže lze napsat např. `\initial{B}ylo nebylo...`

---

<sup>1</sup> $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2\text{e}$  ze standardní instalace  $\zeta\text{TUGu}$

<sup>2</sup>`ftp://ftp.ctug.cz/pub/tex/CTAN/fonts/gothic/yinit`

**N** delšího textu, než je ten uvedený výše, by měl ostatní text kolem iniciály obtékat. Zde jsem však narazil na problém. Druhý a třetí řádek textu částečně do iniciály zasahovaly, a to z důvodu jejího nesprávného umístění. Tento problém jsem po prostudování stylu (programovat v  $\TeX$ u příliš neumím) vyřešil drobnou úpravou. V definici příkazu `\initial` jsem přidal příkaz `\parindent`<sup>3</sup>, takže v upraveném stylu vypadá zmíněná definice takto:

...

```
\def\initial#1{\parindent=0pt\par
\setbox\yinibox\hbox{H}%
\setbox\yinibox\vbox to \ht\yinibox{\offinterlineskip
\hbox{\font #1\ } \vss}%
\initiallinesplus\initiallines
\advance\initiallinesplus by 2
\vskip\initiallinesplus\baselineskip\penalty0
\vskip-\initiallinesplus\baselineskip
\leavevmode\hskip-\wd\yinibox
\copy\yinibox\hangindent=\wd\yinibox
\hangafter=-\initiallines
\ignorespaces}
```

...

Po úpravě se iniciály chovají již tak jak mají, což je možné posoudit i z tohoto článku.

*Josef Barák*  
*xbarak00@kpve.fee.vutbr.cz*

---

<sup>3</sup>Změna hodnoty `\parindent` není rozumným řešením, neboť odstavce pak nebudou odsazeny. Iniciály budou umístěny správně, jestliže se v makru `\initial` místo `\leavevmode` použije `\noindent`. Pozn. red.

Balík 4 $\text{T}_{\text{E}}\text{X}$  (4all $\text{T}_{\text{E}}\text{X}$  na dvoch CD, vydaných v roku 1995), ktorý sa distribuuje aj prostredníctvom  $\zeta\text{TUGu}$ , je skutočne vynikajúcou sadou prostriedkov, ktoré môže potrebovať bežný užívateľ  $\text{T}_{\text{E}}\text{X}$ u. Z hľadiska autorov technických textov sú zvlášť cennými pohodlné a užitočné prostriedky pre vkladanie grafiky (vrátane Postscriptu), pre tlač na rôznych tlačiarniach a pre prácu s bibliografickými databázami. Okrem toho tam sú aj rozmanité štýlové súbory pre rôzne vydavateľstvá vedeckej literatúry a odborných časopisov, celý rad často používaných utilít a mnoho ďalších dôležitých doplnkov — jedným slovom, je to naozaj perfektný balík typu *all-in-one*. Z priestorových dôvodov je najvýhodnejšie používať 4 $\text{T}_{\text{E}}\text{X}$  priamo z CD. Pri súčasných rýchlych CD-mechanikách nemá veľký význam inštalovať celý 4 $\text{T}_{\text{E}}\text{X}$  na pevný disk. Na pevnom disku stačí mať iba to, čo k 4 $\text{T}_{\text{E}}\text{X}$ u pridávame (nové formáty, fonty, nové štýlové súbory atď.).

Jednou z najväčších predností 4 $\text{T}_{\text{E}}\text{X}$  na CD je nepochybne rozsiahla podpora pre prípravu textov v rôznych jazykoch, tak bežných, ako aj exotických, a to prostredníctvom systému BABEL a vopred pripravených formátov. Pre prípravu viacjazyčných textov musia tieto formáty obsahovať vzory pre delenie slov vo zvolených jazykoch. Žiaľ, 4 $\text{T}_{\text{E}}\text{X}$  na CD neobsahuje žiadny formát  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u, umožňujúci písať po slovensky.

V bežnej praxi však mnohí užívatelia pracujú s  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ om práve v slovenčine, a navyše sú zvyknutí na prostredie  $\zeta\text{T}_{\text{E}}\text{X}$ u vzoru 1993 a 1994. To znamená, že používajú editor  $\zeta\text{S}$ ED pre prípravu textov so slovenskou diakritikou, korektor pravopisu  $\text{TEXSPELL}$  pre kontrolu slovenského pravopisu, a tiež program  $\text{VLNKA}$  pre odstránenie jednopísmenkových slov na konci riadkov.

Takto sa dostávame k myšlienke „poslovenčiť“ 4 $\text{T}_{\text{E}}\text{X}$  na CD.

## Problémy pri generovaní $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ formátov pre slovenčinu

Ukázalo sa, že najväčším problémom je generovanie formátov  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u, obsahujúcich vzory pre delenie slov v slovenčine (na CD sú pripravené formáty `plain` pre slovenčinu, čo umožňuje písať po slovensky aj v  $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}_{\text{E}}\text{X}$ u — vid' manuál k 4 $\text{T}_{\text{E}}\text{X}$ u). Formáty  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2\text{e}$  a  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  2.09, obsahujúce vzory pre delenie slov v slovenčine, na CD 4 $\text{T}_{\text{E}}\text{X}$  chýbajú, a práve tieto formáty je potrebné vygenerovať.

Na prvý pohľad táto úloha vyzerá ako triviálna, lebo menu 4 $\text{T}_{\text{E}}\text{X}$  obsahuje podmenu **Utilities - Format**, ktoré má uľahčiť generovanie vlastných formátov. Pokusy vygenerovať formáty  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  touto cestou boli neúspešné z dvoch príčin:

1. Ak spúšťame 4 $\text{T}_{\text{E}}\text{X}$  z CD, potom sa výsledný vygenerovaný formát, napríklad pre češtinu+angličtinu, ukladá na CD-ROM — jasné, že neúspešne. Zmeniť implicitne nastavenú cestu pre uloženie vygenerovaného formátu 4 $\text{T}_{\text{E}}\text{X}$  neumožňuje.
2. V zozname voliteľných jazykov (súbor LANGUAGE.TXT) chýba slovenčina.

Preto ostáva jediná možnosť: vygenerovať formát z príkazového riadku tak, ako je to popísané v manuáli k 4 $\text{T}_{\text{E}}\text{X}$  na CD na stranách 93–96. Ale tu šťastného užívateľa 4 $\text{T}_{\text{E}}\text{X}$ u čaká asi jedno z najväčších sklamaní: postup, uvedený v manuáli, bohužiaľ, nevedie k vytvoreniu formátu. Navyše, spôsob začlenenia vzorov pre delenie slov v slovenčine je odlišný od iných jazykov.

## Príprava na generovanie formátov

Ďalej uvádzame postupnosť krokov pre generovanie rôznych formátov  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . Predpokladáme, že minimálna inštalácia 4 $\text{T}_{\text{E}}\text{X}$ u bola uskutočnená do adresára `C:\4TEX` a nachádza sa v sústave podadresárov pod `C:\4TEX\TEXFILES`. Kompakt-disk sa nachádza v CD-ROM mechanike D:

1. Pred začatím generovania je potrebné v každom prípade vytvoriť adresár `C:\4TEX\TEXFILES\FORMATS`, kam budeme ukladať vytvorené súbory s príponami `.FMT` a `.FRM`
2. V súbore `C:\4TEX\TEXFILES\4SYSTEM\TEXUSER.SET` treba upraviť riadok, ktorý sa začína slovom `OWNFORMAT`, nasledujúcim spôsobom:  
`OWNFORMAT=C:\4TEX\TEXFILES\FORMATS`

## $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2\text{e}$ + BABEL

1. V adresári `C:\4TEX\TEXFILES\4SYSTEM` vytvoríme súbor `LANGUAGE.DAT`, obsahujúci dva riadky:

```
english          d:\emtex\formats\ushyphen.tex
slovak           d:\emtex\formats\masthyph.tex
```

(Uvedenie cesty v druhom prípade je nutné; ak nebude uvedená, načíta sa súbor `MASTHYPH.TEX`, ktorý sa nachádza v inom adresári a *liši sa* od toho, ktorý potrebujeme.)

V prípade potreby pridaním ďalších podobných riadkov zabezpečíme zaradenie do výsledného formátu vzorov pre delenie slov aj pre ďalšie jazyky.

2. Spustíme 4 $\text{T}_{\text{E}}\text{X}$  a stlačíme kláves F9 pre dočasný odchod do DOSu (DOS shell). Prejdeme do adresára C:\4 $\text{T}_{\text{E}}\text{X}$ \ $\text{TEXFILES}$ \4SYSTEM, v ktorom máme súbor LANGUAGE.DAT. Zadáme príkaz pre vygenerovanie formátu:

```
D:\EMTEX\COMPILER\TEX386.EXE /i /8 /ml /mt:28000 LATEX.LTX
```

Parameter /i spustí generovanie formátu, parameter /8 umožní 8-bitový vstup, parameter /ml umožní  $\backslash\text{charsubdef}$ , čo sa využíva v MASTHYPH.TEX.

3. Na upozornenie, že zdrojové súbory  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u sú viac ako rok staré, zareagujeme stlačením klávesu ENTER. Potom na výzvu treba postupne zadávať nasledujúce reťazce ukončené stlačením klávesu ENTER:

```
d:/emtex/formats/fonttext.lx
d:/emtex/formats/fontmath.lx
d:/emtex/formats/preload.lx
d:/emtex/formats/kamendef.tex
d:/emtex/formats/hyphenm.slo
```

4. Zadáme príkaz DOSu

```
EXIT
```

a vrátime sa do 4 $\text{T}_{\text{E}}\text{X}$ u. Ukončíme 4 $\text{T}_{\text{E}}\text{X}$ .

5. V adresári C:\4 $\text{T}_{\text{E}}\text{X}$ \ $\text{TEXFILES}$ \4SYSTEM nájdeme vygenerovaný súbor LATEX.FMT, ktorý premiestnime do adresára C:\4 $\text{T}_{\text{E}}\text{X}$ \ $\text{TEXFILES}$ \FORMATS a premenujeme na SLOV2E.FMT

6. V tom istom adresári C:\4 $\text{T}_{\text{E}}\text{X}$ \ $\text{TEXFILES}$ \FORMATS vytvoríme súbor SLOV2E.FRM, obsahujúci nasledujúce riadky:

```
: big SLOVAK LaTeX 2e
COMPILER=big SLOVAK LaTeX 2e + Babel
STYLE=c:\4tex\texfiles\formats\SLOV2E
EMTEXOPT=/mt:28000
```

Týmto je ukončené generovanie formátu  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2\text{e}$ , ktorý akceptuje vstup v kódovaní Kamenických a obsahuje vzory pre delenie slov v slovenčine.

Nastaviť tento formát pre  $\text{T}_{\text{E}}\text{X}$ -ovanie je možné v hlavnom menu 4 $\text{T}_{\text{E}}\text{X}$ u po zvolení položky choose TeX Format.

## $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 2.09 + BABEL + NFSS

1. Vytvoríme súbor LANGUAGE.DAT tak, ako je uvedené vyššie.
2. Spustíme 4 $\text{T}_{\text{E}}\text{X}$  a stlačíme kláves F9 pre dočasný odchod do DOSu (DOS shell). Prejdeme do adresára C:\4 $\text{T}_{\text{E}}\text{X}$ \ $\text{TEXFILES}$ \4SYSTEM, v ktorom máme súbor LANGUAGE.DAT. Zadáme príkaz pre vygenerovanie formátu:  

```
D:\EMTEX\COMPILER\TEX386.EXE /i /8 /ml /mt:28000 LPLAIN.LTX
```
3. Potom na výzvu treba postupne zadávať nasledujúce reťazce ukončené stlačením klávesu ENTER:



```

d:/emtex/formats/lplain.tex
babel.hyp
d:/emtex/formats/kamendef.tex
d:/emtex/formats/hyphenm.slo
d:/emtex/formats/lfonts.nfa
d:/emtex/formats/fontdef.nfa
d:/emtex/formats/preload.nfa
newlfont.sty

```

4. Keď sa objaví T<sub>E</sub>X-ovský prompt (\*), zadáme príkaz  
`\dump`
5. Zadáme príkaz DOSu  
`EXIT`  
a vrátime sa do 4T<sub>E</sub>Xu. Ukončíme 4T<sub>E</sub>X.
6. V adresári C:\4TEX\TEXFILES\4SYSTEM nájdeme vygenerovaný súbor  
LPLAIN.FMT, ktorý premiestnime do adresára C:\4TEX\TEXFILES\FORMATS  
a premenujeme na SLOV209N.FMT
7. V tom istom adresári C:\4TEX\TEXFILES\FORMATS vytvoríme súbor  
SLOV209N.FRM, obsahujúci nasledujúce riadky:

```

: big SLOVAK LaTeX 2.09 + BABEL + NFSS
COMPILER=big SLOVAK LaTeX 2.09 + Babel + NFSS
STYLE=c:\4tex\textfiles\formats\SLOV209N
EMTEXOPT=/mt:28000

```

Týmto je ukončené generovanie formátu L<sup>A</sup>T<sub>E</sub>X 2.09 s NFSS, ktorý akceptuje vstup v kódovaní Kamenických a obsahuje vzory pre delenie slov v slovenčine.

Nastaviť tento formát pre T<sub>E</sub>X-ovanie je možné v hlavnom menu 4T<sub>E</sub>Xu po zvolení položky choose TeX Format.

## Starý L<sup>A</sup>T<sub>E</sub>X 2.09 + BABEL

1. Viď predchádzajúci postup.
2. Viď predchádzajúci postup.
3. Potom na výzvu treba postupne zadávať nasledujúce reťazce ukončené stlačením klávesu ENTER:

```

d:/emtex/formats/lplain.tex
babel.hyp
d:/emtex/formats/kamendef.tex
d:/emtex/formats/hyphenm.slo
d:/emtex/formats/lfonts.old

```

4. Keď sa objaví T<sub>E</sub>X-ovský prompt (\*), zadáme príkaz  
`\dump`

5. Zadáme príkaz DOSu  
EXIT  
a vrátime sa do 4 $\TeX$ u. Ukončíme 4 $\TeX$ .
6. V adresári C:\4 $\TeX$ \TEXFILES\4SYSTEM nájdeme vygenerovaný súbor LPLAIN.FMT, ktorý premiestnime do adresára C:\4 $\TeX$ \TEXFILES\FORMATS a premenujeme na SLOV209.FMT
7. V tom istom adresári C:\4 $\TeX$ \TEXFILES\FORMATS vytvoríme súbor SLOV209.FRM, obsahujúci nasledujúce riadky:
 

```
: big SLOVAK LaTeX 2.09 + BABEL
COMPILER=big SLOVAK LaTeX 2.09 + Babel
STYLE=c:\4tex\texfiles\formats\SLOV209
EMTEXOPT=/mt:28000
```

Týmto je ukončené generovanie starého formátu  $\LaTeX$  2.09, ktorý akceptuje vstup v kódovaní Kamenických a obsahuje vzory pre delenie slov v slovenčine.

Nastaviť tento formát pre  $\TeX$ -ovanie je možné v hlavnom menu 4 $\TeX$ u po zvolení položky choose TeX Format.

## Pripojenie editora $\zeta$ ED

1. Vytvoríme adresár C:\4 $\TeX$ \TEXFILES\CSTUG a nainštalujeme tam  $\zeta$ ED, TEXSPELL a program VLNKA.EXE (napríklad, prekopirowaním všetkých súborov zo zodpovedajúceho podadresára ...\ $\zeta$ ED inštalácie  $\zeta\TeX$ u, programu VLNKA.EXE a dokumentácie k nemu VLNKA.DOC).
2. Spustíme konfiguračný program C:\4 $\TeX$ \TEXFILES\CSTUG\CSINST.EXE a nastavíme kódovanie Kamenických.
3. V súbore C:\4 $\TeX$ \TEXFILES\4SYSTEM\TEXUSER.SET treba upraviť riadky, ktoré sa začínajú slovami EDITOR a EDDIR, nasledujúcim spôsobom:

```
EDITOR=C:\4TEX\TEXFILES\CSTUG\CSED.EXE
EDDIR=
```

Teraz už môžeme v 4 $\TeX$ u používať editor  $\zeta$ ED pre prípravu vstupných textov s diakritikou v kódovaní Kamenických.

## Pripojenie programu VLNKA

1. Spustíme 4 $\TeX$  a stlačíme kláves F5.
2. Zvolíme “Add utilities or group of utilities” a na ďalšiu otázku ešte raz potvrdíme, že inštalujeme utilitu.
3. Ako popis programu uvedieme: Vlnka

4. Keď sa objaví výzva na zadanie parametrov, zadáme:
 

```
AAA.BBB =Kk0oUuVvSsZzAaIi
```
5. Na ďalšiu otázku odpovedáme “No”
6. Vrátime sa do hlavného menu. Ukončíme 4 $\text{T}_{\text{E}}\text{X}$  a nahradíme reťazec AAA.BBB v súbore C:\4 $\text{T}_{\text{E}}\text{X}$ \TEXFILES\USER.UTL reťazcom %FNAME.%MAIN\_EXT  
Tento krok si vyžaduje dočasné odstránenie atribútu *Read-only* upravovaneho súboru.  
Teraz pre “vlnkovanie” v 4 $\text{T}_{\text{E}}\text{X}$ u stačí stlačiť kláves F5 a zvoliť utilitu Vlnka.

## Pripojenie programu TEXSPELL

Pripojenie programu TEXSPELL pre kontrolu pravopisu hlavného súboru (t.j. súboru %FNAME.%MAIN\_EXT) sa uskutoční presne tak, ako pripojenie programu VLNKA. Ako popis programu uvedieme:

TEXSPELL - Slovak

Parametre môžeme zadať nasledujúco:

```
AAA.BBB =S @K ?1 $C:\4TEX\TEXFILES\CSED\POMOCNY.SLV
```

(Parameter @K znamená kódovanie Kamenických; parameter =S znamená slovenčinu)

Ukončíme 4 $\text{T}_{\text{E}}\text{X}$ . V súbore C:\4 $\text{T}_{\text{E}}\text{X}$ \TEXFILES\USER.UTL nezabudneme nahraďiť reťazec AAA.BBB reťazcom %FNAME.%MAIN\_EXT

Pre kontrolu pravopisu súboru *include* je potrebné nahraďiť reťazec AAA.BBB reťazcom %INAME.%INCL\_EXT

V oboch prípadoch je potrebné dočasne odstrániť atribút *Read-only* upravovaného súboru.

## Záver

Hlavnou výhodou popísaného riešenia je to, že bežný užívateľ, ktorý obyčajne používa L $\text{A}}\text{T}_{\text{E}}\text{X}$ , bude môcť prakticky bez zmeny pracovného prostredia využívať obrovské možnosti 4 $\text{T}_{\text{E}}\text{X}$ u na CD.

Okrem toho, uvedená postupnosť krokov môže byť využitá pre pravidelné obnovenie (update) formátu L $\text{A}}\text{T}_{\text{E}}\text{X}2\text{e}$  a aktualizáciu vzorov pre delenie slov.

Ten istý postup sa hodí aj pre češtinu; treba iba použiť parameter =C (namiesto =S) pre TEXSPELL a tiež zodpovedajúce vzory pre delenie slov v češtine.

Je možné pridať aj ďalšie programy a utility, napríklad CSTOCS pre konvertovanie textov obsahujúcich diakritiku. V takom prípade treba postupovať tak, ako pri pridávaní programov TEXSPELL a VLNKA.

Problémy a nepresnosti v balíku 4 $\TeX$ , ktoré boli spomenuté v úvode tohto článku, budú s najväčšou pravdepodobnosťou odstránené v ďalšom vydaní 4 $\TeX$  na CD, ale popísané riešenie umožňuje vyhnúť sa im už dnes.

Igor Podlubný  
Katedra riadenia výrobných procesov  
Fakulta BERG  
Technická univerzita v Košiciach  
PODLBN@ccsun.tuke.sk

---

---

## Pár poznámek pro $\TeX$ ové novice aneb o účelovosti, Occamově břitvě a černobílosti

---

PETR SOJKA

„Do not multiply entities beyond necessity“

William of Ockham, 1285–1349

$\TeX$  je systém složitý. Jeho učicí křivka je pozvolná a obtížná, snad i trnitá. Tento příspěvek by chtěl těm, kteří se v „ $\TeX$ ovské hierarchii“<sup>1</sup> chtějí šplhat výše, ale také těm, kteří chtějí jen dostat své myšlenky na papír či článek do sborníku, usnadnit prvotní volby. Jaký makrobalík se naučit, jakou literaturu použít, jak postupovat maximálně efektivně. Žádný jednoduchý a černobílý návod ale neexistuje, byť to někteří fundamentalisté tvrdí a prezentují (či publikují) svá stanoviska vycházející z jejich kontextu jako jediná správná. Čtenáři zde nebudou sugerována zaručeně správná a jediná možná řešení, spíše doporučení typu pokud (**D:**)..., pak (**O:**)....

### Účel

Jeden z častých dotazů, se kterým se na mě lidé obracejí, je: zaslechl jsem něco o  $\TeX$ u: jaký makrobalík se mám naučit, jakou knihou začít, co si mám instalovat? Na takové se usměju a začnu malý výslech:

- k jakému účelu chce  $\TeX$  používat,
- jaké typy dokumentů bude připravovat a jaká je požadovaná kvalita,
- jak rozsáhlé dokumenty bude sázet,
- zda či jak často programuje,

---

<sup>1</sup>Viz článek na straně 30.

- v jakém prostředí, pracovišti, s jakým zázemím bude pracovat,
- z jakých podkladů bude vycházet,
- zda jsou nějaká další omezení daná prostředím tazatele

Veškerá doporučení prostředků, maker či dokonce vhodnosti  $\text{T}_{\text{E}}\text{X}$ u je třeba odvíjet od účelu, ke kterému má sloužit.

## $\text{T}_{\text{E}}\text{X}$ ano či ne?

Tato otázka se může zdát od předsedy sdružení kacířská, ale je třeba si ji položit na prvním místě. Je opravdu pro požadovanou publikační činnost vhodný? Nevystačí se s HTML? Nebo pro vysokou míru interakce a malý rozsah tiskoviny raději nezvolit sázecí systém WYSIWYG?

**D:**Jde o jednorázové vysázení svatebního oznámení?

**O:**Nevyplatí se o zvládnutí  $\text{T}_{\text{E}}\text{X}$ u uvažovat.

**D:**Patříte mezi ty 2 % populace zvané informatiči?

**O:** $\text{T}_{\text{E}}\text{X}$  se vám bude líbit a možná budete raději chtít programovat v plain- $\text{T}_{\text{E}}\text{X}$ u nebo psát dokumenty v  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u.

**D:**Píšete dokumentaci k rozsáhlému softwarovému produktu, který se vyvíjí a vyžaduje se kvalitně zpracovaná programová dokumentace?

**O:**Ano,  $\text{T}_{\text{E}}\text{X}$  je pro vás to pravé.

**D:**Máte nakladatelský dům produkující desítky podobných titulů s hladkými texty či s pevnou strukturou dokumentů?

**O:** $\text{T}_{\text{E}}\text{X}$  je pro vás ten pravý, následujte pěknou řádku vašich konkurentů, kteří na to již přišli.

**D:**Vyžaduje sazba dokumentu vysokou míru interakce, spolupráce s grafikem na umístování obrázků apod.?

**O:**Použijte raději QuarkXpress či něco podobného, pokud dokumentů se stejným návrhem nebudete dělat desítky či stovky.

**D:**Chcete digitalizovat poloautomaticky a přesadit 30000 stran Ottova slovníku naučného?

**O:**Ano,  $\text{T}_{\text{E}}\text{X}$  je ideální, a abyste nemuseli všechny styly na různorodé entity programovat znova, sáhněte po  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u, ke kterému je přímo použitelné nepřeborné množství specifických stylů sazby.<sup>2</sup>

**D:**Jde primárně jen o sdílení informací přes Internet bez požadavků na formu – typografickou kvalitu?

**O:**Uvažte, zda nevystačíte s editorem HTML.

---

<sup>2</sup>Že to bylo doporučení správné, je možno se přesvědčit na <http://www.aion.cz/otta>

## Makrobalíky

Jakmile se rozhodnete pro  $\TeX$ , dalším krokem je volba balíků maker. Věčné téma diskusí může být o tom, jaké balíky maker používat. Odpověď je jednoduchá: takové, jaké vám vyhovují, s kterými svých požadavků (kvalita, cena, ...) dosáhnete *při svých znalostech, ve svém prostředí* a pro daný publikační projekt nejrychleji a s nejmenšími náklady.

Jiná je situace, když se vším musíte prokousávat sám, oproti té, když máte vedle sebe  $\TeX$ ovského guru. Jinak se pracuje, máte-li po ruce CD-ROM či běžící instalaci se stovkami hotových stylů, jiná máte-li holý  $\TeX$  a musíte začínat od píky.

**D:**Chcete mít vše pod kontrolou a jste programátor každým coulem?

**O:**Programujte nad plain $\TeX$ em.

**D:**Potřebujete poslat článek do konference a typografie vás nezajímá, neboť vydavatel si to přesadí?

**O:**Použijte standardního značení  $\LaTeX$ u, případně stylu a značení dodaného s pokyny pro autory.

**D:**Sázíte sborník konference a 95 % příspěvků došlo v  $\LaTeX$ u?

**O:**Vysaďte ji v  $\LaTeX$ u.

**D:**Chcete mít kromě tištěné verze bez práce též verzi HTML dokumentu?

**O:**Použijte  $\LaTeX$  a konvertor  $\LaTeX2HTML$ .

**D:**Sázíte extravagantní dokument vyžadující specifickou  $\backslash$ output rutinu a zcela jistě nikde nepublikovaný styl sazby?

**O:**Napište si ji v plainu.

**D:**Ortodoxní plainista vám vyčítá nezvyklý design  $\LaTeX$ ovského dokumentu a špatné fonty?

**O:**Změňte řádek v hlavičce dokumentu na vhodnější styl (například místo standardních použijte styly z třídy stylů vyvinutých holandskou skupinou sdružení uživatelů  $\TeX$ u `ntgclass: artikel1`, `rapport`, `boek` a za moment mu ukažte přesazený dokument s jiným designem a vysazený v jiné rodině písem.

**D:**Sázíte složitý dokument a vše co potřebujete od okolí pro šachovnice po hieroglyfy jste našli jako  $\LaTeX$ ové styly na právě obdrženém CD-ROM  $\TeX$ live II připravené ihned k použití (`plug&play`)?

**O:**`latex mydocument`

**D:**Sázíte část dokumentu do knihy, která bude v  $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\LaTeX$ u, pro svého nadřízeného, který nehodlá používat nic jiného?

**O:**Prolistujte manuál a použijte ho pro značení své části také!

**D:**Chcete si napsat knihu s názvem  $\TeX$ book naruby,  $\LaTeX$  neznáte, píšete o něm pomlouvačné články, ale s  $\TeX$ bookem chodíte denně do postele?

**O:**Napište si svá makra pro svou knihu v plain $\TeX$ u!

**D:**Jste intelektuál?

**O:**Použijte  $\LaTeX$ !<sup>3</sup>

---

<sup>3</sup>Autorem tohoto doporučení je Phil Taylor.

## Značkování a její minimalizace

Kdo dočetl až sem, tomu asi vrtá hlavou, jak to vše souvisí s názorem anglického filosofa čtrnáctého století a jeho principem šetrnosti?

Jedním z hlavních typografických principů je princip jednotnosti a zejména u technických dokumentů zviditelnění logické struktury dokumentu. Logická struktura dokumentu je tedy mimo vlastního textu to hlavní (případně i jediné) značení, které je autorské. Koncentrace na ni a separace vizuálních prvků sazby vyžaduje určitou disciplínu. Její dodržení implicitně na začátečníkovi, který neumí vytvářet makra, vynucuje L<sup>A</sup>T<sub>E</sub>X, což je nesporná výhoda. Začátečník, který začíná v plainu, by na toto měl od začátku s vytvářením maker vhodného značkování myslet. Jinak se mu může stát, že dopadne stejně jako jeho soused u WYSIWYG programu, jehož myš svádí k nekonzistentnímu klikání a značení logicky stejných entit vizuálně různě. Zde by měla nastoupit pomyslná břitva a minimalita použitého značení pro vizuální prvky v textu (a typografii tak skrýt do těl maker – sázecích procedur).

## Validace

Validace, čili kontrola dokumentu, zda jeho jazyk značení vyhovuje formální gramatice (v jazyce SGML se nazývá DTD, Document Type Definition), umožňuje automatizaci kontroly struktury a úplnosti dokumentu. Tak jako u každého programování má autor překladač, který mu říká, kde a proč udělal chybu. Výhodou tohoto postupu je například oproti (plain)T<sub>E</sub>Xu zejména lepší chybová diagnostika, možnost snadno popsateľných automatizovaných transformací dokumentu. To při dokumentu v plainT<sub>E</sub>Xu využívajícím plně možností aktivních znaků, nestandardního oddělování parametrů maker a dalších fines, není možné. Vhodný jazyk značek je též klíčový pro automatizaci a efektivitu práce s dokumentem. Velké publikační domy dávno poznaly výhody formálního jazyka SGML a z toho plynoucí možnosti transformací dokumentů, validace a dalších nesporných výhod. Přecházet však na prostředky SGML kvůli deseti stránkám dokumentů měsíčně se však rozhodně nevyplatí. Alternativou může být balík T<sub>E</sub>Xových maker (ATML), který sází přímo ze značkování HTML při současné možnosti publikace na WWW a použití validačních prostředků jako nsgmls (<http://www.jclark.com/sp/nsgmls.htm>).

## Závěr

Pokud jste čekali nějakou kuchařku či jednoznačné doporučení, asi jste zklamáni. Každá situace je specifická a při každém rozhodnutí je třeba zvažovat

maximum vstupním podmínek a jejich priority. U  $\text{\TeX}$ u je krásné to, že k cíli vede cestiček většinou několik a sazeč má možnost svobodné volby na každém kroku. Pokud mu  $\text{\TeX}$  nedělá vše podle jeho představ, může dokonce upravit i jeho kód pro svou konkrétní aplikaci. Jako u každého jiného programování však efektivita práce kolísá až o řád v závislosti na schopnostech programátora, dostupných prostředcích, disciplíně a dalších faktorech. Doufám, že článek umožní aspoň základní orientaci novým adeptům černého řemesla pomocí  $\text{\TeX}$ u.

*Petr Sojka  
Fakulta informatiky MU  
Botanická 68a  
602 00 Brno  
sojka@muni.cz*

---

---

## Hierarchie $\text{\TeX}$ istů

---

### Beginner:

- Is always searching for a  $\text{\TeX}$  help menu in the text editor.
- Does not know what an operation system is, but thinks that  $\text{\TeX}$  is a part of it.
- Has no idea what `\section` is for.
- Queues the `.TEX` file to the printer.
- Has serious troubles not to forget the `\` before a command name.
- Cannot understand why `WINWORD` is not installed on this nasty machine.

### Novice:

- Knows that no help menu is available.
- Uses `\begin {...}`, but `\end`'s only 50% of them.
- Queues the `.DVI` file to the printer.
- Thinks that penalties are a bad thing for soccer players.
- Has heard somewhere that  $\text{\TeX}$  is good for mathematical formulas.
- Cannot understand why `\section` is better than `{\Large \bf}`.
- Has never heard about `METAFONT`.
- Is amazed why `WINWORD` was so much faster in printing.
- Does not know what a tabular is, but needs one in his document.



- Wonders how to print a backslash and tries `\\`.
- Sometimes forgets `\documentstyle` or `\begin {document}`.

### User:

- Uses `\section` etc., but has never heard about optional arguments.
- Once has created a table of contents using `\tableofcontents`.
- Has heard about dimensions, but never seen one.
- Occasionally types `/` instead of `\`.
- Wonders why some guys like T<sub>E</sub>X so much.
- Wants to write a whole paragraph underlined, but has no idea how.
- Thinks that `\newcommand` is only for freaks.
- Uses tabulars, but doesn't know what "`\halign`" in the error message means.
- Knows that style files contain style information.
- Uses the terms "T<sub>E</sub>X" and "L<sup>A</sup>T<sub>E</sub>X" as synonyms.
- Has learnt that typing "H" after an error message is sometimes funny and never useful.

### Intermediate:

- Uses list environments confidently.
- Has never heard about `\def`.
- Had first bad experience with fragile commands.
- Thinks that Knuth is the name of the showmaster in the local TV channel.
- Has figured out that optional arguments need not be given.
- Has some strange wishes to change L<sup>A</sup>T<sub>E</sub>X style, but fortunately is unable to do it.
- Is proud to have used `\newcommand` or `\def` two times in life.
- Has heard that METAFONT is for font generation.

### Knowledgeable:

- Generates displayed equations, but inexpertly.
- Once has seen *The T<sub>E</sub>Xbook*.
- Writes simple macros in every document.
- Would like to know how to implement optional parameters.
- Frequently asks the local T<sub>E</sub>X guru for troubleshooting.
- Reads `comp.text.tex` every month.
- Considers macro expansion rules completely uninteresting.
- Had tried the picture environment, but returned to pen and ruler quickly.
- Has just to learn about `\displaystyle` to `\scriptscriptstyle`.

- Thinks that `\def` is an abbreviation for `\newcommand`.
- Changes counter values to number sections beginning with zero.
- Has been told that one can alter style files, but doesn't believe.

### Advanced:

- Uses most L<sup>A</sup>T<sub>E</sub>X features with comfort.
- Has read *The T<sub>E</sub>Xbook*, but failed to solve exercise 6.1.
- Thinks that dangerous bent is dangerous.
- Posts his troubles to `comp.text.tex`.
- Cannot imagine an application for `\csname`.
- Has learnt about horizontal and vertical mode.
- Believes that WEB is an arm of SpiderMan.
- Knows that glue isn't dirty.
- Thinks that one must be ill to use `\expandafter`.
- Wonders who is smart enough to write DVI drivers.
- Has heard about `\gdef`, but regards it worthless.

### TeXnician:

- Has read and understood *The T<sub>E</sub>Xbook*, except appendix D.
- Uses `\input` to load his own macro package.
- Would like to use `\if...some day`, but does not see an application for it.
- Has read about `sfcodes`.
- Writes macros with delimited parameters.
- Knows something about inner vertical and restricted horizontal mode.
- Thinks that `\futurelet` is a useless nightmare.
- Has once used `\edef`.
- Implements exotic fonts on his laser printer with METAFONT.

### Hacker:

- Uses `\csname` instead of `\ifcase` if there are more than two alternatives.
- Has installed emT<sub>E</sub>X on several PCs.
- Knows how to generate logos etc. with METAFONT.
- Uses `\xdef` when necessary.
- Can answer any question of novices to intermediates.
- Knows how to fix bugs in knowledgeable's macros.
- Cannot remember when he used `\newcommand` the last time.
- Starts each document with `\documentstyle [mymacros, mylists, mygraphix] {private}`
- Posts answers to `comp.text.tex` every week.
- Has learnt that active characters can do a lot.

### Guru:

- Is completely familiar with DVI formats.
- Has come in troubles since his METAFONT files filled the whole user disk space.
- Fixes bugs in printer drivers by looking at the *C* source.
- Can answer any question about T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X or A<sub>M</sub>S-T<sub>E</sub>X after a few seconds of thought.
- Uses `\afterassignment` twice the day.
- Has read “*The Art of Computer Programming*” and quotes it frequently.
- Cannot imagine why any human being should use WINWORD.
- Begins every posting to `comp.text.tex` with “*As you can read in my new book...*”

### Wizard:

- Has met Mr. Knuth on a congress of typesetting.
- Keeps the world’s record in the usage of `\futurelet`.
- Doesn’t fix bugs in DVI drivers, but writes own instead of.
- Writes document styles during his lunch.
- Has designed a new font to make his thesis optically unique.
- Never reads `comp.text.tex`, why should he?
- Has ported T<sub>E</sub>X to a *Cray 90*.

### TeXpert:

- Fixes bugs in `TEX.WEB`.
- Installs T<sub>E</sub>X on a Hypercube serving 256 printers simultaneously.
- Develops a new font family for each person he writes a letter to.
- Currently works on the problem of a perfect hyphenation in any language.
- Corrects bad layout by patching the DVI file.
- During weekend, writes DVI drivers for printers constructed next year.
- Spends every Christmas with Donald and has married Leslie’s daughter.

Kam zařadíte sebe dnes? A kam zítra?

---

---

# The T<sub>E</sub>X Live Guide, version 2

SEBASTIAN RAHTZ, MICHEL GOOSSENS

---

## Contents

---

<b>1</b>	<b>Introduction</b>	<b>35</b>
1.1	History and acknowledgements . . . . .	35
1.2	Future versions . . . . .	37
<b>2</b>	<b>Structure and contents of the CD-ROM</b>	<b>37</b>
2.1	The TDS tree . . . . .	38
<b>3</b>	<b>Installation and use under Unix</b>	<b>38</b>
3.1	Running T <sub>E</sub> X Live from the CD-ROM . . . . .	39
3.2	Installing T <sub>E</sub> X Live to a hard disk . . . . .	40
3.3	Installing individual packages from T <sub>E</sub> X Live to a hard disk . . . . .	43
3.4	texconfig . . . . .	44
3.5	Building on a new platform . . . . .	44
<b>4</b>	<b>A user's guide to the Web2c system</b>	<b>45</b>
4.1	Kpathsea path searching . . . . .	46
4.2	Filename databases . . . . .	50
4.3	Runtime options . . . . .	57
<b>5</b>	<b>Other packages on the CD-ROM</b>	<b>58</b>
5.1	OzT <sub>E</sub> X . . . . .	58
5.2	CMacT <sub>E</sub> X . . . . .	60
5.3	MiKT <sub>E</sub> X . . . . .	60
5.4	emT <sub>E</sub> X . . . . .	60
<b>6</b>	<b>The texmf.cnf file</b>	<b>60</b>
<b>7</b>	<b>Catalogue of Packages</b>	<b>68</b>

---

# 1 Introduction

This documentation describes the main features of the **T<sub>E</sub>X Live** CD-ROM, a T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X distribution for Unix, Windows32, Amiga and NeXT systems, that includes T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, METAFONT, MetaPost, many other programs such as Makeindex, dvips, xdvi and BIBT<sub>E</sub>X; and a very complete set of macros, fonts and documentation conforming to the *T<sub>E</sub>X Directory Standard* which can be used with nearly every T<sub>E</sub>X setup.

This T<sub>E</sub>X package uses the Web2c implementation of the programs, which tries to make T<sub>E</sub>Xing as easy as possible, and takes full advantage of the efficient and highly customizable Kpathsea library from Karl Berry. It can be run either directly from the CD-ROM, or installed on a hard disk.

The **T<sub>E</sub>X Live** runnable systems contain two experimental extensions to normal T<sub>E</sub>X:

1.  $\varepsilon$ -T<sub>E</sub>X, which adds a small but powerful set of new primitives, and the T<sub>E</sub>X--X<sub>E</sub>L extensions for left to right typesetting; in default mode,  $\varepsilon$ -T<sub>E</sub>X is 100% compatible with ordinary T<sub>E</sub>X. See `share/texmf/doc/html/e-tex/etex.htm` on the CD-ROM for details.
2. pdfT<sub>E</sub>X, which can optionally write Acrobat PDF format instead of dvi; there is no formal documentation for this yet, but the file `share/texmf/tex/pdftex/example.tex` shows how it is used. The L<sup>A</sup>T<sub>E</sub>X `hyperref` package has an option ‘pdftex’ which turns on all the program features.

While  $\varepsilon$ -T<sub>E</sub>X is stable, pdfT<sub>E</sub>X is under continual development; the version on the CD-ROM may not be stable. Most platforms have version 0.11 of May 7th, but some have a slightly earlier one of May 5th, which may have problems including PNG files.

The entire GUTenberg distribution for Windows is included on the CD-ROM, ready to install, as are the following complete packages:

- OzT<sub>E</sub>X 3.0 for Macintosh
- CMacTeX 2.6 for Macintosh
- Macintosh utilities (Alpha, Excalibur, etc.)
- MikTeX for Windows 95
- emTeX for MSDOS and OS/2
- T<sub>E</sub>X shells for Windows and DOS (Winedt, e4t, TeXtelmExtel, emTeXgi)

These are provided unchanged from CTAN, and have not been integrated in any way with the rest of the CD-ROM.

## 1.1 History and acknowledgements

This CD-ROM distribution is a joint effort by the T<sub>E</sub>X Users Group, the UK T<sub>E</sub>X Users Group, and the French T<sub>E</sub>X Users (GUTenberg), with the support of the Dutch, German and Czech/Slovak user groups. Discussion began in late 1993

when the Dutch T<sub>E</sub>X Users Group was starting work on its 4AllT<sub>E</sub>X CD-ROM for MSDOS users, and it was hoped at that time to issue a single, rational, CD-ROM for all systems. This was far too ambitious a target, but it did spawn not only the very successful 4AllT<sub>E</sub>X CD-ROM, but also the TUG Technical Council working group on a *T<sub>E</sub>X Directory Structure*, which specified how to create consistent and manageable collections of T<sub>E</sub>X support files. The final draft of the TDS was published in the December 1995 issue of *TUGboat*, and it was clear from an early stage that one desirable product would be a model structure on CD-ROM. The CD-ROM you now have is a very direct result of the working group's deliberations. It was also clear that the success of the 4AllT<sub>E</sub>X CD-ROM meant that Unix users would benefit from a similarly easy system, and this is the other main strand of **T<sub>E</sub>X Live**.

We undertook to make a new Unix-based TDS CD-ROM in the autumn of 1995, and quickly identified Thomas Esser's teT<sub>E</sub>X as the ideal setup, as it already had multi-platform support and was built with portability across file systems in mind. Thomas agreed to help, and work began seriously at the start of 1996. The first edition was released in May 1996. At the start of 1997, Karl Berry completed a major new release of his Web2c package, which included nearly all the features which Thomas Esser had added in teT<sub>E</sub>X, and we decided to base the 2nd edition of the CD-ROM on the standard Web2c, with the addition of teT<sub>E</sub>X's `texconfig` script.

We are particularly grateful to: Karl Berry for extra advice, encouragement, and (of course) for providing the Web2c distribution; Thomas Esser, without whose marvellous teT<sub>E</sub>X package this CD-ROM would certainly not exist, and whose continual help makes it a better product; and Ulrik Vieth, for checking many assumptions at the start, and providing a great deal of extra material for the documentation tree.

Fabrice Popineau did the excellent port of Web2c 7.0 to Windows 95/NT and provided much help; Andreas Scherer contributed the Amiga compilation; Gregor Hoffleit contributed the TeXview material for NextStep users, and the NextStep binaries. At Florida State University Supercomputer Research Institute, Mimi Burbank arranged access to a slew of different computers to compile T<sub>E</sub>X on, and acted as an essential guinea-pig whenever asked. Michel Goossens provided access to computers at CERN, and Robin Fairbairns stepped in to provide an Alpha running Linux at Cambridge.

Some of this documentation is drawn from the teT<sub>E</sub>X guide by Thomas Esser and Dirk Hillbrecht; the catalogue of packages depends very much on the ongoing work of Graham Williams (<mailto:Graham.Williams@cbr.dit.csiro.au>), who kindly agreed to allow us to use it here. Mimi Burbank, Robin Fairbairns and Ulrik Vieth worked hard to improve this text.

## 1.2 Future versions

*This CD-ROM is not a perfect product!* We plan to re-issue it once a year, and would like to provide more help material, more utilities, more installation programs, and (of course) an ever-improved and checked tree of macros and fonts. This work is all done by hard-pressed volunteers in their limited spare time, and a great deal remains to be done. If you can help, don't hesitate to put your name forward!

Corrections, suggestions and additions for future revisions should be sent to:

Sebastian Rahtz  
7 Stratfield Road  
Oxford OX2 7BG  
United Kingdom  
`mailto:s.rahtz@elsevier.co.uk`

Updates, notes, and suggestions will be made available on CTAN in `info/texlive`. A WWW page for information and ordering details is at `http://www.tug.org/tex-live.html`.

## 2 Structure and contents of the CD-ROM

The CD-ROM top level directories are:

- bin** The  $\TeX$  family programs, arranged in separate platform directories;
- info** Documentation in GNU 'info' format for the  $\TeX$  system;
- macintosh** The  $\text{Oz}\TeX$  and  $\text{C}\text{Mac}\TeX$  packages ready to install, plus some other utilities;
- man** Documentation in Unix man pages for the  $\TeX$  system;
- msdos** DOS  $\TeX$  packages —  $\text{em}\TeX$ , and three  $\TeX$  shells;
- support** The source of all programs, including the main  $\text{Web}2\text{c}\ \TeX$  and  $\text{META-FONT}$  distribution; this directory also includes various bits of  $\TeX$ -related software which are *not* installed by default, such as  $\text{Musix}\TeX$  support programs, and a complete set of Ghostscript;
- share** The main support tree of macros, fonts and documentation;
- wingut** The GUTenberg distribution for Windows; this consists of compressed archives which must be unpacked and installed on a hard disk. Please see the detailed instructions in French;
- win32**  $\TeX$  packages for Windows 95 and NT users ( $\text{Mik}\TeX$ , and the original package of the Win32 port of  $\text{Web}2\text{c}$ ).

There are also two installation scripts for Unix systems, `install-cd.sh` and `install-pkg.sh`; we discuss them on p. 38.

## 2.1 The TDS tree

The **T<sub>E</sub>X Live share/texmf** tree consists of various ‘collections’, each of which has a set of ‘packages’, of which there are over 400 on the CD-ROM. Normal installation allows the user to copy all of a collection to a local hard disk from the CD-ROM, but it is also possible to install just one package of a collection. Each of the collections is divided into *basic* (1), *recommended* (2) and *other* (3). The collections are:

**ams** The American Mathematical Society macro packages and fonts

**bibtex** BIB<sub>T<sub>E</sub>X</sub> styles and databases

**doc** General guides and documentation in various formats, including HTML and PDF

**dvips** Support for Rokicki’s dvi to PostScript driver

**fonts** Font sources, metrics, PostScript and bitmap forms

**formats** Eplain, Rev<sub>T<sub>E</sub>X</sub>, physxxx, texsis, alate<sub>x</sub>, text1, lollipop, etc.

**generic** Extra macros for use with any format

**graphics** Macro packages for graphics

**lang** Support for non-English languages

**latex** L<sup>A</sup><sub>T<sub>E</sub>X</sub>, including official tools and all L<sup>A</sup><sub>T<sub>E</sub>X</sub> 2<sub>ε</sub> contributed packages

**metapost** Support for MetaPost

**plain** Macros for plain <sub>T<sub>E</sub>X</sub>

**systems** Binaries for Unix platforms

**texlive** Basic material for the distribution

The appendix starting on p. 68 lists all the packages in alphabetical order with the collection they are found in, and a brief description. Thus all packages in collection **latex1** are what one must have to get started with L<sup>A</sup><sub>T<sub>E</sub>X</sub>, packages in **latex2** are recommended for most users, and **latex3** contains optional packages. The directory **share/texmf/lists** contains lists of all files in each package (used by the installation package).

## 3 Installation and use under Unix

You can use the **T<sub>E</sub>X Live** CD-ROM in three ways:

1. You can mount the CD-ROM on your file system, adjust your **PATH**, and run everything off the CD-ROM; this takes very little disk space, and gives you immediate access to everything on the CD-ROM; although the performance will not be optimal, it is perfectly acceptable on, for instance, PCs running Linux;
2. You can install all or part of the system to your local hard disk; this is the best method for many people, if they have enough disk space to spare (a minimum of about 10 megabytes, or 100 megabytes for a recommended good-sized system);



3. You can install selected packages to work either with your existing  $\text{T}_{\text{E}}\text{X}$  system or a  **$\text{T}_{\text{E}}\text{X}$  Live** system you installed earlier.

Each of these methods is described in more detail in the following sections.

**Warning:** This CD-ROM is in ISO 9660 (High Sierra) format, with Rock Ridge extensions. In order to take full advantage of the CD-ROM on a Unix system, your system needs to be able to use the Rock Ridge extensions. Please consult the documentation for your *mount* command to see if it is possible. If you have several different machines on a local network, see if you can mount the CD-ROM on one which *does* support Rock Ridge, and use it from the others.

Linux, FreeBSD, Sun, SGI and DEC Alpha systems should be able to use the CD-ROM with no problems. We would appreciate receiving detailed advice from other system users who also succeed, for future versions of this documentation.

The discussion below about installation assumes you have been able to mount the CD-ROM with full Rock Ridge compatibility.

### 3.1 Running $\text{T}_{\text{E}}\text{X}$ Live from the CD-ROM

The organisation of Web2c means that you can run programs simply by adding the appropriate directory under `bin` on the CD-ROM to your `PATH`, and the support files will all be found with no further ado. The following table shows the list of available directories and the systems they apply to.

<code>alpha-linux</code>	DEC Alpha Linux
<code>alpha-osf3.2</code>	DEC Alpha OS 3.2
<code>amiweb2c</code>	Amiga
<code>hppa11-hpux9.05</code>	HP9000 HPUX 9.05
<code>hppa11-hpux10.20</code>	HP9000 HPUX 10.20
<code>i386-linux</code>	Intel PC with Linux (ELF)
<code>i586-freebsd2.2</code>	Intel PC with Free BSD
<code>i686-linux</code>	Intel Pentium Pro with Linux
<code>mab-nextstep3</code>	NextStep 3
<code>mips-irix4.0.5</code>	SGI IRIX 4.0.5
<code>mips-irix5.3</code>	SGI IRIX 5.3
<code>mips-irix6.3</code>	SGI IRIX 6.3
<code>mips-ultrix4.4</code>	DECstation Ultrix 4.4
<code>rs6000-aix3.2.5</code>	IBM RS 6000 AIX 3.2.5
<code>rs6000-aix4.1.1</code>	IBM RS 6000 AIX 4.1.1
<code>sparc-sunos4.1.3</code>	Sun Sparc Sunos 4.1.3
<code>sparc-solaris2.5</code>	Sun Sparc Solaris 2.5
<code>sparc-solaris2.4</code>	Sun Sparc Solaris 2.4

sparc-linux  
win32

Sun Sparc Linux  
Windows 95 or NT

You may worry that when you subsequently make fonts or change configuration, things will go wrong because you cannot change files on the CD-ROM. However, you can maintain a parallel, writeable,  $\text{T}_{\text{E}}\text{X}$  tree on your hard disk; this is searched before the main tree on the CD-ROM. The default location is `/usr/local/texmf`, but you can override this by setting the `TEXMFLOCAL` environment variable.

Thus *sh* or *bash* users on an Intel PC running Linux who mount the **T<sub>E</sub>X Live** CD-ROM on `/cdrom` by issuing the command:

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

might add the following to their `.profile` script:

```
PATH=/cdrom/bin/i386-linux:$PATH
```

```
export PATH
```

If in doubt, ask your local system support guru to help you work out how to mount your CD-ROM or which directory to use for your system.

Appropriate support files will be installed on your hard disk the first time you need them. It is a good idea to immediately run the `texconfig` script to initialize things, and check it all works.

### 3.2 Installing **T<sub>E</sub>X Live** to a hard disk

All of the necessary steps to install all or part of the distribution on your hard disk are achieved by mounting the CD-ROM, changing to the top-level directory, and typing:

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

(On some Unix systems, you may need to use `sh5` or `bsh`). This works by accessing lists of collections and packages from the CD-ROM, and trying to guess what sort of computer system you are on; it should start by displaying the following:

```
Initializing collections... Done.
```

```
Counting selected collections... Done.
```

```
Calculating disk space requirements for  
collections...Done.
```

```
Initializing system packages... Done.
```

It will then show the main control screen (Figure 1), which lets you change four things:

1. the type of system you are on, or want to install for;
2. the collections you want to install, at *basic*, *recommended* or *other* level;
3. the location on your hard disk to put the files;
4. some runtime behaviour features.

You choose options by typing a letter or number and pressing return. In the example, a Linux ELF system has been detected, the default of all collections to recommended level has been chosen, and the default installation directory is `/usr/local`; note that the disk space required for the current installation configuration is also displayed. If you make a suggested setup, you need about 100 megabytes of disk free; however, the basic setup will only take about 10 megabytes, and you can enhance it with selected packages as you need them.

Under the directory you choose for installation, the installation script will put the binaries in a subdirectory of `bin`, and the support tree in `share/texmf`.

The `options` item lets you decide whether to make new fonts be created in another location (if you want the main package mounted read-only for most users), and whether to make symbolic links for the *man* and GNU *info* pages in the ‘standard’ locations.

When you choose `<C>` for collections, you will see the display of available collections, the level of installation selected, and the disk space required (Figure 2). You can set alternative levels of installation for each collection, ranging from *none* to *all*. You can either set this for all collections at once, or choose a particular collection and set its level (Figure 3).

When you are finished, return to the main screen, and ask the installation to start. It will take each of the collections and systems that you requested, consult the list of files on the CD-ROM, and build a master list of files to transfer. These will then be copied to your hard disk, and the initialization sequence run (creating format files etc.). When this has finished, all you need do is add the correct subdirectory of `bin` in the `TEX` installation to your

```
===== > TeX Live installation procedure <=====
==> Note: Letters/digits in brackets indicate menu items <===
==>      for commands or configurable options                <===
Detected system: Intel PC with Linux (ELF)
<C> collections:   21 out of 30, disk space required: 163955 kB
<S> systems:      1 out of 20, disk space required:   7946 kB
                    total disk space required: 171901 kB

<D> directories:
      TEXDIR = /usr/local

<O> options:
      [ ] alternate directory for automatically generated fonts
      [ ] create symlinks in standard directories

Other commands:
      <I> start installation, <H> help, <Q> quit
Enter command:
```

Figure 1: Installation screen, example 1

```

Current collections setup:                total size : 171901 kB
=====
      name      selection      size
<1>  ams        [recommended]  6359 kB
<2>  bibtex     [recommended]  6584 kB
<3>  doc        [recommended] 26531 kB
<4>  dvips      [recommended]   563 kB
<5>  fonts      [recommended] 21862 kB
<6>  formats    [recommended]  1003 kB
<7>  generic    [recommended]   501 kB
<8>  graphics   [recommended] 10373 kB
<9>  lang       [recommended]  3287 kB
<W>  metapost   [recommended]  1280 kB
<X>  latex      [recommended] 28333 kB
<Y>  plain      [recommended]   756 kB
<Z>  texlive    [recommended] 56523 kB
                                SUM: 163955 kB
=====
global commands: select <n>one / <b>asic / r<e>commended / <a>ll
                  for all collections
<R>  return to platform menu
<Q>  quit
Enter command to modify current selection:

```

Figure 2: Installation screen, example 2

```

Collection: Fonts
=====
Fonts, including metrics, virtual fonts and sources
=====
<N>  no packages
<B>  basic packages          [ 2007 kB]
<E>  basic + recommended packages [ 21862 kB]
<A>  all packages           [ 34303 kB]
=====
<R>  return to collection menu
<Q>  quit
Enter command:

```

Figure 3: Installation screen, example 3

path, and start using  $\text{T}_{\text{E}}\text{X}$ . If you want to move the binaries up one level, e.g. from `/usr/local/bin/alpha-osf3.2` to `/usr/local/bin`, you need to edit

```
share/texmf/web2c/texmf.cnf and change the line
prefix = $SELFAUTOPARENT
to
```

```
prefix = $SELFAUTODIR
```

You can of course change the value of `prefix` to any directory you like, and move the support directory there.

### 3.3 Installing individual packages from T<sub>E</sub>X Live to a hard disk

You may want to use the T<sub>E</sub>X Live CD-ROM to either update an existing setup, or add features to an earlier installation from the CD-ROM. The main installation program is intended for the first time only, and subsequently you should use the `install-pkg.sh` script on the CD-ROM. Run this by mounting the CD-ROM, changing to the mounted directory, and typing

```
sh install-pkg.sh options
```

The script supports nine options; the first four let you set the individual package you want to install, the whole collection (i.e., `ams2`), the name of the mounted CD-ROM directory, and the name of the directory containing the list files (normally these latter two will be set automatically):

```
--package=name
--collection=name
--cddir=name
--listdir=name
```

What actually happens is controlled by four more switches; the first two allow you to exclude documentation or source files from the installation; the third stops the default action of running `MakeTeXls-R` on completion to rebuild the file database, and the last does nothing but list the files that would be installed:

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

Finally, you can specify that instead of installing the files, the script should make a `tar` archive in a specified location:

```
--archive=name
```

Thus, if we simply wanted to see the files that make up the package `fancyhdr` before we installed, our command and output would be as follows:

```
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
```

Other examples of usage are:

- Install the L<sup>A</sup>T<sub>E</sub>X package `arseneau`:  
`install-pkg.sh --package=arseneau`
- Install the L<sup>A</sup>T<sub>E</sub>X package `alg` with no source files and no documentation:  
`install-pkg.sh --package=alg --nosrc --nodoc`
- Install all the packages available in the ‘extra’ Plain T<sub>E</sub>X collection:  
`install-pkg.sh --collection=plain3`
- Place all files which are need for P<sub>S</sub>Tricks in a tar file in `/tmp`:  
`install-pkg.sh --package=pstricks --archive=/tmp/pstricks.tar`

### 3.4 texconfig

After the installation program has copied all files to their final locations, you can call a program called `texconfig` that allows you to configure the system to fit your local needs. This can be called at any other time to change your setup, with a full-screen (which requires the `dialog` program) or command-line interface. It should be used for all maintenance, like changes of installed printers, or rebuilding of the file database. Both modes have help text to guide you through the facilities.

### 3.5 Building on a new platform

If you have a platform for which we have not provided binary sources, you will need to compile T<sub>E</sub>X and friends from scratch. This is not as hard as it sounds. What you need is all in the directory `support/texk-7.0` on the CD-ROM.

To compile T<sub>E</sub>X, you should get `gcc`, `flex` and a recent version of GNU `make`. `gcc-2.5.8`, `flex-2.4.7` and GNU `make-3.72.1` or newer should be fine. You may be able to work with other C compilers and Make programs, but you will need a good understanding of building Unix programs to sort out problems.

You should first install the support tree from the **T<sub>E</sub>X Live** CD-ROM (do a basic install, with no system binaries chosen). Then copy the `texk-7.0` directory to your disk, and run

```
configure -prefix=$TEXMF
```

where `$TEXMF` is the place where you installed **T<sub>E</sub>X Live**.

Now type `make install-exec` and relax...

## 4 A user's guide to the Web2c system

Web2c contains a set of T<sub>E</sub>X-related programs, i.e., T<sub>E</sub>X itself, METAFONT, MetaPost, BIBT<sub>E</sub>X, etc. The original implementation was by Tomas Rokicki, who in 1987 developed a first T<sub>E</sub>X-to-C system adapting change files under Unix, which were primarily the work of Howard Trickey and Pavel Curtis. Tim Morgan became the maintainer of the system, and during this period the name changed to Web-to-C. In 1990, Karl Berry took over the work, assisted by dozens of additional contributors. The latest result is Web2c Version 7, which was released in February 1997, and forms the basis of the present **T<sub>E</sub>X Live** CD-ROM.

The Web2c 7.0 system runs on Unix, Windows 95/NT, DOS, Amiga, and other operating systems. It uses Knuth's original sources for T<sub>E</sub>X and other basic programs written in `web` and translates them into C source code. Moreover, the system offers a large set of macros and functions developed to augment the original T<sub>E</sub>X software. The most commonly used components are:

- `bibtex` Maintaining bibliographies.
- `dmp troff` to MPX (MetaPost pictures).
- `dvicopy` Virtual font expansion.
- `dvitomp` DVI to MPX (MetaPost pictures).
- `dvitype` DVI to human-readable text.
- `gftodvi` Generic font proofsheets.
- `gftopk` Generic to packed fonts.
- `gftype` GF to human-readable text.
- `makempx` MetaPost label typesetting.
- `mf` Creating typeface families.
- `mft` Prettyprinting METAFONT source.
- `mpost` Creating technical diagrams.
- `mpto` MetaPost label extraction.
- `newer` Compare modification times.
- `patgen` Creating hyphenation patterns.
- `pktogf` Packed to generic fonts.
- `pktype` PK to human-readable text.
- `pltotf` Property list to TFM.
- `pooltype` Display WEB pool files.
- `tangle` WEB to Pascal.
- `tex` Typesetting.
- `tftopl` TFM to property list.
- `vftovp` Virtual font to virtual property list
- `vptovf` Virtual property list to virtual font.
- `weave` WEB to T<sub>E</sub>X.

The precise functions and syntax of these programs are described in the documentation of the individual packages or of Web2c itself. However, knowing a few principles governing the whole family of programs will help you to benefit optimally from your Web2c installation.

All programs honor the standard GNU options:

```
--help      print basic usage summary.  
--verbose   print detailed progress report.  
--version   print version information, then exit.
```

For locating files the Web2c programs use the path searching library Kpathsea. This library uses a combination of environment variables and a few configuration files to optimize searching the T<sub>E</sub>X directory tree. Web2c 7.0 can handle more than one directory tree simultaneously, which is useful if one wants to maintain T<sub>E</sub>X's standard distribution and local extensions in two distinct trees. To speed up file searches the root of each tree has a file `ls-R`, containing an entry showing the name and relative pathname for all files “hanging” under that root.

#### 4.1 Kpathsea path searching

Let us first describe the generic path searching mechanism of the Kpathsea library.

We call a *search path* a colon- or semicolon-separated list of *path elements*, which are basically directory names. A search path can come from (a combination of) many sources. To look up a file “`my_file`” along a path “`./dir`”, Kpathsea checks each element of the path in turn: first `./my_file`, then `/dir/my_file`, returning the first match (or possibly all matches).

In order to adapt optimally to all operating systems' conventions, on non-Unix systems Kpathsea can use filename separators different from “colon” (“`:`”) and “slash” (“`/`”).

To check a particular path element  $p$ , Kpathsea first checks if a prebuilt database (see “Filename database” on p. 50) applies to  $p$ , i.e., if the database is in a directory that is a prefix of  $p$ . If so, the path specification is matched against the contents of the database.

If the database does not exist, or does not apply to this path element, or contains no matches, the filesystem is searched (if this was not forbidden by a specification starting with “`!`” and if the file being searched for must exist). Kpathsea constructs the list of directories that correspond to this path element, and then checks in each for the file being searched for.

The “file must exist” condition comes into play with VF files and input files read by T<sub>E</sub>X's `\openin` command. Such files may not exist (e.g., `cmr10.vf`), and so it would be wrong to search the disk for them. Therefore, if you fail to update `ls-R` when you install a new VF file, it will never be found.



Each path element is checked in turn: first the database, then the disk. If a match is found, the search stops and the result is returned.

Although the simplest and most common path element is a directory name, Kpathsea supports additional features in search paths: layered default values, environment variable names, config file values, users' home directories, and recursive subdirectory searching. Thus, we say that Kpathsea *expands* a path element, meaning transforming all the specifications into basic directory name or names. This is described in the following sections in the same order as it takes place.

Note that if the filename being searched for is absolute or explicitly relative, i.e., starts with “/” or “./” or “./.”, Kpathsea simply checks if that file exists.

### 4.1.1 Path sources

A search path can come from many sources. In the order in which Kpathsea uses them:

1. A user-set environment variable, for instance, `TEXINPUTS`. Environment variables with a period and a program name appended override; e.g., if “`latex`” is the name of the program being run, then `TEXINPUTS.latex` will override `TEXINPUTS`.
2. A program-specific configuration file, for example, a line “`S /a:/b`” in `dvips' config.ps`.
3. A Kpathsea configuration file `texmf.cnf`, containing a line like: “`TEXINPUTS=/c:/d`” (see below).
4. The compile-time default.

You can see each of these values for a given search path by using the debugging options (see “Debugging actions” on p. 55).

### 4.1.2 Config files

Kpathsea reads *runtime configuration files* named `texmf.cnf` for search path and other definitions. The search path used to look for these files is named `TEXMFCNF` (by default such a file lives in the `share/texmf/web2c` subdirectory). All `texmf.cnf` files in the search path will be read and definitions in earlier files override those in later files. Thus, with a search path of “`.: $\$$ TEXMF`”, values from `./texmf.cnf` override those from  `$\$$ TEXMF/texmf.cnf`.

While reading the description of the format of the file `texmf.cnf` below, please also refer to p. 60, which lists the `texmf.cnf` file on the CD-ROM.

- Comments start with “`%`” and continue to the end of the line.
- Blank lines are ignored.
- A `\` at the end of a line acts as a continuation character, i.e., the next line is appended. Whitespace at the beginning of continuation lines is not ignored.

- Each remaining line must look like  

$$\mathit{variable}[\mathit{.programe}] [=] \mathit{value}$$
where the “=” and surrounding whitespace is optional.
- The *variable* name may contain any character other than whitespace, “=”, or “.”, but sticking to “A-Za-z\_” is safest.
- If “*.programe*” is present, the definition only applies if the program that is running is named *programe* or *programe.exe*. This allows different flavors of T<sub>E</sub>X to have different search paths, for example.
- *value* may contain any characters except “%” and “@”. The “*\$var.prog*” feature is not available on the right-hand side; instead, you must use an additional variable (see the definition of the variable `latex2e_inputs` for example). A “;” in *value* is translated to “:” if running under Unix; this is useful to write a single `texmf.cnf` which can be used under both Unix and NT.
- All definitions are read before anything is expanded, so you can use variables before they are defined.

A configuration file fragment illustrating most of these points is shown below:

```
% TeX input files -- i.e.,
% anything found by \input or \openin ...
latex209_inputs = .:$TEXMF/tex/latex209//:$TEXMF/tex//
latex2e_inputs = .:$TEXMF/tex/latex//:$TEXMF/tex//
TEXINPUTS = .:$TEXMF/tex//
TEXINPUTS.latex209 = $latex209_inputs
TEXINPUTS.latex2e = $latex2e_inputs
TEXINPUTS.latex = $latex2e_inputs
```

### 4.1.3 Path expansion

Kpathsea recognizes certain special characters and constructions in search paths, similar to that in Unix shells. As an general example, the following complex path: `~$USER/{foo,bar}//baz` expands to all subdirectories under directories `foo` and `bar` in `$USER`’s home directory that contain a directory or file `baz`. These expansions are explained in the sections below.

### 4.1.4 Default expansion

If the highest-priority search path (see “Path sources” on p. 47) contains an *extra colon* (i.e., leading, trailing, or doubled), Kpathsea inserts at that point the next-highest-priority search path that is defined. If that inserted path has an extra colon, the same happens with the next-highest. For example, given an environment variable setting

```
setenv TEXINPUTS /home/karl:
and a TEXINPUTS value from texmf.cnf of
.: $\$$ TEXMF//tex
then the final value used for searching will be:
/home/karl:. $\$$ TEXMF//tex
```

Since it would be useless to insert the default value in more than one place, Kpathsea changes only one extra “:” and leaves any others in place: it checks first for a leading “:”, then a trailing “:”, then a doubled “:”.

#### 4.1.5 Brace expansion

A useful feature is brace expansion, which means that, for instance,  $v\{a,b\}w$  expands to  $vaw:vbw$ . Nesting is allowed. This can be used to implement multiple  $\TeX$  hierarchies, by assigning a brace list to  $\$TEXMF$ .

For example, in `texmf.cnf`, you find the following definition:

```
texdir =  $\$$ TEXMFLOCAL/tex,!! $\$$ TEXMFMAIN/tex
```

Then you can write something like:

```
TEXINPUTS = .; $\$$ texdir//
```

which means that after looking in the current directory, first the full directory tree  $\$TEXMFLOCAL/tex$  (on disk) and then the  $!!\$TEXMFMAIN/tex$  tree (using the data base file `ls-R` *only*) will be searched. It is a convenient way for running two parallel  $\TeX$  structures, one “frozen” (like on a CD-ROM) and the other being continuously updated with new versions as they become available. By using the  $\$texdir$  variable in all definitions, one is sure to always search the up-to-date tree first.

#### 4.1.6 Subdirectory expansion

Two or more consecutive slashes in a path element following a directory  $d$  is replaced by all subdirectories of  $d$ : first those subdirectories directly under  $d$ , then the subsubdirectories under those, and so on. At each level, the order in which the directories are searched is *unspecified*.

If you specify any filename components after the “//”, only subdirectories with matching components are included. For example, “/a//b” expands into directories /a/1/b, /a/2/b, /a/1/1/b, and so on, but not /a/b/c or /a/1.

Multiple “//” constructs in a path are possible, but “//” at the beginning of a path is ignored.

#### 4.1.7 List of special characters and their meaning: a summary

The following list summarises the meaning of special characters in Kpathsea configuration files.

- : Separator in path specification; at the beginning or the end of a path it substitutes the “default” path expansion.
- ; Separator on non-Unix systems (acts like :).
- \$ Variable expansion.
- ~ Represents the user’s home directory.
- {...} Brace expansion, e.g., a{1,2}b will become a1b:a2b.
- // Subdirectory expansion. It can occur in the middle or at the end of a path (not at the beginning).
- % Start of comment.
- \ Continuation character (allows multi-line entries).
- !! Search *only* database to locate file, *do not* search the disk.

## 4.2 Filename databases

Kpathsea goes to some lengths to minimize disk accesses for searches. Nevertheless, at installations with enough directories, searching each possible directory for a given file can take an excessively long time (this is especially true if many hundreds of font directories have to be traversed.) Therefore, Kpathsea can use an externally-built “database” file named `ls-R` that maps files to directories, thus avoiding the need to exhaustively search the disk.

A second database file `aliases` allows you to give additional names to the files listed in `ls-R`. This can be helpful to adapt to “8.3” filename conventions in source files.

### 4.2.1 `ls-R` filename database

As explained above, the name of the main filename database must be `ls-R`. You can put one at the root of each  $\TeX$  installation hierarchy you wish to search (`$TEXMF` by default); most sites have only one hierarchy. Kpathsea looks for `ls-R` files along the `TEXMFDBS` path.

The recommended way to create and maintain “`ls-R`” is to run the `MakeTeXls-R` script coming with the distribution. It is invoked by the various “`MakeTeX...`” scripts. In principle, this script just runs the command

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

presuming your system’s `ls` produces the right output format (GNU’s `ls` is all right). To ensure that the database is always up to date, it is easiest to rebuild it regularly via `cron`, so that for changes in the installed files — perhaps after installing or updating a  $\LaTeX$  package — the file `ls-R` is automatically updated.

If a file is not found in the database, by default Kpathsea goes ahead and searches the disk. If a particular path element begins with “`!!`”, however, *only* the database will be searched for that element, never the disk.

## 4.2.2 kpsewhich: Standalone path searching

The `kpsewhich` program exercises path searching independent of any particular application. This can be useful as a sort of `find` program to locate files in  $\TeX$  hierarchies (this is used heavily in the distributed “`MakeTeX...`” scripts).

`kpsewhich option... filename...`

Options can start with either “-” or “--”, and any unambiguous abbreviation is accepted.

`Kpathsea` looks up each non-option argument on the command line as a filename, and returns the first file found. There is no option to return all the files with a particular name (you can run the Unix “`find`” utility for that).

The more important options are described next.

`--dpi=num` Set the resolution to *num*; this only affects “`gf`” and “`pk`” lookups. “-D” is a synonym, for compatibility with `dvips`. Default is 600.

`--format=name`

Set the format for lookup to *name*. By default, the format is guessed from the filename. In fact, the recognized filename extensions and the allowable *names* (including any leading “.”) are the same.

You can also specify an integer for *name*; this is the only way to specify formats that don’t have an associated suffix, such as MetaPost support files and `dvips` configuration files. It’s also somewhat faster, since no unused formats need to be initialized. The integers appear in the output of “`--help`”. Currently recognized file type numbers, with their description, possible file extensions, and the corresponding environment variables (between parentheses<sup>1</sup>) as follows:

- 0 Generic font files `.gf` (GFFONTS, GLYPHONTS, TEXFONTS)
- 1 packed font files `.pk` (PKFONTS, TEXPKS, GLYPHONTS, TEXFONTS)
- 2  $\TeX$  bitmap font (GLYPHONTS)
- 3 Adobe PostScript font metrics `.afm` (AFMFONTS)
- 4 METAFONT memory dump `.base` (MFBASES, TEXMFINI)
- 5 BIB $\TeX$  bibliography database `.bib` (BIBINPUTS, TEXBIB)
- 6 BIB $\TeX$  styles `.bst` (BSTINPUTS)
- 7 Runtime configuration files `.cnf` (TEXMFCNF)
- 8 Web2c filename database `ls-R` (TEXMFDBS)
- 9  $\TeX$  memory dump `.fmt` (TEXFORMATS, TEXMFINI)
- 10  $\TeX$  generic font maps `.map` (TEXFONTMAPS)
- 11 MetaPost memory dump `.mem` (MPMEMS, TEXMFINI)
- 12 METAFONT source files `.mf` (MFINPUTS)
- 13 METAFONT program strings `.pool` (MFPOOL, TEXMFINI)
- 14 METAFONT prettyprinter style files `.mft` (MFTINPUTS)
- 15 MetaPost sources `.mp` (MPINPUTS)
- 16 MetaPost program strings `.pool` (MPPPOOL, TEXMFINI)

---

<sup>1</sup>You can find definitions for these environment variables in the file `texmf.cnf` (p. 60)

- 17 MetaPost support files (MPSUPPORT)
- 18  $\Omega$  compiled process .ocp (OCPINPUTS)
- 19  $\Omega$  font metrics .ofm (OFMFonts, TEXFonts)
- 20  $\Omega$  property list .opl (OPLFonts, TEXFonts)
- 21  $\Omega$  translation process files .otp (OTPinputs)
- 22  $\Omega$  virtual fonts .ovf (OVFFonts, TEXFonts)
- 23  $\Omega$  virtual property lists .ovp (OVPFonts, TEXFonts)
- 24 graphics/figure .eps .epsi (TEXPicts, TEXInputs)
- 25 Source input files read by  $\TeX$  .tex .ltx .dtx .texi .texinfo  
.txi .cls .sty .eps .epsi (TEXInputs)
- 26  $\TeX$  documentation .ps .pdf .doc .txt (TEXDocs)
- 27  $\TeX$  program strings .pool (TEXPOOL, TEXMFINI)
- 28  $\TeX$  system package sources .dtx .ins (TEXSources)
- 29 PostScript header/font .pro (TEXPSHEADERS, PSHEADERS)
- 30 Troff fonts (TRFonts)
- 31  $\TeX$  font metric files .tfm (TFMFonts, TEXFonts)
- 32 PostScript type1 fonts .pfa .pfb  
(T1Fonts, T1Inputs, TEXPSHEADERS, PSHEADERS)
- 33 virtual fonts .vf (VFFonts, TEXFonts)
- 34 dvips configuration files config.xxx, xxx.map (TEXCONFIG)
- 35 MakeIndex style files .ist (TEXIDXSTYLE, INDEXSTYLE)

These environment variables are set by default in the configuration file `texmf.cnf`. It is only when you want to override one or more of the values specified in that file that you might want to set them explicitly in your execution environment.

Note that the “`--format`” and “`--path`” options are mutually exclusive.

`--mode=string`

Set the mode name to *string*; this also only affects “`gf`” and “`pk`” lookups. No default: any mode will be found.

`--must-exist`

Do everything possible to find the files, notably including searching the disk. By default, only the `ls-R` database is checked, in the interest of efficiency.

`--path=string`

Search along the path *string* (colon-separated as usual), instead of guessing the search path from the filename. “`//`” and all the usual expansions are supported. The options “`--path`” and “`--format`” are mutually exclusive.

`--prognam=name`

Set the program name to *name*. This can affect the search paths via the “`.prognam`” feature in configuration files. The default is “`kpsewhich`”.

`--show-path=name`

shows the path used for file lookups of file type *name*. Either a filename extension (“`.pk`”, “`.vf`”, etc.) or an integer can be used, just as with

“`--format`” option.  
`--debug=num`  
sets the debugging options to *num*.

### 4.2.3 Examples of use

Let us now have a look at Kpathsea in action.

```
>> kpsewhich -format=.tex article.cls  
/usr/local/share/texmf/tex/latex/base/article.cls
```

We are looking for the file `article.cls` in the  $\TeX$  source file directories (type `.tex`, format type 25). We find it in the subdirectory `tex/latex/base` below the “`TEXMF`” root directory. To save space, in the following examples we will denote with `...` the repetitive part `/usr/local/share/texmf` preceding each file path.

```
>> kpsewhich tugboat.bib  
.../bibtex/bib/beebe/tugboat.bib
```

$\text{BIB}\TeX$  bibliography databases correspond to format type `.bib`. Here we located file `tugboat.bib`.

```
>> kpsewhich cmr10.pk  
.../fonts/pk/ljfour/public/cm/cmr10.600pk
```

```
>> kpsewhich -dpi=300 cmr10.pk
```

```
>> kpsewhich ptmb8r.pk  
.../fonts/pk/modeless/dpi597/ptmb8r.pk
```

```
>> kpsewhich -dpi=300 ptmb8r.pk  
.../fonts/pk/modeless/dpi300/ptmb8r.pk
```

Font bitmap glyph files of type `.pk` correspond to format type 2. They are used by visualization programs like `dvips` and `xdvi`. On our system we found the Computer Modern file `cmr10` for the mode `ljfour`, at a base resolution of 600 dpi (dots per inch). However, when specifying that we are only interested in a resolution of 300dpi (`-dpi=300`) we are told there is no such font available on the system. In fact, a program like `dvips` or `xdvi` would go off and actually build the `.pk` files at the required resolution using the script `MakeTeXPK`. The last two commands look for a file `ptmb8r.pk`. When specifying no explicit resolution the system returns one (at 597 dpi) which is closest to the “default” set in the `MakeTeXPK` script (600 dpi). However, when specifying the desired resolution (300 dpi) the full path name of the relevant target file is shown.

Next we turn our attention to `dvips`’s header (format type 29) and configuration files (format type 34).

```

>> kpsewhich tex.pro
.../dvips/base/tex.pro

>> kpsewhich -format=34 psfonts.map
.../dvips/base/psfonts.map

>> kpsewhich -format=.map config.ps
.../dvips/config/config.ps

```

We first look at a few of the commonly used files, namely the general prolog `tex.pro` for  $\TeX$  support, before turning our attention to the generic configuration file (`config.ps`) and the PostScript font map `psfonts.map`. Note how we fool the system by asking for `config.ps` as if it had a suffix of `.map`.

We now look a little closer at the URW Times PostScript support files. The name for these in Berry's font naming scheme is "`utm`". The first file we look at is the configuration file, which contains the name of the map file.

```

>> kpsewhich -format=34 config.utm
.../dvips/config/config.utm

```

The contents of that file is

```
p +utm.map
```

which points to the file `utm.map`, which we want to locate next.

```

>> kpsewhich utm.map
.../dvips/urw/utm.map

```

In this map file, which resides in `dvips`'s `urw` subdirectory, the file names of the Type1 PostScript fonts referenced are defined. The contents looks like (we only show part of the lines):

```

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

```

Let's, for instance take the Times Regular instance `utmr8a.pfb`, and find its position in the `texmf` directory tree by using a search with format type 32.

```

>> kpsewhich utmr8a.pfb
.../fonts/type1/urw/utm/utmr8a.pfb

```

It should be evident from these few examples how one can easily locate the whereabouts of a given file. This is especially important if you suspect that the wrong version of a file is picked up somehow, since `kpsewhich` will show you the first file encountered.



## 4.2.4 Debugging actions

Sometimes it is necessary to really investigate how a program resolves file references. To make this feasible in a convenient way Kpathsea offers various debug levels:

- 1 `stat` calls (file tests). When running with an up-to-date `ls-R` database this should almost give no output.
- 2 References to hash tables (like `ls-R` database, map files, configuration files).
- 4 File open and close operations.
- 8 General path information for file types searched by Kpathsea. This is useful to find out where a particular path for the file was defined.
- 16 Directory list for each path element (only relevant for searches on disk).
- 32 File searches.

A value of `-1` will set all the above options; in practice you will probably always use these levels if you need any debugging.

Similarly, with the `dvips` program one can, by setting some debug switches, follow in detail where files are picked up from. Alternatively, when a file is not found, the debug trace shows in which directories the program looks for the given file, so that one can get an indication what the problem is.

Generally speaking, as most programs call the Kpathsea library internally, you can select a debug option by using the `KPATHSEA_DEBUG` environment variable, and setting it to (a combination of) values as described in the above list.

Let us consider, as an example, a small  $\text{\LaTeX}$  source file, `hello_world.tex`, which contains the following input.

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

This little file only used the font `cmr10`, so let us look how `dvips` prepares the PostScript file.

```
>> dvips -d4100 hello_world -o
```

In this case we have combined `dvips`'s debug class 4 (font paths) with Kpathsea's path element expansion (see `dvips` Reference Manual). We get something like shown below (we have rearranged the output for easier display).

```
debug:start search(file=texmf.cnf, must_exist=1,
                   find_all=1,
path=./usr/local/bin/texlive:/usr/local/bin:
     /usr/local/bin/share/texmf/web2c:/usr/local:
     /usr/local/share/texmf/web2c:
     /././teTeX/TeX/share/texmf/web2c:).
```

```

kdebug:start search(file=ls-R, must_exist=1,
                    find_all=1,
                    path=/usr/local/texmf:/usr/local/share/texmf).
kdebug:search(ls-R) =>/usr/local/share/texmf/ls-R
kdebug:start search(file=aliases, must_exist=1,
                    find_all=1,
                    path=/usr/local/texmf:/usr/local/share/texmf).
kdebug:search(aliases) =>
kdebug:start search(file=config.ps, must_exist=0,
                    find_all=0,
                    path=./usr/local/texmf/dvips//:
                    !!/usr/local/share/texmf/dvips//).
kdebug:search(config.ps) =>
/usr/local/share/texmf/dvips/config/config.ps
kdebug:start search(file=/root/.dvipsrc,
                    must_exist=0, find_all=0,
                    path=./usr/local/texmf/dvips//:
                    !!/usr/local/share/texmf/dvips//).
kdebug:search($HOME/.dvipsrc) =>
...
kdebug:start search(file=psfonts.map, must_exist=0,
                    find_all=0,
                    path=./usr/local/texmf/dvips//:
                    !!/usr/local/share/texmf/dvips//).
kdebug:search(psfonts.map) =>
/usr/local/share/texmf/dvips/base/psfonts.map
First dvips locates its working files. It first found texmf.cnf (with the
definitions of the paths of the other files), then the file data base ls-R (to
optimize file searching). It goes on to find the generic configuration file
config.ps, and then looks for the customization file .dvipsrc (which, in this
case is not found). Finally dvips locates the generic map file for PostScript
fonts psfonts.map (defining the relation between the internal and external
names for the PostScript fonts).

At this point dvips identifies itself to the user:
dvipsk 5.66a Copyright 1986-97 Radical Eye Software
(www.radicaleye.com)
then goes on to look for the prolog file texc.pro,
kdebug:start search(file=texc.pro, must_exist=0,
                    find_all=0,
                    path=./usr/local/texmf/dvips//:
                    !!/usr/local/share/texmf/dvips//:
                    /usr/local/texmf/fonts//type1//:
                    !!/usr/local/share/texmf/fonts//type1//).
kdebug:search(texc.pro) =>
/usr/local/share/texmf/dvips/base/texc.pro

```

After having found the file, dvips outputs date and time, and informs us that it will generate the file `hello_world.ps`, then that it needs the font file `cmr10`, and that the latter is declared as “resident”

```
' TeX output 1997.05.01:1316' -> hello_world.ps
```

```
Defining font () cmr10 at 10.0pt
```

```
Font cmr10 <CMR10> is resident.
```

Now the search is on for the file `cmr10.tfm`, which is found, then a few more prolog files (not shown), and finally for the Type1 instance `cmr10.pfb` of the font (which is found) and included in the output file (see last line).

```
kdebug:start search(file=cmr10.tfm, must_exist=1,
```

```
                find_all=0,
```

```
    path=./usr/local/texmf/fonts/tfm//:
```

```
        !!/usr/local/share/texmf/fonts/tfm//:
```

```
        /var/tex/fonts/tfm//).
```

```
kdebug:search(cmr10.tfm) =>
```

```
/usr/local/share/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=./usr/local/texmf/dvips//:
```

```
        !!/usr/local/share/texmf/dvips//:
```

```
        /usr/local/texmf/fonts//type1//:
```

```
        !!/usr/local/share/texmf/fonts//type1//).
```

```
kdebug:search(cmr10.pfb) =>
```

```
/usr/local/share/texmf/fonts/type1/public/cm/cmr10.pfb
```

```
<cmr10.pfb>[1]
```

### 4.3 Runtime options

Another of the nice features of Web2c 7.0 is its possibility to control a number of memory parameters (in particular, array sizes) via the runtime file `texmf.cnf` read by Kpathsea. A detailed list of all set-table parameters can be found in that file (see p. 60, Part 3 starting at line 261). The most interesting values are:

`main_memory` Total words of memory available, for  $\TeX$ , METAFONT, and MetaPost. You must make a new format file for each different setting. For instance, you could generate a “huge” version of  $\TeX$ , and call the format file `hugetex.fmt`. Using the standard way of specifying the program name used by Kpathsea the particular value of the `main_memory` variable will

then be read from `texmf.cnf` (See p. 60, line 280 for the generic value and line 281 for the “huge” one instantiated by `huguetex`).

**extra\_mem\_bot** Extra space for “large”  $\TeX$  data structures: boxes, glue, breakpoints, etc. Especially useful if you use  $\text{P}\mathcal{I}\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ .

**font\_mem\_size** Number of words for font information available for  $\TeX$ . This is more or less the total size of all TFM files read.

**hash\_extra** Additional space for the hash table of control sequence names. Approximately 10,000 control sequences can be stored in the main hash table; if you have a large book with numerous cross-references, this might not be enough. On line 297 and 298 of file `texmf.cnf` as shown in p. 60 you see that both the `huguetex` and `pdftex` program invocations ask for an extra 10,000 control sequences (the default value of `hash_extra` is zero, as seen on line 296).

Of course, this facility is no substitute for truly dynamic arrays and memory allocation, but since this is extremely difficult to implement in present  $\TeX$ , these runtime parameters provide a practical compromise allowing some flexibility.

## 5 Other packages on the CD-ROM

While the main portion of  **$\TeX$  Live** (the fonts, macros and documentation) can be used on any  $\TeX$  system, the set of runnable binaries is not suitable for everyone. To make the disk as widely useful as possible, we have included the original distributions of four complete  $\TeX$  systems, two for Macintosh, one for Windows 95, and one for DOS and OS/2. Windows 3.1 users should look at the GUTenberg distribution on the CD-ROM.

### 5.1 Oz $\TeX$ <sup>2</sup>

Oz $\TeX$  is a Macintosh  $\TeX$  system created by Andrew Trevorrow. The Oz $\TeX$  application includes  $\TeX$ , `INITEX`, a DVI previewer, a DVI-to-PostScript translator (Tom Rokicki’s `dvips`) and a driver for QuickDraw printers. Oz $\TeX$  also includes `dvidvi`, `dvicopy`, and Angus Duggan’s PostScript utilities: `psbook`, `psnup`, `psselect` and `pstops`.

The version of `dvips` included in Oz $\TeX$  supports Hyper $\TeX$  and the partial downloading of PostScript fonts. It has also been enhanced for Mac users in a number of ways: Standard Mac PostScript fonts (LWFN files) can be downloaded, fully or partially. All Oz $\TeX$ -specific `\special` commands are supported, such as the inclusion of `PICT/PNTG/EPSF` files. The `dvips` output can be sent directly to the current printer.

---

<sup>2</sup>This section was written by Andrew Trevorrow.

OzTeX's previewer has lots of features to make it easy to proofread DVI files. It can handle PK and PostScript fonts. Anti-aliasing is supported. Virtual fonts are processed on the fly. The previewer supports most of the `\special` commands generated by L<sup>A</sup>T<sub>E</sub>X's `color`, `graphics/x` and `hyperref` packages. It recognizes all dvips-specific `\specials` and those it cannot handle (like rotation) are silently ignored.

OzTeX includes all the most popular formats and macro packages. Plain T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, AMS-T<sub>E</sub>X, AMS-L<sup>A</sup>T<sub>E</sub>X and REV<sub>T</sub>E<sub>X</sub> are all installed and ready to run.

OzTeX is easy to extend and customize. A default configuration file is read when OzTeX starts up; it contains a host of parameters for setting up search paths, telling T<sub>E</sub>X how much memory to allocate for various arrays, specifying which TFMs are for PostScript fonts, etc. A Config menu makes it easy to load other config files at any time. And for even more flexibility, OzTeX can automatically load a specified config file just before typesetting, previewing or printing.

### 5.1.1 Additional programs

The usual assortment of T<sub>E</sub>X-related programs are provided with OzTeX, including OzMF, a Mac implementation of METAFONT, and OzMP, a Mac port of John Hobby's MetaPost program for producing PostScript pictures using a METAFONT-like language.

The following programs are also distributed with OzTeX, courtesy of their authors; BibT<sub>E</sub>X by Vince Darley; MakeIndex by Rick Zaccone; Excalibur, a T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X spelling checker, by Rick Zaccone and Robert Gottshall; and AlphaLite, a T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X-savvy text editor, by Pete Keleher.

For the latest information about OzTeX, keep an eye on the Web page at the URL <http://www.kagi.com/authors/akt/oztex.html>.

An even better way to keep up-to-date is to join the `oztex-info` mailing list. To subscribe, send some e-mail to

`majordomo@maths.adelaide.edu.au`

with the following line in the *body* of the message:

```
subscribe oztex-info
```

OzTeX is distributed as shareware, so you are welcome to try it out before paying the registration fee. The individual fee is US\$30 and the site fee is US\$300. See the "Shareware Fee" item in OzTeX's Help menu for details on how to pay. E-mail support is provided to registered users. Send all queries and comments to Andrew Trevorrow ([akt@kagi.com](mailto:akt@kagi.com)).

## 5.2 CMacTeX<sup>3</sup>

CMacTeX is an implementation of T<sub>E</sub>X for the Macintosh by Thomas Kiffe (<mailto:tkiffe@math.tamu.edu>). It includes the three main parts of any T<sub>E</sub>X installation — T<sub>E</sub>X, METAFONT and dvips. It also includes two dvi previewers, a utility for printing dvi files on a non PostScript printer, a PostScript previewer and numerous utilities for manipulating T<sub>E</sub>X fonts. Full support for the automatic generation of pk font files is an integral part of the distribution. CMacTeX can be configured to work in an integrated fashion with BBedit, Alpha, and MPW. It will run on any Macintosh with 8 MB of RAM and System 7.

CMacTeX is shareware. The registration fee is US\$35 for a single-user license and US\$150 for a site license.

Installation instructions can be found in the file `/macintosh/cmactex/ReadMeFirst`

## 5.3 MiKTeX<sup>4</sup>

MiKTeX 1.07 is an implementation by Christian Schenk (<mailto:cschenk@berlin.snafu.de>) of T<sub>E</sub>X and METAFONT related utilities for Windows NT and Windows 95. The MiKTeX distribution includes T<sub>E</sub>X; L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub> Dec'96 including standard packages; METAFONT; MetaPost; dvips MakeIndex; BIB<sub>T</sub>E<sub>X</sub>; YAP (Yet Another Previewer); TeXware (dvitype etc.); METAFONTware (gftopk etc.); psutils (psselect, pstops etc.); and DVICopy.

Installation instructions can be found in the file `/win32/miktex/README.TXT`

## 5.4 emTeX

The emTeX distribution for DOS and OS/2 is written by Eberhard Mattes (<mailto:mattes@azu.informatik.uni-stuttgart.de>). It includes the T<sub>E</sub>X typesetter, the METAFONT font generation program, printer drivers, screen previewers, and tools like BIB<sub>T</sub>E<sub>X</sub> and MakeIndex. It also includes the macro packages L<sup>A</sup>T<sub>E</sub>X 2.09 and L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>. Fonts are included as pixel files and METAFONT source files.

Installation instructions can be found in the file `/msdos/emtex/README.ENG`

## 6 The `texmf.cnf` file

```
1 % original texmf.cnf -- runtime path configuration file for kpathsea.  
2 % (If you change or delete 'original' on the previous line, the
```

---

<sup>3</sup>This section is taken from the CMacTeX documentation.

<sup>4</sup>This section is drawn from the documentation.

```

3  % distribution won't install its version over yours.)
4  % Public domain.
5  %
6  % What follows is a super-summary of what this .cnf file can
7  % contain. Please read the Kpathsea manual for more information.
8  %
9  % texmf.cnf is generated from texmf.cnf.in, by replacing @var@ with the
10 % value of the Make variable 'var', via a sed file texmf.sed, generated
11 % (once) by kpathsea/Makefile (itself generated from kpathsea/Makefile.in
12 % by configure).
13 %
14 % Any identifier (sticking to A-Za-z_ for names is safest) can be assigned.
15 % The '=' (and surrounding spaces) is optional.
16 % No % or @ in texmf.cnf.in, for the sake of autogeneration.
17 % (However, %'s and @'s can be edited into texmf.cnf
18 % or put in envvar values.)
19 % $foo (or ${foo}) in a value expands to the envvar or cnf value of foo.
20 %
21 % Earlier entries (in the same or another file) override later ones, and
22 % an environment variable foo overrides any texmf.cnf definition of foo.
23 %
24 % All definitions are read before anything is expanded, so you can use
25 % variables before they are defined.
26 %
27 % If a variable assignment is qualified with '.PROGRAM', it is ignored
28 % unless the current executable (last filename component of argv[0]) is
29 % named PROGRAM. This foo.PROGRAM construct is not recognized on the
30 % right-hand side. For environment variables, use FOO_PROGRAM.
31 %
32 % Which file formats use which paths for searches is described in the
33 % various programs' and the kpathsea documentation.
34 %
35 % // means to search subdirectories (recursively).
36 % A leading !! means to look only in the ls-R db, never on the disk.
37 % A leading/trailing/doubled : in the paths will be expanded into the
38 % compile-time default. Probably not what you want.
39
40 % Part 1: Search paths and directories.
41
42 % The root of everything below.
43 prefix = $SELFAUTOPARENT
44
45 % You can set an environment variable to override this if you're testing
46 % a new TeX tree, without changing anything else.
47 %
48 % You may wish to use one of the $SELFAUTO... variables here so TeX will
49 % find where to look dynamically. See the manual and the definition
50 % below of TEXMFCNF.
51 %
52 % If you have multiple trees, you can use shell brace notation, for example:
53 % TEXMF = {/usr/local/mytex,/usr/local/othertext}

```

```

54 % and also set TEXMFDBS to /usr/local/mytex:/usr/local/othertex.
55 TEXMFLOCAL=/usr/local/texmf
56 TEXMFMAIN = $prefix/share/texmf
57 TEXMFLS_R = $TEXMFLOCAL
58 % Where to look for ls-R files. There need not be an ls-R in the
59 % directories in this path, but if there is one, Kpathsea will use it.
60 TEXMFDBS = $TEXMFLOCAL;$TEXMFMAIN
61
62 % Where you want generated files to go. Choose one of the texmf trees
63 % listed in $TEXMF. The following is the default:
64 VARTEXMF = $TEXMFLOCAL
65
66 % The TeX inputs and fonts directories.
67 texdir = {$TEXMFLOCAL/tex,!!$TEXMFMAIN/tex}
68 omegadir = {$TEXMFLOCAL/omega,!!$TEXMFMAIN/omega}
69 fontdir = {$TEXMFLOCAL/fonts,!!$TEXMFMAIN/fonts}
70 dbmain=!!$TEXMFMAIN
71 %
72 dbtex = $texdir
73 dbomega = $omegadir
74 dbfonts = $fontdir
75
76 % TeX input files -- i.e., anything to be found by \input or \openin,
77 % including .sty, .eps, etc.
78 TEXINPUTS = .;$dbtex//
79
80 % LaTeX 2e specific macros are stored in latex.
81 % latex209 is not supported, at the request of the authors of LaTeX
82 %
83 latex_inputs = .;$dbtex/latex//;$dbtex/generic//
84 TEXINPUTS.latex = $latex_inputs
85 TEXINPUTS.latextex = $latex_inputs
86
87 % Omega
88 TEXINPUTS.lambda2e = .;$dbomega/lambda//;$latex_inputs
89 TEXINPUTS.lambda = .;$dbomega/lambda//;$latex_inputs
90
91 % Fontinst needs to read afm files.
92 TEXINPUTS.fontinst = .;$dbtex//;$dbfonts/afm//
93
94 % Plain TeX. Have the command tex check all directories as a last
95 % resort, we may have plain-compatible stuff anywhere.
96 plain_inputs = $dbtex/plain//;$dbtex/generic//
97 TEXINPUTS.textex = .;$dbtex//;$plain_inputs
98 TEXINPUTS.tex = .;$plain_inputs;$dbtex//
99 TEXINPUTS.omega = .;$dbomega//;$plain_inputs;$dbtex//
100
101 % INITEX. May as well make this separate so it can search on disk;
102 % initex is seldom run, and might be used directly after files have been
103 % added, when ls-R may not up be to date.
104 TEXINPUTS.initex = .;$texdir//

```



```

105 TEXINPUTS.iniomega = .;$omegadir//;$texdir//
106
107 % Earlier entries override later ones, so put this last.
108 TEXINPUTS.viomega = .;$dbomega//;$dbtex//
109 TEXINPUTS = .;$dbtex//
110
111 % Metafont, MetaPost inputs.
112 MFINPUTS=.;{$TEXMFLOCAL,$dbmain}/metafont//;{$dbfonts,$VARTEXFONTS}/source//
113 MPINPUTS=.;{$TEXMFLOCAL,$dbmain}/metapost//
114
115 % Dump files (fmt/base/mem) for vir{tex,mf,mp} to read (see
116 % web2c/INSTALL), and string pools (.pool) for ini{tex,mf,mp}. It is
117 % silly that we have six paths and directories here (they all resolve to
118 % a single place by default), but historically ...
119 TEXFORMATS = .;{$TEXMFLOCAL,$dbmain}/web2c
120 MFBASES = .;{$TEXMFLOCAL,$dbmain}/web2c
121 MPMEMS = .;{$TEXMFLOCAL,$dbmain}/web2c
122 TEXPOOL = .;{$TEXMFLOCAL,$dbmain}/web2c
123 MFPOOL = .;{$TEXMFLOCAL,$dbmain}/web2c
124 MPPOOL = .;{$TEXMFLOCAL,$dbmain}/web2c
125
126 % If you have a read-only central font directory and therefore need to
127 % cache MakeTeXPK-created fonts locally, set this as appropriate and
128 % also set the 'vartexfonts' feature in MT_FEATURES in MakeTeXnames.cnf.
129 VARTEXFONTS = /var/tex/fonts
130
131 % Device-independent font metric files.
132 VFFONTS = .;$dbfonts/vf//
133 TFMFONTS = .;$dbfonts/tfm//;$VARTEXFONTS/tfm//
134
135 % The $MAKETEX_MODE below means the drivers will not use a cx font when
136 % the mode is ricoh. If no mode is explicitly specified, kpse_prog_init
137 % sets MAKETEX_MODE to /, so all subdirectories are searched.
138 % See the manual.
139 PKFONTS = .;{$dbfonts,$VARTEXFONTS}/pk/$MAKETEX_MODE//
140
141 % xdvi needs to find bitmaps for PostScript fonts, which can be
142 % generated by the standalone program gsftopk, among others.
143 PKFONTS.xdvi = .;{$dbfonts,$VARTEXFONTS}/pk/{$MAKETEX_MODE,modelless}//
144
145 % Similarly for the GF format, which only remains in existence because
146 % Metafont outputs it (and MF isn't going to change).
147 GFFONTS = .;$dbfonts/gf/$MAKETEX_MODE//
148
149 % A backup for PKFONTS and GFFONTS. Not used for anything.
150 GLYPHFONTS = .;$dbfonts
151
152 % For texfonts.map and included map files used by MakeTeXPK.
153 % See ftp://ftp.tug.org/tex/fontname.tar.gz.
154 TEXFONTMAPS = .;{$TEXMFLOCAL,$dbmain}/fontname
155

```

```

156 % BibTeX bibliographies and style files.
157 BIBINPUTS = .;{$TEMFLOCAL,$dbmain}/bibtex/bib//
158 BSTINPUTS = .;{$TEMFLOCAL,$dbmain}/bibtex/bst//
159
160 % MFT style files.
161 MFTINPUTS = .;{$TEMFLOCAL,$dbmain}/mft//
162
163 % PostScript headers, prologues (.pro), encodings (.enc) and fonts.
164 TEXPSHEADERS = .;$TEMFLOCAL/dvips;$dbmain/dvips//;$dbfonts/type1//
165
166 % PostScript Type 1 outline fonts.
167 T1FONTS = .;$dbfonts/type1//;$TEMFLOCAL/dvips//;$dbmain/dvips//
168
169 % PostScript AFM metric files.
170 AFMFONTS = .;$dbfonts/afm//
171
172 % Dvips' config.* files (this name should not start with 'TEX!').
173 TEXCONFIG = .;$TEMFLOCAL/dvips;$dbmain/dvips//
174
175 % Makeindex style (.ist) files.
176 INDEXSTYLE = .;{$TEMFLOCAL,$dbmain}/makeindex//
177
178 % Used by DMP (ditroff-to-mpx), called by makempx -troff.
179 TRFONTS = /usr/lib/font/devpost
180 MPSUPPORT = .;{$TEMFLOCAL,$dbmain}/metapost/support
181
182 % For xdvi to find mime.types and .mailcap, if they do not exist in
183 % $HOME. These are single directories, not paths.
184 % (But the default mime.types, at least, may well suffice.)
185 MIMELIBDIR = $prefix/etc
186 MAILCAPLIBDIR = $prefix/etc
187
188 % TeX documentation and source files, for use with kpsewhich.
189 TEXDOCS = .;{$TEMFLOCAL,$dbmain}/doc//
190 TEXSOURCES = .;{$TEMFLOCAL,$dbmain}/source//
191
192 % Omega-related fonts and other files.
193 OFMFONTS = .;{$dbfonts,$VARTEXFONTS}//ofm//
194 OPLFONTS = .;{$dbfonts,$VARTEXFONTS}//opl//
195 OVFFONTS = .;{$dbfonts,$VARTEXFONTS}//ovf//
196 OVPFONTS = .;{$dbfonts,$VARTEXFONTS}//ovp//
197 OTPINPUTS = .;{$TEMFLOCAL,$dbmain}/omega/otp//
198 OCPINPUTS = .;{$TEMFLOCAL,$dbmain}/omega/ocp//
199
200 % For MakeTeX.common, MakeTeX.site, ls-R.
201 web2cdir = {$TEMFLOCAL,$dbmain}/web2c
202 TEXMFCNF_DIR = $TEMFMAIN/web2c
203
204 % The MakeTeX* scripts rely on KPSE_DOT. Do not change it.
205 KPSE_DOT = .
206

```

```

207 % This definition isn't used from this .cnf file itself (that would be
208 % paradoxical), but the compile-time default in paths.h is built from it.
209 % The SELFAUTO* variables are set automatically from the location of
210 % argv[0], in kpse_set_programname.
211 %
212 % About the /. construction;
213 % 1) if the variable is undefined, we'd otherwise have an empty path
214 % element in the compile-time path. This is not meaningful.
215 % 2) if we used /$VARIABLE, we'd end up with // if VARIABLE is defined,
216 % which would search the entire world.
217 %
218 % The TETEXDIR stuff isn't likely to relevant unless you're using teTeX,
219 % but it doesn't hurt.
220 %
221 TEXMFCNF = .;$SELFAUTOLOC;$SELFAUTODIR;$SELFAUTODIR/share/texmf/web2c;\
222 $SELFAUTOPARENT;$SELFAUTOPARENT/share/texmf/web2c;/. $TETEXDIR;\
223 /.{$TEXMFLOCAL,$dbmain}/web2c;$web2cdir
224
225 % Part 2; Non-path options.
226
227 % Write .log/.dvi/etc. files here, if the current directory is unwritable.
228 % TEXMFOUTPUT = /tmp
229
230 % If a dynamic file creation fails, log the command to this file, in
231 % either the current directory or TEXMFOUTPUT. Set to the
232 % empty string or 0 to avoid logging.
233 MISSFONT_LOG = missfont.log
234
235 % Set to a colon-separated list of words specifying warnings to suppress.
236 % To suppress everything, use TEX_HUSH = all; this is equivalent to
237 % TEX_HUSH = checksum;lostchar;readable;special
238 TEX_HUSH = 0
239
240 % Enable system commands via \write18{...}?
241 shell_escape = 0
242
243 % Allow TeX \openout on filenames starting with '.' (e.g., .rhosts)?
244 openout_any = 0
245
246 % Enable the MakeTeX... scripts by default? These must be set to 0 or 1.
247 % Particular programs can and do override these settings, for example
248 % dvips's -M option. Your first chance to specify whether the scripts
249 % are invoked by default is at configure time.
250 %
251 % These values are ignored if the script names are changed; e.g., if you
252 % set DVIPSMAKEPK to 'foo', what counts is the value of the environment
253 % variable/config value 'FOO', not the 'MAKETEXPK' value.
254 %
255 % MAKETEXTEX = 0
256 % MAKETEXPK = 0
257 % MAKETEXMF = 0

```

```

258 % MAKETEXTFM = 0
259
260 % What MetaPost runs to make MPX files. This is passed an option -troff
261 % if MP is in troff mode. Set to '0' to disable this feature.
262 MPXCOMMAND = makempx
263
264 % Part 3; Array and other sizes for TeX (and Metafont and MetaPost).
265 %
266 % If you want to change some of these sizes only for a certain TeX
267 % variant, the usual dot notation works, e.g.,
268 % main_memory.hugetex = 20000000
269 %
270 % If a change here appears to be ignored, try redumping the format file.
271
272 % Memory. Must be less than 8,000,000.
273 %
274 % main_memory is relevant only to initex, extra_mem_* only to non-ini.
275 % Thus, have to redump the .fmt file after changing main_memory; to add
276 % to existing fmt files, increase the other. (To get an idea of how
277 % much, try \tracingstats=2 in your TeX source file;
278 % web2c/tests/memtest.tex might also be interesting.)
279 %
280 % To increase space for boxes (as might be needed by, e.g., PiCTeX),
281 % increase extra_mem_bot.
282 %
283 main_memory = 263000 % words of memory available; also applies to mf&mp
284 main_memory.hugetex = 1100000
285 extra_mem_top = 0 % extra high memory for chars, tokens, etc.
286 extra_mem_bot = 0 % extra low memory for boxes, glue, breakpoints, etc.
287
288 % Words of font info for TeX (total size of all TFM files, approximately).
289 font_mem_size = 100000
290 font_mem_size.hugetex = 400000
291 font_mem_size.pdftex = 400000
292
293 % Total number of fonts. Must be >= 50 and <= 2000 (without tex.ch changes).
294 font_max = 500
295 font_max.hugetex = 900
296
297 % Extra space for the hash table of control sequences (which allows 10K
298 % names as distributed).
299 hash_extra = 0
300 hash_extra.hugetex = 10000
301 hash_extra.pdftex = 10000
302
303 % Max number of characters in all strings, including all error messages,
304 % help texts, font names, control sequences.
305 % These values apply to TeX and MP.
306 pool_size = 125000
307 pool_size.hugetex = 500000
308 pool_size.pdftex = 500000

```

```

309 % Minimum pool space after TeX/MP's own strings; must be at least
310 % 25000 less than pool_size, but doesn't need to be nearly that large.
311 string_vacancies = 25000
312 string_vacancies.hugetex = 45000
313 string_vacancies.pdfTEX = 45000
314 max_strings = 15000          % max number of strings
315 max_strings.hugetex = 55000      % max number of strings
316 max_strings.pdfTEX = 55000      % max number of strings
317 pool_free = 5000            % min pool space left after loading .fmt
318
319 % Hyphenation trie. As distributed, the maximum is 65535; this should
320 % work unless 'unsigned short' is not supported or is smaller than 16
321 % bits. This value should suffice for UK English, US English, French,
322 % and German (for example). To increase, you must change
323 % 'ssup_trie_opcode' and 'ssup_trie_size' in tex.ch (and rebuild TeX);
324 % the trie will then consume four bytes per entry, instead of two.
325 %
326 % US English, German, and Portuguese; 30000.
327 % German; 14000.
328 % US English; 10000.
329 %
330 trie_size = 64000
331
332 hyph_size = 1000            % number of hyphenation exceptions, >610 and <32767.
333 buf_size = 3000            % max length of input lines or control sequence names
334 nest_size = 100           % simultaneous semantic levels (e.g., groups)
335 max_in_open = 15          % simultaneous input files and error insertions
336 param_size = 500          % simultaneous macro parameters
337 save_size = 4000          % for saving values outside current group
338 save_size.pdfTEX = 30000    % for saving values outside current group
339 save_size.hugetex = 30000    % for saving values outside current group
340 stack_size = 300          % simultaneous input sources
341
342 % These work best if they are the same as the I/O buffer size, but it
343 % doesn't matter much. Must be a multiple of 8.
344 dvi_buf_size = 16384 % TeX
345 gf_buf_size = 16384 % MF
346
347 % It's probably inadvisable to change these. At any rate, we must have:
348 % 45 < error_line < 255;
349 % 30 < half_error_line < error_line - 15;
350 % max_print_line < 60 ;
351 % These apply to Metafont and MetaPost as well.
352 error_line = 79
353 half_error_line = 50
354 max_print_line = 79

```

## 7 Catalogue of Packages

Table 1: **T<sub>E</sub>X Live** packages

<i>Package</i>	<i>Collection</i>	<i>Description</i>
a4	latex3	Originally for L <sup>A</sup> T <sub>E</sub> X 2.09 but updated for L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> . Mostly superseded by L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> support for a4 paper but defines the extra option of widemargins. The geometry package is usually better.
aiaa	latex3	A bundle of L <sup>A</sup> T <sub>E</sub> X/BIB <sub>T</sub> E <sub>X</sub> files and sample documents to aid those producing papers and journal articles according to the guidelines of the American Institute of Aeronautics and Astronautics (AIAA)
achemso	latex3	L <sup>A</sup> T <sub>E</sub> X and BIB <sub>T</sub> E <sub>X</sub> style for American Chemical Society
acronym	latex3	This package ensures that all acronyms used in the text are spelled out in full at least once. It also provides an environment to build a list of acronyms.
adrlist	latex3	Using address lists in L <sup>A</sup> T <sub>E</sub> X.
aguplus	latex3	Styles for American Geophysical Union.
alalex	formats3	An extended L <sup>A</sup> T <sub>E</sub> X with better modularity
alg	latex3	L <sup>A</sup> T <sub>E</sub> X environments for typesetting algorithms
algorithms	latex3	Defines a floating algorithm environment designed to work with the algorithmic package.
alpha-linux	systems1	System binaries for Alpha running Linux.
alpha-osf3.2	systems1	System binaries for Alpha running OSF 3.2.
altfont	latex3	A generalised replacement for some parts of psnfss and mfnfss. Similar to psfont with the PostScript specific code removed.
amiweb2c	systems1	An Amiga port of the complete UNIX-TeX system.
amsfonts	ams2	A set of miscellaneous T <sub>E</sub> X fonts from the American Mathematical Society that augment the standard set normally distributed with T <sub>E</sub> X. The set includes: Extra mathematical symbols; Blackboard bold letters (uppercase only); Fraktur letters; Subscript sizes of bold math italic and bold Greek letters; Subscript sizes of large symbols such as sum and product; Added sizes of the Computer Modern small caps font; Cyrillic fonts (from the University of Washington); Euler math fonts.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
amslatex	ams2	A collection of loosely related files that are distributed together by the American Mathematical Society. These files are miscellaneous enhancements to L <sup>A</sup> T <sub>E</sub> X whose aim is superior information structure of mathematical documents and superior printed output.
amstex	ams2	American Mathematical Society plain T <sub>E</sub> X macros
answers	latex3	Styles for setting questions (or exercises) and answers.
apa	latex3	L <sup>A</sup> T <sub>E</sub> X class and BIBT <sub>E</sub> X style used to format text according to the American Psychological Association Publication Manual (4th ed.) specifications for manuscripts or, with an option to the package, in an APA journal style format or as a regular document.
apl	fonts3	Fonts for typesetting APL programs.
arabtex	lang3	Macros and fonts for typesetting Arabic
arseneau	latex2	Miscellaneous macros by Donald Arseneau.
ascii	fonts3	Support for IBM extended ASCII font.
astro	fonts3	Astronomical (planetary) symbols.
aurora	dvips3	Header files for dvips to make colour separations
autotab	latex3	Generating tabular setups.
babel	latex1	Multilingual support for L <sup>A</sup> T <sub>E</sub> X.
backgammon	fonts3	Style for typesetting backgammon boards.
bakoma	fonts2	Computer Modern and AMS fonts in PostScript Type1 form.
barcodes	fonts3	Fonts for making barcodes.
barr	graphics3	Diagram macros by Michael Barr.
base	latex1	Basic L <sup>A</sup> T <sub>E</sub> X system.
bbding	fonts3	An NFSS-interface to the symbol font bbding containing many of the Zapf dingbats fonts.
bbm	fonts3	Blackboard variant fonts for Computer Modern, with L <sup>A</sup> T <sub>E</sub> X support
bbtbase	bibtex1	Basic BIBT <sub>E</sub> X styles.
bbtdoc	bibtex2	Basic BIBT <sub>E</sub> X documentation/
beebe	bibtex2	Nelson Beebe's collection of T <sub>E</sub> X-related bibliographies, and BIBT <sub>E</sub> X style files
beton	latex3	Typeset a L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> document with the Concrete fonts designed by Don Knuth and used in his book "Concrete Mathematics".

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
biblist	latex3	BIBTEX styles by Joachim Schrod.
blue	formats3	Kees van der Laan's BLUE format, a concise but expressive document preparation system modelled on Knuth's manmac
bm	latex2	This is a package for accessing bold symbols in math mode. (Similar to the AMS <code>\boldsymbol</code> command, but taking more care over spacing, delimiters etc.).
booktabs	latex2	Nicer layout of tables
borceux	graphics3	Diagram macros by Francois Borceux
bridge	latex3	Macros for typesetting Bridge diagrams.
calc	latex2	Adds infix expressions to perform arithmetic in the L <sup>A</sup> T <sub>E</sub> X commands <code>\setcounter</code> , <code>\addtocounter</code> , <code>\setlength</code> , and <code>\addtolength</code> .
calendar	plain3	Plain macros for making nice calendars
calrsfs	latex3	Nicer calligraphic letters.
camel	latex3	Comprehensive bibliography manager (prototype citation engine for L <sup>A</sup> T <sub>E</sub> X3). Will become BIBTEX 1.0 on release. Under development.
caption	latex2	Extends caption capabilities for figures and tables, such as the caption width, style, font. Many aspects are tunable as options.
carlisle	latex2	Miscellaneous small packages by David Carlisle
ccfonts	latex	L <sup>A</sup> T <sub>E</sub> X support for Concrete fonts.
chess	fonts3	Macros and fonts for typesetting Chinese Chess board diagrams.
cellular	plain3	Cellular table construction
changebar	latex2	Generate changebars in L <sup>A</sup> T <sub>E</sub> X documents.
chemcono	latex3	A L <sup>A</sup> T <sub>E</sub> X style file for using compound numbers in chemistry documents. It works like <code>\cite</code> and the <code>\thebibliography</code> , using <code>\fcite</code> and <code>\theffbibliography</code> instead. It allows compound names in documents to be numbered and does not affect the normal citation routines.
chemsym	latex3	Macros for typing chemical symbols
cheq	fonts3	Adobe chess font.
cherokee	fonts3	Fonts for Cherokee scripts
chess	fonts3	Fonts for typesetting chess boards.
circ	graphics3	Macros for typesetting circuit diagrams. Several electrical symbols like resistor, capacitor, transistors etc., are defined. The symbols can be connected with wires.



Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
circuit	graphics3	Language for drawing circuit diagrams.
cirth	fonts3	Fonts for Cirth
cite	latex2	Supports compressed, sorted lists of numerical citations: [8,11–16].
cm	fonts1	Computer Modern fonts
cmbright	fonts2	Computer Modern Bright fonts
cmcyralt	latex3	Alternative Russian encoding support
cmextra	fonts2	Extra Computer Modern fonts, from the American Mathematical Society
cmpica	fonts3	A Computer Modern Pica variant
cmps	fonts2	Type1 versions of PostScript fonts, from Blue Sky and Y&Y.
codepage	latex3	Support for variant code pages.
colorsep	dvips3	Support for colour separation when using dvips
concmath	fonts3	Concrete math fonts derived from Computer Modern math fonts using parameters from Concrete text fonts. A $\LaTeX$ package providing the necessary font definition code is included.
concrete	fonts3	Concrete fonts
context	generic3	pphtex is a package that can be used to typeset chemical formulas. The package is a separate module of the context macro package for $\TeX$ (context is a full featured, parameter driven macro package, which fully supports advanced interactive documents).
count1to	latex3	Setting $\backslash\text{count1}$ to $\backslash\text{count9}$
croatian	lang3	Fonts for typesetting Croatian scripts
crosswrd	latex3	Macros for typesetting crossword puzzles.
csfonts	fonts2	Czech/Slovak-tuned METAFONT Computer Modern fonts.
cslatex	latex	$\LaTeX$ support for Czech/Slovak typesetting.
csplain	plain2	Plain $\TeX$ support for Czech/Slovak typesetting.
curves	graphics3	Draws curves in the $\LaTeX$ picture environment using parabolas between points with continuous slope at points. Equivalent to technical pens with compasses and French curves.
custom-bib	latex2	Package generating customized $\text{BIB}\TeX$ bibliography styles from a generic file using docstrip.
dancers	fonts3	Font for the Sherlock Holmes ‘Dancing Men’
dates	latex3	Macros for parsing date strings.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
deleg	latex3	Provides a more flexible numbering of equations, subequations, and ‘recycled’ equations, including ‘partial’ equation numbers (‘3a’, ‘3b’ etc.).
devanagari	lang3	Fonts for typesetting Devanagari
dialogl	latex3	Macros for constructing interactive L <sup>A</sup> T <sub>E</sub> X scripts
dinbrief	latex3	German letter DIN style.
doc1	texlive1	Basic documentation for <b>T<sub>E</sub>X Live</b> .
doc2	texlive2	Recommended documentation for <b>T<sub>E</sub>X Live</b> .
dotseqn	latex3	Flush left equations with dotted letters to the numbers
draftcopy	latex3	Places the word DRAFT (or other words) in light grey diagonally across the background (or at the bottom) of each (or selected) pages of the document.
dratex	graphics3	General drawing macros entirely in T <sub>E</sub> X
dropping	latex3	A L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> macro for dropping the first character(s) (or word(s)) of a paragraph. This is an extension of the L <sup>A</sup> T <sub>E</sub> X 2.09 package dropcaps. This package automatically takes care of finding the font name.
dtk	latex3	Macros for the DANTE publication.
duerer	fonts3	Computer Duerer fonts.
dvips	generic1	Tom Rokicki’s dvi to PostScript driver
dvipsbas	dvips1	Basic support files for dvips.
ean	generic3	Font for making EAN barcodes.
easy	latex3	Macros for simplifying the writing of equations.
ec	fonts2	The ec fonts support the complete L <sup>A</sup> T <sub>E</sub> X T1 encoding, as defined at the 1990 TUG conference hold at Cork/Ireland. They are intended to be as stable as the cm fonts are, i.e., there shall be no more changes to the tfm files. The ec fonts also contain a Text Companion Symbol font, called tc, featuring many useful characters needed in typesetting, for example oldstyle digits, currency symbols (including the newly created Euro symbol), the permille sign, copyright, trade mark and servicemark as well as a copyleft sign, and many others. Recent releases of L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> support the ec fonts. The dc fonts, which were termed as preliminary versions, will disappear from the archives.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
ecc	fonts3	‘European’ (T1 encoded) version of the Concrete fonts.
edmac	plain3	A macro package for typesetting scholarly critical editions
eepic	graphics2	A set of T <sub>E</sub> X macros for L <sup>A</sup> T <sub>E</sub> X implementing several extensions to EPIC and the L <sup>A</sup> T <sub>E</sub> X picture drawing environment, including the drawing of lines at any slope, the drawing of circles in any radii, and the drawing of dotted and dashed lines much faster with much less T <sub>E</sub> X memory, and providing several new commands for drawing ellipses, arcs, splines, and filled circles and ellipses.
eiad	latex3	Macros and EIAD fonts.
elsevier	latex3	Preprint style for Elsevier Science journals
elvish	fonts3	Font for typesetting Tolkien Elvish script
endfloat	latex3	Place all figures on pages by themselves at the end of the document with markers like “[Figure 3 about here]” appearing in the text (by default) near to where the figure (or table) would normally have occurred.
engwar	fonts3	Font for typesetting Tolkien Engwar script, by Michael Urban
enbvg	latex3	Printing addresses on envelopes
envlab	latex3	A L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> package for producing mailing envelopes and labels, including barcodes and address formatting according to the US Postal Service rules. Redefines the standard <code>\makeLabels</code> command of the L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> letter documentclass.
eplain	formats2	Simple but powerful extended version of the plain format, adding support for bibliographies, tables of contents, enumerated lists, verbatim input of files, numbered equations, tables, two-column output, footnotes and commutative diagrams.
epslatex	doc2	An extensive document which explains how to use Encapsulated PostScript (EPS) files in L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> documents. Includes explanations of Bounding Boxes, and more.
eqname	latex3	Style for different equation numbering.
eqnarray	latex3	More generalised equation arrays with numbering.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
euler	latex3	Provides a setup for using the AMS Euler family of fonts for math in $\LaTeX$ documents. “The underlying philosophy of Zapf’s Euler design was to capture the flavor of mathematics as it might be written by a mathematician with excellent handwriting.” [concrete-tug] The euler package is based on Knuth’s macros for the book “Concrete Mathematics”. The text fonts for the Concrete book are provided by the beton package.
everysel	latex3	$\LaTeX$ package which provides hooks into <code>\selectfont</code> .
everyshi	latex3	Hooks for taking action at every <code>\shipout</code>
exam	latex3	Package for typesetting exam scripts.
exams	latex3	Exam questions can be multiple choice or free form long/short answer questions. Options include the typesetting of the exam itself, an exam showing all the answers and a collection of questions and answers. Questions can be parametrized. Use of a random generator provides for automatic shuffling of multiple choice items.
expdlist	latex3	Expanded description environments
export	latex3	This package allows the user to export/import the values of $\LaTeX$ registers (counters, rigid and rubber lengths only). It is definitely NOT for faint-hearted users.
fancyhdr	latex3	Support for sophisticated control of page headers and footers in $\LaTeX 2\epsilon$ . It supersedes fancyheadings.
fancyheadings	latex2	Better control over page headers and footers in $\LaTeX$ . This is an up-to-date version for $\LaTeX$ . For $\LaTeX 2\epsilon$ it is now called fancyhdr.
fancyvrb	latex2	Sophisticated handling of verbatim text, to write it out, read it in, and typeset it
fax	latex3	Document class for preparing faxes.
fc	fonts3	Fonts for African languages, complementary to Computer Modern.
feynmf	graphics3	Macros and fonts for creating Feynman (and other) diagrams.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
float	latex2	Improves the interface for defining floating objects such as figures and tables. Introduces the boxed float and the ruled float. You can define your own floats and improve the behaviour of the old ones. Also incorporates the H option of the superseded here package. You can select this as automatic default with <code>\floatplacement{figure}{H}</code> .
floatfig	latex3	Allows text to be wrapped around figures.
floatflt	latex3	Float text around figures and tables which do not span the full width of a page. This is an improved version of floatfig. It is more or less similar to floatingtable. The tables/figures can be set left/right or alternating on even/odd pages. Works with the multicol package. Doesn't work well in the neighbourhood of list environments unless you change your L <sup>A</sup> T <sub>E</sub> X document.
fncychap	latex3	This package provides six predefined chapter headings. Each can be modified using a set of simple commands. Optionally one can modify the formatting routines in order to create additional chapter headings. This package was previously known as FancyChapter.
foiltex	latex3	A L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> class for overhead transparencies. Can be used with fancybox to place a variety of borders around the slides.
fontinst	latex3	T <sub>E</sub> X macros for converting Adobe Font Metric files to T <sub>E</sub> X metric and virtual font format
fontmisc	fonts1	Miscellaneous METAFONT input files
fontname	texlive1	Karl Berry's scheme for naming fonts in T <sub>E</sub> X
fonts	latex1	Extra line and circle fonts for L <sup>A</sup> T <sub>E</sub> X
footnote	latex3	More sophisticated footnotes.
footnpag	latex3	Allows footnotes on individual pages to be numbered from 1, rather than being numbered sequentially through the document.
formats	texlive2	Prebuilt T <sub>E</sub> X format and METAFONT base files
fp	latex3	Provides an extensive collection of arithmetic operations for fixed point real numbers of high precision.
french	lang2	Style for French typography
fribrief	latex3	A L <sup>A</sup> T <sub>E</sub> X class for writing letters.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
fundus	latex3	Providing L <sup>A</sup> T <sub>E</sub> X access to various font families.
futhark	fonts3	Fonts for the Older Futhark script
g-brief	latex3	A document class for L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> . Serves for formatting formless letters in german or english language.
gb4e	latex3	Government Binding styles.
general	doc1	Useful general documentation.
genmisc	generic3	Miscellaneous small files for all formats, specific to the <b>T<sub>E</sub>X Live</b> CD-ROM.
geometry	latex3	A package which allows L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> users to customise page layout (page sizes) using an easy and flexible user interface. You can specify <code>\geometry{body={6.5in,8.75in}, top=1.2in, left=2cm, nohead}</code> . This is an update of the now superseded pagesize package.
german	latex2	Style for German typography.
germbib	bibtex2	German variants of standard BIB <sub>T</sub> E <sub>X</sub> styles.
go	fonts3	Fonts and macros for typesetting go games.
gothic	fonts3	Gothic and ornamental initial fonts by Yannis Haralambous.
graphics	latex2	The primary L <sup>A</sup> T <sub>E</sub> X package for the support of the inclusion of graphics generally produced with other tools. This package aims to give a consistent interface to including the file types that are understood by your printer driver.
gray	fonts3	Fonts for gray scales
hands	fonts3	Pointing hand fonts
harpoon	latex3	Extra harpoons, using the graphics package
harvard	bibtex2	The Harvard bibliography style family.
harvmac	plain3	Paul Ginsparg's Harvard macros for scientific articles
hh	latex3	Fancy boxing effects
histogr	latex3	Drawing histograms with the L <sup>A</sup> T <sub>E</sub> X picture environment.
hppa1.1-hpux10.20	systems1	System binaries HP running hpux10.20.
hppa1.1-hpux9.05	systems1	System binaries HP running hpux9.05.
html	doc2	Various T <sub>E</sub> X documentation converted to HTML
hyper	latex3	Redefines L <sup>A</sup> T <sub>E</sub> X cross-referencing commands to insert <code>\special</code> commands for HyperTeX dvi viewers

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
hyperref	latex3	Redefines L <sup>A</sup> T <sub>E</sub> X cross-referencing commands to insert <code>\special</code> commands for HyperTeX dvi viewers, or translation to Acrobat.
hyphen	lang1	Collection of hyphenation patterns.
i386-linux	systems1	System binaries for Intel machines running Linux.
i586-freebsd2.2	systems1	System binaries for Intel PC running FreeBSD 2.2.
i686-linux	systems1	System binaries for Intel Pentium Pro running Linux.
ieeepes	latex3	Allows typesetting of transactions, as well as discussions and closures, for the IEEE Power Engineering Society Transactions journals.
ifacmtg	latex3	Elsevier Science preprint style for IFAC meetings.
indxcite	latex3	A package to automatically generate an Author Index based on citations made using B <sub>I</sub> B <sub>T</sub> E <sub>X</sub> . It requires the use of the harvard and index packages and L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> .
info	doc2	Documentation in GNU info form
inputenc	latex3	Controlling input encoding
ipa	latex3	Style for using International Phonetic Alphabet fonts
isostds	latex3	Class and package files for typesetting ISO International Standard documents. Several standard documents have been printed by ISO from camera-ready copy prepared using L <sup>A</sup> T <sub>E</sub> X and these files. One set of files is for generic ISO typesetting and the other is an extension set of packages for typesetting ISO 10303 standards.
jadetex	latex3	Macro package on top of L <sup>A</sup> T <sub>E</sub> X to typeset T <sub>E</sub> X output of Jade DSSSL implementation.
jknappen	latex2	Miscellaneous macros, mostly for making use of extra fonts, by Jörg Knappen.
jsmisc	plain3	Miscellaneous useful macros by Joachim Schrod.
jura	latex3	A document class for German law students.
knuth	doc3	Knuth's own documentation, including the T <sub>E</sub> Xbook and the METAFONTbook
koma-script	latex3	A replacement for the article/report/book classes with emphasis on European rules of typography and paper formats as laid down by Jan Tschichold.
kuvio	graphics3	Drawing macros and fonts for diagrams.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
labels	latex3	Support for printing sheets of sticky labels (but could also be used for business cards). The number of rows and columns of labels, and their size, can be changed.
lamstex	formats3	A merge of the best in AMST <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X
lastpage	latex3	Reference the number of pages in your L <sup>A</sup> T <sub>E</sub> X document (as in a page footer that says: Page N of M).
lgreek	latex3	Macros for using Silvio Levy's Greek fonts
lineno	latex3	Adds line numbers to selected paragraphs with reference possible through the L <sup>A</sup> T <sub>E</sub> X <code>\ref</code> and <code>\pageref</code> cross reference mechanism. Version 2.00 supports numbering of one in five lines and switching the line numbers from the left to the right side of the page in twoside mode.
listings	latex3	Package for pretty-printed program listings, with support for a range of languages.
localloc	latex3	Macros for localizing T <sub>E</sub> X register allocations
logic	fonts3	METAFONT font for drawing logic diagrams.
lollipop	formats3	A new generation format
lshort	latex1	Short introduction to L <sup>A</sup> T <sub>E</sub> X.
ltablex	latex3	Modifies the tabularx environment to combine the features of the tabularx package (auto-sized columns in a fixed width table) with those of the longtable package (multi-page tables).
ltxdoc	latex1	Class for documented L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> classes.
ltxmisc	latex2	Miscellaneous L <sup>A</sup> T <sub>E</sub> X styles.
lw35ps	fonts2	Font metrics and L <sup>A</sup> T <sub>E</sub> X font description files for standard 35 PostScript fonts.
ly1	latex3	Support for LY1 L <sup>A</sup> T <sub>E</sub> X encoding, i.e. the Y&Y texnansi encoding.
mab-nextstep3	systems1	System binaries for Next boxes running NextStep 3.
mailing	latex3	Macros for mail merging
makeindex	texlive1	Documentation for the MakeIndex program.
malvern	fonts3	A new sans-serif font family
mapcodes	latex3	Support for multiple character sets and encodings.
maple	latex3	Styles and examples for the MAPLE newsletter.
mathcomp	latex2	A package which provides access to some interesting characters of the Text Companion fonts (TS1 encoding) in math mode.



Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
mcite	latex3	Support for collapsing multiple citations into one, as customary in physics journals
mdwtools	latex3	Miscellaneous tools by Mark Wooding, including support for $\mathcal{O}$ , a doafter command, footnotes, mathenv for various alignment in maths, list handling, trivial maths oddments, rewrite of $\LaTeX$ 's <code>tabular</code> and <code>array</code> environments, verbatim handling, and syntax diagrams.
mff	latex3	A package to provide something similar to 'multiple master' fonts, but using METAFONT; you specify a font by a set of METAFONT parameters, and $\TeX$ makes up a .mf file to generate the required font; this package is not integrated with NFSS (or MakeTeXTFM) yet fun.
mflogo	latex1	$\LaTeX$ package and font definition file to access the Knuthian 'logo' fonts described in 'The METAFONTbook' and the METAFONT and MetaPost logos in $\LaTeX$ documents.
mfmisc	fonts1	Small support files for METAFONT.
mfnfss	latex3	Font description files for extra fonts like yinit and ygoth
mfpic	graphics3	Macros which generate METAFONT code for drawing pictures.
mftoops	fonts3	a METAFONT package for generating (encapsulated PostScript) files readable by CorelDRAW!, Adobe Illustrator and Fontographer. METAFONT writes PostScript code to a LOG-file, and from the LOG-file the code can be extracted by either $\TeX$ or AWK. DOS batch files, $\TeX$ source, and AWK source are placed in the subdirectory PROGS.
midnight	generic3	A set of useful macro tools
minitoc	latex3	Table of contents per chapter.
mips-irix4.0.5	systems1	System binaries for SGI box running Irix 4.0.5.
mips-irix5.3	systems1	System binaries for SGI box running Irix 5.3.
mips-irix6.3	systems1	System binaries for SGI box running Irix 6.3.
mips-ultrix4.4	systems1	System binaries for MIPS machine running Ultrix 4.4.
misc	latex3	Miscellaneous small macro files for $\LaTeX$ .
misc209	latex2	Miscellaneous small macro files for $\LaTeX$ 2.09.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
mnras	plain3	Styles for the Monthly Notices of the Royal Astronomical Society.
monster	latex3	Towards a more rational and modular L <sup>A</sup> T <sub>E</sub> X, by Matt Swift; a set of powerful tools
moreverb	latex3	A verbatim mode that can handle TABs properly, can number lines, can number lines in an included file, can produce boxed verbatims, etc.
mpbase	metapost1	Basic MetaPost support files.
mpfnmark	latex3	A package which provides the command <code>\mpfootnotemark</code> , which can be used in the same way as <code>\footnotemark</code> . The difference between these two macros is that within minipage environments the latter uses the standard footnote marker style (defined by <code>\thefootnote</code> ), while the new command uses the minipage footnote marker style (defined by <code>\thempfootnote</code> ).
mslapa	latex3	L <sup>A</sup> T <sub>E</sub> X and BIB <sub>T</sub> E <sub>X</sub> style files for a respectably close approximation to APA (American Psychological Association) citation and reference style.
mtbe	plain3	Examples from <i>Mathematical T<sub>E</sub>X by Example</i> by Arvind Borde
musictex	generic3	Typesetting music with T <sub>E</sub> X.
musixtex	generic3	Extended MusicT <sub>E</sub> X, with better slurs
myletter	latex3	Another letter package.
nassflow	latex3	Drawing Nassi-Schneidermann diagrams.
natbib	bibtex2	A bibliography style that handles author-year and numbered references.
newalg	latex3	Format algorithms like Cormen, Leiserson and Rivest.
newsletr	plain3	Macros for making newsletters
newthm	latex3	A modified version of the theorem-style which provides generation of lists of theorems.
niceframe	latex3	Support for fancy framing of pages
nomenc1	latex3	Nomenclature package for producing lists of symbols using the capabilities of the MakeIndex program.
ntgclass	latex2	Versions of the standard L <sup>A</sup> T <sub>E</sub> X article and report classes, rewritten to reflect a more European design, by the Dutch T <sub>E</sub> X Users Group

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
numline	latex3	Macros for numbering lines.
objectz	latex3	Macros for typesetting Object Z
oca	fonts3	OCR font
ocr-a	fonts3	Fonts for OCR-A
ocr-b	fonts3	Fonts for OCR-A
ogham	fonts3	Fonts for typesetting Ogham script
ogonek	latex3	Support for Polish typography and the ogonek
oldstyle	latex3	Font information needed to load the <code>cmmi</code> and <code>cmmb</code> fonts for use to produce oldstyle numbers
osmanian	fonts3	Osmanian fonts by Alan Stanier for writing Somali
ot2cyr	fonts2	Macros to use to the OT2 Cyrillic encoding
overword	latex3	This package provides two macros which can be used as building blocks for the parsing of text. For an example of their use, see the calendar package.
pandora	fonts3	The Pandora font family
paper	latex3	A class derived from <code>article</code> , tuned for producing papers for journals. Introduces new layout options and font commands for sections/parts. Defines a new <code>keywords</code> environment, and subtitle and institution commands for the title section. New commands for revisions. And more.
parallel	latex3	Provides a parallel environment which allows two columns of text to be typeset. Useful for typesetting two languages side-by-side.
patch	latex3	Macros for package management.
pb-diagram	latex3	Diagram package, using LAMST $\TeX$ fonts, by Paul Burchard.
pdcmac	plain3	Damian Cugley's macro tools.
pdf	doc2	Documentation in PDF format.
pdftex	plain2	Macro packages for variant $\TeX$ which writes PDF format.
phonetic	fonts3	METAFONT Phonetic fonts, based on Computer Modern
physe	formats3	PHYSE format
phyzzx	formats3	A $\TeX$ format for physicists
picipar	latex3	Insert pictures into paragraphs. (NOTE: Piet van Oostrum does not recommend this package. Picins is recommended instead.)
pictex	graphics2	Picture drawing macros for L $\TeX$ .

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
piff	latex3	Macro tools by Mike Piff
plainmisc	plain1	Miscellaneous useful macros for plain $\TeX$
plaintex	plain1	Basic Plain $\TeX$ macros
plfonts	fonts3	Computer Modern variant fonts for Polish
plgraph	generic3	$\LaTeX$ graphics package with wrapper to allow it to be used with generic plain $\TeX$
pmgraph	latex3	A set of extensions to $\LaTeX$ picture environment, including a wider range of vectors, and a lot more box frame styles.
poligraf	generic3	Page preparation for prepress, color separation, crop-marks, color and gray scale bars, booklet preparation, etc.
prelim2e	latex3	Allows the marking of preliminary versions of a document
prettyref	latex3	Additional functionality to $\LaTeX 2_{\epsilon}$ label-reference mechanism. It allows the author to “preformat” all types of labels.
progkeys	latex3	The file ‘programs.sty’ is intended to allow a parameterized way of typesetting programs with $\TeX$ / $\LaTeX$ commands inside. The file ‘keywords.sty’ allows use and define sets of keywords that will be typeset with different fonts, according to the wish of the user.
program	latex3	Typesetting programs and algorithms
proofs	latex3	Macros for building proof trees.
psfonts	fonts2	PostScript fonts for use with $\TeX$ and $\LaTeX$ .
psfrag	graphics2	A set of macros and a PostScript header which allows $\LaTeX$ constructions (equations, picture environments, etc.) to be precisely superimposed over Encapsulated PostScript figures. The user can use his/her favorite drawing tool to create an EPS figure, placing simple text “tags” where each replacement is to be placed. PSfrag will automatically remove those tags from the figure and replace them with the $\LaTeX$ construction that the user specifies, properly aligned, scaled, and/or rotated.
psizzl	formats3	A $\TeX$ format from SLAC

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
pslatex	latex2	A small package that makes L <sup>A</sup> T <sub>E</sub> X default to ‘standard’ PostScript fonts. It is basically a merger of the times and mathptm styles from the psnfss suite of packages. You must have installed standard L <sup>A</sup> T <sub>E</sub> X and PSNFSS PostScript fonts to use this package. The main novel feature is that the pslatex package tries to compensate for the visual differences between the Adobe fonts by scaling Helvetica by 90%, and ‘condensing’ Courier (i.e. scaling horizontally) by 85%. The package is supplied with a (unix) shell file for a ‘pslatex’ command that allows standard L <sup>A</sup> T <sub>E</sub> X documents to be processed, without needing to edit the file.
psnfss	fonts2	Font definition files, macros and font metrics for common PostScript fonts
psnfssx	latex3	Extra styles and encodings for PS fonts, including Y&Y encoding support.
pspicture	latex2	Replacement for core L <sup>A</sup> T <sub>E</sub> X picture macros to use PostScript <code>\special</code> commands
pstricks	graphics2	An extensive collection of PostScript macros that is compatible with most T <sub>E</sub> X macro packages, including Plain T <sub>E</sub> X, L <sup>A</sup> T <sub>E</sub> X, AMS-T <sub>E</sub> X, and AMS-L <sup>A</sup> T <sub>E</sub> X. Included are macros for color, graphics, pie charts, rotation, trees and overlays. It has many special features, including: a wide variety of graphics (picture drawing) macros, with a flexible interface and with color support. There are macros for coloring or shading the cells of tables.
punk	fonts3	Donald Knuth’s punk font
qobitree	graphics3	L <sup>A</sup> T <sub>E</sub> X macros for typesetting trees
qsymbols	latex3	For defining systematic mnemonic abbreviations, starting with ‘ for math symbols and \” for arrows, from the amssymb and stmaryrd packages
ragged2e	latex3	L <sup>A</sup> T <sub>E</sub> X package which defines new commands <code>\Centering</code> , <code>\RaggedLeft</code> , and <code>\RaggedRight</code> and new environments <code>Center</code> , <code>FlushLeft</code> , and <code>FlushRight</code> , which set ragged text and are easily configurable to allow hyphenation.
rcs	latex3	Use RCS (revision control system) tags in L <sup>A</sup> T <sub>E</sub> X documents.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
realcalc	plain3	Macros for real arithmetic calculations.
refman	latex3	Variant report and article styles
revtex	latex2	Styles for American Physical Society, American Institute of Physics, and Optical Society of America. Only works in compatibility mode under L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> .
rlepsz	generic3	A macro package for use with epsf.tex which allows PostScript labels in an eps file to be replaced by T <sub>E</sub> X labels.
rotating	latex2	A package built on the standard L <sup>A</sup> T <sub>E</sub> X graphics package to perform all the different sorts of rotation one might like, including complete figures and tables and captions.
rotfloat	latex3	Rotate floats.
rplain	latex3	Redefines the ‘plain’ pagestyle. The page numbers are now in the lower right corner.
rs6000-aix3.2.5	systems1	System binaries for RS6000 running AIX 3.2.5.
rs6000-aix4.1.1	systems1	System binaries for RS6000 running AIX 4.1.1.
sauter	fonts3	Extensions to the CM fonts, providing a parameterization scheme to build fonts at true design sizes
scale	latex3	A package to scale a document by $\sqrt{2}$ . This is useful if you are preparing a document on e.g. A5 paper and want to print on A4 paper to achieve a better resolution.
script	latex3	Variant report / book styles
semantic	latex3	Eases the typesetting of notation of semantics and compilers. Includes T-diagrams, various derivation symbols and inference threes.
seminar	latex2	Produce overhead slides (transparencies) with bells and whistles.
setspace	latex3	Provides commands and environments for doing double and one-and-a-half line spacing based on pt size. If a different spacing is required then the <code>\setstretch{baselinestretch}</code> command is supported. The spacing environment takes one argument which is the baselinestretch to use, e.g., <code>\begin{spacing}{2.5}</code> .
shadbox	latex3	A tool to shade the background of any box – text, figure, table etc. – using Plain (L <sup>A</sup> )T <sub>E</sub> X.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
shadethm	latex3	Package that allows declarations of the form <code>\newshadetheorem{thm}{Theorem}</code> or <code>\newshadetheorem{}[]{} or \newshadetheorem{}-{}[]</code> to produce shaded boxes from the usual command <code>\begin{theorem} ... \end{theorem}</code> . The color package is required.
showlabels	latex3	Show label commands in the margin.
siam	generic3	Styles for SIAM publications
siggraph	latex3	Document class for formatting papers according to the specifications for submission to the annual ACM Siggraph conference
slidenotes	latex3	A class package for the easy production of a slide collection with annotations. Builds on the report style (or variants).
smallcap	latex3	Support for all 4 shapes of Small caps in DC1.3 where SC becomes a family, rather than a shape ( <code>\scshape</code> is replaced by <code>\scfamily</code> ). Thus you can write <code>\bf\scfamily\slshape</code> to get small caps bold slanted.
songbook	latex3	Package for typesetting song lyrics.
sparc-linux	systems1	System binaries for Sparc running Linux.
sparc-solaris2.4	systems1	System binaries for Sparc running Solaris 2.4.
sparc-solaris2.5	systems1	System binaries for Sun Sparc running Solaris 2.5.
sparc-sunos4.1.3	systems1	System binaries for Sparc running SunOS 4.1.3.
sprite	graphics3	Macros to set bitmaps with T <sub>E</sub> X
ssquote	latex3	L <sup>A</sup> T <sub>E</sub> X package and font definition file to access the 'cmssq' fonts, i.e. Computer Modern Sans Serif Quotation Style. The L <sup>A</sup> T <sub>E</sub> X package also defines a <code>chapterquotes</code> environment as an example application.
startex	formats3	A T <sub>E</sub> X format designed to help students write short reports and essays. It provides the user with a suitable set of commands for such a task. It is also more robust than plain T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X.
stmaryrd	fonts2	St Mary Road symbols for functional programming.
subeqn	latex3	Package for subequation numbering.
subeqnarray	latex3	Equation array with sub numbering.
subfigure	latex3	Figures divided into subfigures.
supertabular	latex3	A multi-page tables package.

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
swift	latex3	Miscellaneous macros by Matt Swift.
tap	plain3	An advanced table package.
taylor	graphics3	Diagram macros by Paul Taylor.
tbe	plain3	Examples from Arvind Borde's <i>T<sub>E</sub>X by Example</i>
tengwar	fonts3	Font for typesetting Tolkien Tengwar script, by Michael Urban
tex-ps	generic3	Various extra support file for dvips.
texdraw	graphics3	Graphical macros, using embedded PostScript.
texip	formats3	Macros from <i>T<sub>E</sub>X in Practice</i>
texlive	texlive1	Basic material for <b>T<sub>E</sub>X Live</b> .
text1	formats3	T <sub>E</sub> X format from the University of Washington
textcomp	latex3	Supports the Text Companion fonts which provide many text symbols (such as baht, bullet, copyright, musicalnote, onequarter, section, and yen) in the TS1 encoding.
textfit	latex3	Package to support fitting of text to a given width or height by scaling the font
textmerg	latex3	Merge text in T <sub>E</sub> X and L <sup>A</sup> T <sub>E</sub> X. Useful, for example, in mail merge.
thesis	latex3	A class for producing a thesis based on the report class for a more European and more flexible look. Supports options like noindent, noitemization, headline, nocenter, crosshair, and chapterbib.
timesht	latex3	Package for typesetting time sheets.
tipa	fonts3	Fonts and macros for IPA phonetics characters.
tools	latex2	Standard L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> tools, for extended tabular, verbatim and theorem support
tracking	latex3	Automatically adjust spaces between symbols in words or phrases to fit them into a specified length. Any chain of symbols (including spaces) in the current font may be treated.
treesvr	latex3	Tree macros.
treetex	plain3	Allows the automatic layout of n-ary trees with arbitrary node sizes in L <sup>A</sup> T <sub>E</sub> X, using an external C program to do much of the hard work.
tugboat	generic2	Macros for TUGboat articles (plain and L <sup>A</sup> T <sub>E</sub> X).
type1cm	latex3	A package that removes the restriction when using scalable versions of the cm fonts (Type1 Bakoma, or versions from BSR/Y&Y, or True Type versions from Kinch, PCTeX etc.) where L <sup>A</sup> T <sub>E</sub> X restricts the cm fonts to discrete sizes.



Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
typehtml	latex3	Typeset HTML (i.e., World Wide Web documents) directly from L <sup>A</sup> T <sub>E</sub> X. Can handle almost all of HTML2, and most of the math fragment of the draft HTML3.
uaclasses	latex3	This package provides a L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> document class named ‘ua-thesis’ for typesetting theses and dissertations in the official format required by the University of Arizona. Moreover, there is a fully compatible alternative document class ‘my-thesis’ for private “nice” copies of the dissertation, and the respective title pages are available as separate packages to work with “any” document class.
ucthesis	latex3	A modified version of the standard L <sup>A</sup> T <sub>E</sub> X REPORT style that is accepted for use with University of California PhD dissertations and Masters theses.
ulsy	latex3	Extra mathematical characters
umlaute	latex3	An interface to inputenc for using alternate input encodings
umrand	fonts3	Package for page frames.
underlin	latex3	Package for underlining. Be advised that underlining is considered bad style in typesetting. See also ulem which is a specific package for L <sup>A</sup> T <sub>E</sub> X.
useful	doc2	Useful documentation; various L <sup>A</sup> T <sub>E</sub> X guides, FAQ, fontname docs, etc.
utthesis	latex3	Produces a thesis that meets the requirements of the Graduate School of The University of Texas at Austin :-)
uwthesis	latex3	University of Washington thesis
vdm	latex3	Typesetting VDM schemas
vector	latex3	Macros for more convenient representation of vectors in L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> , both symbolically and as implicit or explicit rows/columns of elements
vertex	plain3	Styles for economics working papers and journals
vita	latex3	This class provides necessary macros to prepare your Curriculum Vitae or Resume.
vrbl	latex3	Verbatim macros via plain T <sub>E</sub> X
vrslon	latex3	Defines a command which produces a version number in the .dvi-file when L <sup>A</sup> T <sub>E</sub> X is run.
wasy	fonts3	The wasy fonts (Waldis symbol fonts)

Catalogue of Packages *continued*

<i>Package</i>	<i>Collection</i>	<i>Description</i>
wasysym	latex2	Makes some additional characters available that come from the wasy fonts (Waldis symbol fonts). These fonts are not automatically included in NFSS2/L <sup>A</sup> T <sub>E</sub> X 2 <sub>ε</sub> since they take up important space and aren't necessary if one makes use of the packages amsfonts or amssymb. Symbols include: join, box, diamond, leadsto, sqsubset, lhd, rhd, apprle, ocircle, invneg, logof, varint, male, female, phone, clock, lightning, pointer, sun, bell, permil, smiley, various electrical symbols, shapes, music notes, circles, signs, astronomy, etc.
williams	latex3	Miscellaneous macros by Peter Williams.
win32	systems1	System binaries for Windows 32.
wnri	fonts3	METAFONT fonts for Old English, Indic languages in transcription, and American Indian languages.
wsuipa	fonts2	Washington State University IPA phonetic fonts
xymtex	latex3	Typesetting chemical structures.
xypic	graphics2	Sophisticated macros and fonts, originally designed for commutative diagrams, but with general applicability.
yhmath	latex3	Extended maths fonts for L <sup>A</sup> T <sub>E</sub> X.
youngtab	latex3	A package for typesetting Young-Tableaux mathematical symbols for the representations of groups, providing two macros, <code>\yng(#1)</code> and <code>\young(#1)</code> to generate the whole Young-Tableaux.
ytex	formats3	Macro package developed at MIT.
zed-csp	latex3	Typesetting Z and CSP format specifications

V tomto článku bych chtěl uvést několik věcí, které se mi na koncepci L<sup>A</sup>T<sub>E</sub>Xu nelíbí, a vysvětlit, proč jsem zůstal u plainu. Nechci tímto článkem vyvolávat nepokoje v řadách ČSTUGu<sup>1</sup>. Vážím si samozřejmě lidí, kteří perfektně rozumějí L<sup>A</sup>T<sub>E</sub>Xu, a tento článek nechce nijak znevažovat jejich práci. Nevylučuji ovšem, že článek vyvolá nějakou diskusi.

L<sup>A</sup>T<sub>E</sub>X je nadstavba postavená nad formátovacím programem T<sub>E</sub>X. Podle mého názoru hlavní cíle L<sup>A</sup>T<sub>E</sub>Xu jsou následující:

- Odstínit poměrně složitou problematiku T<sub>E</sub>Xu od koncového uživatele.
- Vytvořit vlastní jazyk vstupních textů.
- Umožnit formátovat jednoduché dokumenty podle předzpracovaných stylů.
- Umožnit snadnou výměnu dokumentů a jejich nové přeformátování.

Těchto cílů bylo dosaženo jen částečně. Pokusím se dokázat, že úplné dosažení stanovených cílů není možné z principiálních důvodů a že brzdou v této věci je, překvapivě, samotný T<sub>E</sub>X.

## Nedostatečnost jazyka T<sub>E</sub>Xu

T<sub>E</sub>X nebyl stvořen se záměrem, aby se nad ním dělaly univerzální nadstavby na úrovni jeho jazyka maker. Rozhodně nelze tyto nadstavby prezentovat veřejnosti jako univerzální systémy. Na to není programovací jazyk T<sub>E</sub>Xu stavěný. Proto nechápu, proč se o to autoři různých vylepšení L<sup>A</sup>T<sub>E</sub>Xu neustále snaží.

Můžete argumentovat, že vždy musí existovat hranice, pod níž nemusím zkoumat, jak systém funguje, mám-li si zachovat duševní zdraví. Například při jakékoli činnosti s počítačem potřebujeme určitou míru abstrakce. Programujeme-li třeba databázovou aplikaci, zajímají nás elementární nástroje vývojového prostředí a nezabýváme se tím, jak putují elektrony po polovodičích centrálního procesoru.

Pro tento způsob odstínění problému byla na každém stupni abstrakce vytvořena v počítačových vědách dobrá koncepce, která odstínění umožní. Musíme si ale přiznat, že T<sub>E</sub>X takovou dobrou koncepci nenabízí. Nelze pomoci

---

<sup>1</sup>Článek je publikován i přesto, že je dle recenzentů cíleně zkreslující, a tedy pro valnou část čtenářů i zavádějící. Též jazykové formulace a hrubé chyby nesplňují běžná kritéria pro publikování, nicméně redakce vyhověla autorovi a do textu nedělala již korekce. Pozn. red.

jeho makrojazyka vytvořit novou úroveň práce, při které není potřeba vracet se k vestavěným vlastnostem  $\text{T}_{\text{E}}\text{X}$ u. Z toho pramení většina problémů  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u (jak uvidíme jednotlivě dále). Základní chybou  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u tedy je, že si stanovil cíle, pro které nebyly vytvořeny podmínky v  $\text{T}_{\text{E}}\text{X}$ u samotném.

Uvedeme příklad. V běžných programovacích jazycích může programátor elementárními prostředky ošetřit všechny neočekávané situace programu. Všichni víme, že dobrý program by měl mít ošetřeny všechny alternativy vstupu, a považujeme za ostudu, když program havaruje se systémovou chybou. Makrojazyk  $\text{T}_{\text{E}}\text{X}$ u ale nemá potřebné nástroje na ošetření všech závad na úrovni „ $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  error“ (srovnej: chyba ošetřená programem). Chyba často vyjde najevo až po činnosti vnitřních algoritmů  $\text{T}_{\text{E}}\text{X}$ u a projeví se jako „ $\text{T}_{\text{E}}\text{X}$  error“ (srovnej: systémová chyba programu).

Pokud by se autor makra snažil vše ošetřit na úrovni svého makra, byl by vzniklý kód z 90 % o tom, jak ošetřit uživatelské výstřelky, a ve zbylých 10 % o vlastní sazbě. To je myslím zbytečná investice jednak lidského potenciálu (mám na mysli lidi, kteří taková makra dělají, a jiné, kteří jejich výtvořiny musejí číst) a jednak strojového potenciálu (mám na mysli strojový čas a paměť počítačů).

## Snaha $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u o odstínění složitosti $\text{T}_{\text{E}}\text{X}$ u

Jedním z cílů  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u bylo umožnit nepoučenému uživateli používat  $\text{T}_{\text{E}}\text{X}$ , aniž by musel pronikat do složitostí tohoto programu. Nakonec ale sami  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ovní odborníci přiznávají, že bez  $\text{T}_{\text{E}}\text{X}$ booku to prostě nejde. Takže snaha o odstínění problematiky  $\text{T}_{\text{E}}\text{X}$ u je vlastně zbytečná.

Představte si, že si nějaký uživatel přečte  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ovou příručku a nabude dojmu, že mu bude stačit rozumět problematice sazby na úrovni této příručky. Pak se jednou překlepne třeba při sestavování tabulky a na terminálu na něj  $\text{T}_{\text{E}}\text{X}$  křičí: `Extra alignment tab has been changed to \cr`. Uživatel začne znovu listovat ve své příručce a zjistí, že tam o žádném `\cr` není jediná zmínka. Má pak tři možnosti: (1) Zmáčkne Enter a podobně se zachová i u dalších chyb. Pomyslí si, že ten  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  je něco tajemného a mystického. (2) Propadne zoufalství a jde od toho. Dojde k závěru, že je lepší zůstat u Wordu. Vždyť stačí vzít tabulku v Excelu a jednoduše ji přemístit do Wordu a jaképak smolení se s nějakým podezřelým `\cr`. (3) Pořídí si  $\text{T}_{\text{E}}\text{X}$ book a po intenzivním studiu nakonec řekne: „aha“. V tuto chvíli ale už nepotřebuje, aby mu  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  zakrýval složitost  $\text{T}_{\text{E}}\text{X}$ u.

Nebudu podrobně popisovat zoufalý výraz v očích každého začínajícího uživatele  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u, když po použití `\vspace` tak, že nad značkou ani pod ní není prázdný řádek, dostane nečekávaný výsledek. Ve své příručce nemá žádnou zmínku o tom, co to je `\vadjust`, takže mu připadá chování `\vspace` jako tajuplné. Podobně zoufale se uživatelé dívají na obrazovku svého počítače, pokud jim uplavou plovoucí objekty (tabulky a obrázky) úplně jinam, než předpokládali.

Přitom oni chtěli jen použít značku `\caption` a to je přinutilo definovat objekt v prostředí, v němž může uplavat na jinou stránku.

## Utajení skutečností

Utajení některých důležitých vlastností  $\TeX$ u v  $\LaTeX$ ových příručkách v největší míře souvisí s předchozím bodem o snaze odstínit složitost. Ovšem některé chybějící informace jsou těžko odůvodnitelné. Za všechny uvedu jen dva příklady.

$\LaTeX$ ový uživatel se nikde nedočte o makru `\choose`, které je elementární zkratkou použití primitivu `\atopwithdelims` a vytváří binomické koeficienty. Můžeme sice v  $\LaTeX$ u napsat

```
\left(\begin{array}{c} n \\ k \end{array}\right)
```

ale lidé aspoň s nepatrnou mírou vkusu tuší, že to je podivná snůška  $\TeX$ ových primitivů (`\left`, `\right`) kombinovaná s  $\LaTeX$ ovým monstrem `array`.

Druhým příkladem je chybějící zmínka o primitivu `\eqno` a `\leqno` pro značkování rovnic. Skutečně, v celém  $\LaTeX$ u to nenajdete a v jiných příručkách taky ne. Autoři těchto příruček zřejmě žili v domněnání, že jejich čtenáři si za všech okolností vystačí s automatickým značkováním rovnic, které jim poskytuje  $\LaTeX$ . Ovšem chyba lávky. Zúčastnil jsem se jednoho  $\LaTeX$ ového kurzu, kde řešili problém, jak dostat na okraj rovnice konkrétní značku, nepřidělenou žádným automatem. Například značku (P). Řešení tohoto úkolu bylo impozantní. Nejprve se vysázela rovnice, pak se pomocí záporného `\space` vycouvalo zpět na úroveň rovnice, dále se použilo `\begin{flushright}`, napsala se značka a konečně se to uzavřelo pomocí `\end{flushright}`. To asi nepotřebuje dalšího komentáře.

## Nerozlišování mezi $\LaTeX$ em a $\TeX$ em

Skoro v žádné  $\LaTeX$ ové příručce se nedozvíme přesnou souvislost mezi  $\LaTeX$ em a  $\TeX$ em. Občas v některé najdeme mlhavé přirovnání rozebírající vztah závodního vozu s pohodlným rodinným korábem, ale vyložení technických souvislostí považují asi autoři příruček za složité.

Z uvedených důvodů se v  $\LaTeX$ ových příručkách nerozlišuje mezi vlastnostmi  $\TeX$ u a  $\LaTeX$ u. Pro jednoduchost se všude mluví o  $\LaTeX$ u. Například  $\LaTeX$  formátuje odstavce,  $\LaTeX$  rozdělí text na více stránek. Víme, že to není pravda. Když pak někteří  $\LaTeX$ oví uživatelé přijdou na to, že kromě  $\LaTeX$ u existuje ještě nějaký podivný  $\TeX$ , jsou obvykle zmateni. Své zmatení dávají často na jevo novotvarem  $(\LaTeX)$ .

## Vlastní jazyk vstupních textů

L<sup>A</sup>T<sub>E</sub>X se pokusil definovat vlastní jazyk pro značkování vstupních textů. Snaha byla, aby značky podléhaly pokud možno jednotným principům a byly pro uživatele pochopitelné. Jazyk má také pomoci uživateli vyhnout se různým syntaktickým úskalím jazyka T<sub>E</sub>Xu.

Srovnajme způsoby zápisu, které zhruba(!) znamenají totéž:

T <sub>E</sub> X	L <sup>A</sup> T <sub>E</sub> X
<code>\hsize=30pc</code>	<code>\setlength{\textwidth}{30pc}</code>
<code>\vskip1cm</code>	<code>\vspace{1cm}</code>
<code>{a\over b}</code>	<code>\frac{a}{b}</code>
<code>\hbox{abc}</code>	<code>\mbox{abc}</code>
<code>\vbox{...}</code>	<code>\begin{minipage} ... \end{minipage}</code> nebo <code>\parbox{...}</code>

Bohužel (nebo bohudík?), v L<sup>A</sup>T<sub>E</sub>Xu je možno psát i některé přímé T<sub>E</sub>Xové konstrukce. To je další nedůslednost v odstínění problematiky T<sub>E</sub>Xu před L<sup>A</sup>T<sub>E</sub>Xovým uživatelem. Jakmile to L<sup>A</sup>T<sub>E</sub>Xový uživatel zjistí, začne většinou míchat oba dva způsoby zápisu: T<sub>E</sub>Xový i L<sup>A</sup>T<sub>E</sub>Xový. To dokazuje, že snaha o vytvoření lepšího, přehlednějšího, a přitom stejně mocného jazyka, se zcela nepovedla.

T<sub>E</sub>X nám například hlásí `Overfull \hbox` a L<sup>A</sup>T<sub>E</sub>X jako makro nemá prostředky toto hlášení změnit na `Overfull \mbox`. Záznam o sazbě je tedy L<sup>A</sup>T<sub>E</sub>Xovému uživateli předkládán pomocí T<sub>E</sub>Xového a nikoli L<sup>A</sup>T<sub>E</sub>Xového slovníku. Považuji to za matení nebohých uživatelů.

## Strukturované značkování

Pod tímto pojmem rozumím značky, které člení dokument do struktury (například kapitola, citát, poznámka) a nezabývají se typografií dokumentu (zda použít `\medskip` nebo `\bigskip` a jaký zavést do sazby font).

L<sup>A</sup>T<sub>E</sub>Xoví zastánci zdůrazňují, že koncepce vstupního jazyka L<sup>A</sup>T<sub>E</sub>Xu vede uživatele ke strukturovanému značkování. To není pravda.

Uživatele, který teprve začíná pracovat s T<sub>E</sub>Xem nebo s jakýmkoli jiným sázečím systémem, velmi těžko přesvědčíme o výhodách strukturovaného značkování. Chce to dlouhodobou praxi, než člověk na výhody tohoto způsobu značkování přijde. Většina lidí se rozhodne pořídit první dokument proto, aby jednotlivé jeho části vypadaly přesně tak a tak. Nasadí si tak na oči klapky, které se těžko sundávají.

Mé zkušenosti s dokumenty různých autorů ukazují, že v používání strukturovaného značkování není rozdíl mezi uživateli plainu a L<sup>A</sup>T<sub>E</sub>Xu. Spíš bych řekl,

že dlouhodobá praxe vede ke strukturovanému značkování bez závislosti na tom, jakými prostředky je text pořízen.

Domnívám se také, že člověk snáze dospěje ke koncepci strukturovaného značkování, pokud má vzhled dokumentu absolutně pod kontrolou svých vlastních maker. Používat značky vymezující strukturu dokumentu a nevědět, co se *přesně* stane v době, kdy mě bude zajímat vzhled dokumentu, může být pro uživatele, který je zodpovědný i za typografii dokumentu, frustrující<sup>2</sup>.

Idea strukturovaného značkování se nejlépe projeví při dělbě práce: písárka a sazeč. Pokud sazeč chce postavit sazbu dokumentu na plainu, dohodne se s písárkou na některých značkách definujících strukturu dokumentu. Jestliže chce písárka průběžně vidět, jak dokument vypadá, dají se některé značky narychlo nadefinovat (například `\let\značka=\relax`). Při vlastním zpracování dokumentu pak sazeč definuje jednotlivé značky podle požadavků na typografii dokumentu. Vzhled knihy mu vzniká rovnou pod rukama a před očima.

Pokud sazeč při návrhu značek vychází z plainu, má písárka většinou podstatně méně práce. Sazeč totiž může s výhodou využít mocný nástroj v podobě separátorů parametrů v makrech  $\TeX$ u, který není v  $\LaTeX$ u pro potřeby uživatele využit. Ve smluvených značkách je pak daleko nižší výskyt kučeravých závorek a dalších droblenek, které nejsou zcela potřeba. Například pro název kapitoly se ti dva mohou dohodnout na:

```
\kap Název kapitoly  
<prázdný řádek>
```

Je to jednak přehlednější a jednak to vyžaduje od písárky méně námahy, než kdyby postupovala podle  $\LaTeX$ u.

## Ideální dělba práce

V předchozí úvaze jsme naznačili ideální dělbu práce. Autor nebo písárka pořídí text se značkami podle pokynů sazeče. Sazeč, který se naučí  $\TeX$ , pak sám doplní makra podle pokynů typografa. Domnívám se, že to byl původní záměr autora programu  $\TeX$ .

$\LaTeX$  se snaží uvedenou dělbu práce obejít. Autor může nabýt dojmu, že mu stačí napsat text podle jednoduché příručky k  $\LaTeX$ u a sazečem se stává samotný  $\LaTeX$ . Protože schází lidský faktor mezi autorem a konečným vzhledem dokumentu, trpí tím často autor i dokument.

Z jedné strany dává  $\LaTeX$  autorovi najevo, že typografii zvládne automat sám a dobře (což vůbec není pravda), a z druhé strany nakonec nutí autora získat znalosti, které by při lidské dělbě práce nebyly pro něj potřebné. Pokud má autor

---

<sup>2</sup>Všeobecně rozšířený názor je právě opačný. Autor by se měl při psaní plně abstrahovat od vizuální podoby a koncentrovat se na logické značkování. Pozn. red.

realizovat přesně požadavky typografa, musí proniknout nejen do problematiky  $\text{T}_{\text{E}}\text{X}$ u, ale též do problematiky složitých maker  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u, ve kterém původně svůj dokument napsal. Tím se z autora stává jakýsi polyhistor. Autor na počátku pořizování svého dokumentu určitě netušil, že na něj budou nakonec kladeny tak vysoké požadavky.

## Předzpracované styly

Mezi zastánci  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u se může rozšířit klamná vize, že všechny požadavky na sazbu, které kdy budou potřebovat, zvládnou výběrem vhodného stylu. Vůbec tedy není potřeba rozumět  $\text{T}_{\text{E}}\text{X}$ u, stačí zjistit, jaký styl použít, napsat jeho název do hlavičky a máme vystaráno.

Chtěl bych zdůraznit, že základní styly v  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u jsou po typografické stránce poměrně dost nevhodné. Především jde o volbu velikostí nadpisů a jejich umístění v sazbě. Rovněž existence pružných vertikálních výplňků je ve většině případů naprosto nepřipustná. Zbavit se jich lze změnou poměrně malého množství parametrů, ale ve svém důsledku to znamená přepsat si aspoň `\output` rutinu podle vlastní koncepce. Jestliže to ale uděláme a provedeme to dobře, jsme na nejlepší cestě vytvořit si vlastní makro, které nazveme například `Mij\text{T}_{\text{E}}\text{X}` nebo jinak.  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  jako takový už nepotřebujeme.

Svémi základními styly dělá navíc  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$   $\text{T}_{\text{E}}\text{X}$ u negativní reklamu. Dovedu si představit, že kupříkladu nějaký nakladatel projeví nezávazný zájem o  $\text{T}_{\text{E}}\text{X}$  a udělá povrchní průzkum věci. Přitom jako první narazí na  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  a prefabrikované dokumenty jednoho typu. Může se stát, že průzkum na této úrovni skončí a nakladatel o  $\text{T}_{\text{E}}\text{X}$  přestane mít další zájem. Je pravda, že může ztratit zájem i o plain, dívá-li se na problém takto: plain bez dodatečných maker neumí vytvořit obsah ani rejstřík, zatímco Ventura to udělá kliknutím myši.

Dalším problémem  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u je snášenlivost stylů různých výrobců mezi sebou. Toto není možné elementárními prostředky  $\text{T}_{\text{E}}\text{X}$ u nikdy uspokojivě vyřešit. Vždy budou existovat detaily, ve kterých se mohou některé styly vzájemně pohádat. Přitom výsledný efekt a případné chybové hlášení je velmi často dosti daleko od podstaty věci a náprava se těžko hledá. Odstranění takových chyb pak vyžaduje nejen perfektní znalost  $\text{T}_{\text{E}}\text{X}$ booku, ale též důkladnou znalost kódů jednotlivých stylů a kódu samotného  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u. A především notnou dávku trpělivosti. Dospějete postupně k závěru, že na úrovni plainu byste se stejnými znalostmi a s podstatně menším úsilím vytvořili makro, které veškeré typografické požadavky řeší také. Důležitá je efektivita práce.

Vždy budou existovat požadavky, které nepůjdou vyřešit pouze zavedením dalšího stylu do  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u. Pak se uživatel musí rozhodnout, zda modifikovat styl, nebo si problém vyřešit sám.



## Složitost maker $\LaTeX$ u

Většina stylů musí být aspoň částečně schopna pracovat univerzálně v různých podmínkách. Mají proto ošetřeno plno věcí, které by autor makra „na míru“ nikdy nedělal. Kód takových stylů se pak stává zbytečně složitým a nepřehledným. Příkladem takového nepřehledného a zbytečně složitého kódu (protože chce být univerzální) je samotné jádro  $\LaTeX$ u. Navíc, jak plyne z nedostatečnosti makrojazyka  $\TeX$ u, je požadavek na univerzálnost  $\LaTeX$ u principiálně těžko dosažitelný.

Ve většině případů složitost  $\LaTeX$ u ve svém dokumentu uživatelé v plné šíři nevyužijí. Určitě plno maker  $\LaTeX$ u zůstává při sazbě dokumentu v nečinnosti. Ta makra, která něco činí, by mohla stejnou práci vykonávat efektivněji a jednodušeji. To snese přirovnání o dělové kouli proti mouše. Přitom si ta moucha pohodlně uletí, protože než se tam ta dělová koule přivalí, to chvíli trvá.

Zkuste se namátkou zeptat nějakého  $\LaTeX$ isty, kterými cestami se ubírá napsané „ř“ ve vstupním kódu dokumentu, než se dopracuje k výstupnímu písmenu „ř“ v dvi. Bazírujte na naprosto přesném popisu expanze všech maker, která jsou při této činnosti použita. Většina lidí vám na tuto otázku nebude schopna dát odpověď. Ona totiž ta odpověď vůbec není jednoduchá. Dal jsem si tu práci a jeden půlden jsem v kódu  $\LaTeX$ u tuto cestu hledal. Když jsem zjistil, jakými zákoutími a skulinami tato cestička vede, zhrozil jsem se a začal jsem vážně pochybovat, jak to může vůbec stabilně fungovat. Věc je závislá na velkém množství dalších okolností, takže v žádné chvíli nemáme absolutní jistotu, zda „ř“ povede vždy na „ř“. Zákonitě to musí zlobit při použití primitivu `\uppercase`. Proto  $\LaTeX$  tento primitiv dost nepřehledně předefinovává.

Když použiji věci, o kterých vím do posledního šroubu, jak fungují v  $\TeX$ u samotném, nebudu mít nikdy jistotu, co to udělá v  $\TeX$ u zavírovaném  $\LaTeX$ em. Tato nejistota mě od  $\LaTeX$ u poměrně značně odrazuje.

K veřejnému použití jsem například nabídl makro na čárové kódy EAN. Občas mi nějaký uživatel napíše, že to nefunguje, a přitom to použil v  $\LaTeX$ u. Odpovídám stručně: „Použijte (cs)plain, za  $\LaTeX$  já neručím. Pokud to náhodou funguje v  $\LaTeX$ u, považujte to za zázrak<sup>3</sup>. Stejně si nebudete jistý, zda to za půl roku v tom  $\LaTeX$ u ještě bude fungovat.“ Umíte si představit, jak mi povstaly vlasy hrůzou na hlavě, když mi jeden dobrák ze zahraničí oznámil, že z mého makra `ean.tex` vytvořil odrůdu `ean.cls`!

---

<sup>3</sup>Makro EAN je beze změny používáno v redakci.  $\LaTeX$  2<sub>ε</sub> přináší jedinou obtíž, která již byla ve Zpravodaji zmíněna. Ve snaze o zavedení českých pravidel pro dělení slov obsahujících spojovník je znak „minus“ aktivní, pokud se při načítání `czech.sty` použije parametr „split“. Proto se `ean.tex` musí načítat v preambuli a před vlastním voláním `ean` je nutno nastavit `\catcode\-=12`.

Většina stylů psaných v plain  $\TeX$ u je v  $\LaTeX$ u přímo použitelná. Není to žádný zázrak, spíše ukázka možností modularity výstavby dokumentu. Pozn. red.

## Přenositelnost dokumentů

Je potřeba odlišit dva typy přenosu dokumentů. V první řadě se budeme ptát po přenositelnosti textu dokumentu včetně struktury, ale bez přesných specifikací formátu sazby. Příjemce pak dokument formátuje podle svých požadavků. Výsledek může vypadat docela jinak, než jak dokument viděl na obrazovce autor. Takový způsob přenosu se používá například při sestavování sborníků. V druhém typu přenosu budeme chtít přenést zcela věrně nejen text dokumentu, ale i formát sazby. Požadavkem je, aby příjemce viděl na obrazovce totéž co autor.

L<sup>A</sup>T<sub>E</sub>X je poměrně dobře koncipován pro první typ přenosu. Svým jazykem totiž deklaruje jednotný způsob značkování dokumentů, který by se mohl s trochou nadsázky nazvat mezinárodním standardem. Praxe ale ukazuje, že nic není tak růžové, jak se na první pohled zdá. Už jste někdy vytvářeli sborník příspěvků rozličných autorů? Uvedu příklad ze své zkušenosti: 50 % příspěvků přišlo ve Wordu či něčem podobném. Po troše konverzní práce je využitelný většinou jen holý text. 40 % příspěvků přišlo v různých verzích L<sup>A</sup>T<sub>E</sub>Xu. Posledních 10 % přišlo v plainu nebo jako holý ASCII text. Nakonec se ukázalo, že s těmito příspěvky bylo nejméně práce.

L<sup>A</sup>T<sub>E</sub>X jako standardní jazyk pro značkování struktury dokumentu nám příliš nepomohl. Příspěvky v L<sup>A</sup>T<sub>E</sub>Xu totiž byly „každý pes jiná ves“. To je dáno jednak různými verzemi současného L<sup>A</sup>T<sub>E</sub>Xu a jednak neschopností většiny příspěvateľů oddělit strukturu dokumentu od sazby. V jejich příspěvcích se to hemžilo značkami jako `\vspace{5mm}`, `\clearpage`, `\`, `{\Large Nadpis}` atd. Aby to k něčemu bylo, museli by všichni příspěvateľé přistoupit na společný způsob značkování. Nestačí říci, že je to L<sup>A</sup>T<sub>E</sub>X. Je potřeba vyjmenovat všechny povolené značky vymezující strukturu a pro jistotu uvést též všechny obraty, které jsou pro konečnou typografii sborníku nevhodné. Ukazuje se, že těch pár obrátů, které zůstávají povoleny, lze zpracovat jednoduchým makrem. Ačkoli tyto obraty vypadají L<sup>A</sup>T<sub>E</sub>Xovsky, není nakonec nutno použít L<sup>A</sup>T<sub>E</sub>X.

Pokud jde o přenositelnost dokumentů včetně formátu, je na tom L<sup>A</sup>T<sub>E</sub>X dost špatně. Především různé verze L<sup>A</sup>T<sub>E</sub>Xu mohou vést k různým výsledkům. Rovněž koncepce stylů v tuto chvíli naráží na problémy. Většina uživatelů L<sup>A</sup>T<sub>E</sub>Xu totiž implicitně předpokládá, že příjemce jeho dokumentu má povinnost mít instalovány všechny stylové soubory, které jsou v jeho dokumentu použity. Je pravda, že L<sup>A</sup>T<sub>E</sub>X má nástroje k vložení všech pomocných zdrojů do dokumentu. To je vhodné pro výměnu dokumentů mezi uživateli, ale ti to většinou nepoužívají. Méně znalí uživatelé se navíc chovají k záhlaví svého dokumentu jako k nějakému tabu, které někde obkreslili a někdo poučenější jim do toho dopsal volání nějakých nových stylů. Výsledkem často je, že zasláný dokument používá v záhlaví dvacet různých stylů, přičemž se nakonec ukáže, že jsou potřeba jen tři. Při obdržení takového dokumentu je proto nejprve

nutné minimalizovat počet použitých stylů, dále zvážit, zda kvůli jednomu efektu v dokumentu má cenu toulat se po Internetu a shánět styly, které ještě v naší instalaci  $\TeX$ u nemáme. Mnohdy je jednodušší chyby způsobené nezavedením stylu prostě přeskočit, vždyť ono to nějak dopadne. Pak přijde velké překvapení: ono to dopadlo většinou uspokojivě. Proč tam ten styl vlastně byl?

Na druhé straně uživatelé plainu dobře vědí, jaká makra jsou v zasílaném dokumentu potřeba. Vycházejí z toho, že příjemce bude mít standardní a v jednotlivých verzích skoro neměnný plain nebo csplain. Všechny definice, které rozšiřují uvedené formáty, pak přibalí naprosto vědomě do svého dokumentu. Dokumenty pak jsou bezproblémově přenositelné. Schopnost přesně rozlišovat mezi jádrem formátu a nadstavbou je především vlastností uživatelů plainu, protože jsou sami nuceni tomu pořádněji rozumět.

Uvedu jeden kuriózní příklad. Návrh stanov  $\zeta$ TUGu byl napsán v plainu a bylo jej tedy možné vystavit na síti tak jak je. Bez obav, že bude mít příjemce při zpracování problémy. Na druhé straně volební kandidátka byla napsána v  $\LaTeX$ u a na síti jsme ji v podobě zdrojového textu neviděli. Víte proč? Autor tohoto dokumentu použil pro čtverečky na zakřížkování zvolených jmen speciální font. To ovšem v  $\LaTeX$ u znamená, že je potřeba mít správnou verzi souboru `fd` pro tento font a plno dalších záležitostí. Jinak to nebude fungovat. Autor kandidátky mě proto explicitně požádal, abych z důvodů komplikací s přenositelností dokument ve zdrojové podobě nevystavoval. Já sám jsem půl hodiny pracoval na tom, než jsem byl schopen kandidátku bezproblémově vytisknout. Z Internetu jsem musel přitáhnout použité styly, `fd` toho fontu a font samotný. Kdybych tušil, že se jedná o přihlouplé čtverečky, měl bych pro ně makro postavené na primitivech `\hbox`, `\vbox`, `\vrule` a `\hrule` uděláno během pár vteřin. A dokument by byl přenositelný.

## Nestálost verzí $\LaTeX$ u

To je kapitola sama pro sebe. Divím se, že pohyb verzí a zmatek, který kolem toho vznikl, neodradil uživatele od používání  $\LaTeX$ u. Všichni uživatelé  $\LaTeX$ u 2 $\epsilon$  jsou vlastně pokusnými králíky a jsou zúčastněni na velkém projektu, který směřuje snad někdy k  $\LaTeX$ u 3.

Jako poměrně velký podraz vůči uživatelům považuji moment, kdy se pojmem  $\LaTeX$  začalo nazývat něco, co se od původního a hojně používaného makra tohoto jména ( $\LaTeX$  2.09) poměrně zásadně liší. Jaký terminologický chaos vznikl po tomto dravém nástupu  $\LaTeX$ u 2 $\epsilon$ , to jistě každý čtenář dobře ví. Když nyní řeknu slovo  $\LaTeX$ , nikdo neví, co tím mám na mysli.

Každý půlrok přichází na svět nová verze  $\LaTeX$ u 2 $\epsilon$ , která je s předchozí slučitelná jen na uživatelské úrovni. Pokud ovšem uživatel přesáhne svými znalostmi běžné příručky k  $\LaTeX$ u a tyto znalosti použije a podepře je vlastnostmi

nedokumentovaných funkcí konkrétního jádra  $\LaTeX$ u, má za půl roku smůlu.

Mnozí z nás byli svědky neslučitelnosti různých verzí počestění  $\LaTeX$ u s různými verzemi jádra  $\LaTeX$ u  $2_\varepsilon$ . Také víme, že tyto problémy vycházely najevo na naprosto neočekávaných místech (například použití makra `\ldots`). To samo o sobě dokumentuje obtížnou kontrolu nad složitými makry<sup>4</sup>  $\LaTeX$ u a nesnadnou implementaci národní nadstavby k  $\LaTeX$ u.

Půlroční změny verzí  $\LaTeX$ u musí udržovat uživatele i administrátory systémů v neustálé pozornosti. To je úmorné a vyčerpávající. Přiznávám se, že já jako administrátor systému pro naši katedru tuto rychlost pohybu nestačím sledovat a docela mě unavuje.

## Jednorázová investice do vzdělání

Naučit se rozumět vestavěným algoritmům  $\TeX$ u a správnému použití primitivů nebo `maker plainu` je činnost, kterou stačí udělat jednou pro vždy. Naopak, v případě stylů  $\LaTeX$ u člověk musí být neustále ve střehu a musí se postupně stát pochodujícím slovníkem na styly, aby dokázal ke konkrétnímu účelu vybrat a správně použít ten správný styl té jediné správné verze. Přiznávám, že to přesahuje kapacitu mého mozkového potenciálu, a proto zůstávám při zemi: pouze se znalostmi algoritmů  $\TeX$ u, jeho primitivů a `maker plainu`. Tyto věci nejsou v pohybu, takže se nemusím učit každý rok něco nového.

## Závěr

$\TeX$  je koncipován jako otevřený systém. Kdo má chuť a sílu si osvojit jeho detaily, má v ruce mocný nástroj na vytváření nadstaveb podle svých představ. Není podle mého názoru férové vnucovat uživatelům „jedinou správnou nadstavbu“ a v příručkách tvrdit, že při jejím použití nebudou muset vědět, co to je  $\TeX$ .  $\LaTeX$  se tím snaží zastřešit otevřený základ univerzální nadstavbou, čímž neguje výhody základu. Protože je toto zastřešení navíc děravé z důvodů nedostatečnosti samotného základu, utrpí tím nakonec oba. Jednak základ a jednak nadstavba. A to je škoda.

Domnívám se, že kdyby se více úsilí věnovalo výkladům a vysvětlování principů samotného  $\TeX$ u, bylo by to jen ku prospěchu věci. Udržovat neustále

---

<sup>4</sup>Tato makra nejsou složitá. Problémy plynou z toho, že  $\zeta\TeX$  se podporou  $\zeta$ -fontů odchýlil od mezinárodního standardu.  $\LaTeX$  2.09 byl psán dosti nepřehledně a nestrukturovaně, makra jsou často implementována na nečekaných místech.  $\LaTeX$   $2_\varepsilon$  tyto nečnosti postupně odstraňuje a opravy, které postupně zavádí  $\LaTeX$  3 team, se musí promítnout i do souborů podporujících národní podporu. Jinak dochází k nečekanému předefinování „českého“ makra na „mezinárodní standard“, což způsobí nějaký typ katastrofy. Pozn. red.

přehled o vlastnostech kódu  $\LaTeX$ u a všech jeho současných stylů a verzí je podle mého názoru daleko náročnější, než se jednou pro vždy seznámit se samotným  $\TeX$ em.

---

---

## Příprava a tisk jednoduché brožury s případným využitím stylu *fancyheadings*

---

JOSEF BARÁK

### Úvod

Mnoho dokumentů v  $\TeX$ u je navrženo ve formátu A4, který je však pro běžné čtení poměrně nepohodlný. Častá velikost učebnic a brožur je proto přibližně A5. Tímto formátem docílíme nejen větší přehlednosti, ale i úspory papíru.

Vzhledem k tomu, že jsem nedávno připravoval brožuru s fyzikální tematikou a musel přitom vyřešit některé problémy, chtěl bych se podělit o své zkušenosti s uživateli  $\TeX$ u, kteří dosud takovouto práci nikdy nedělali. Protože používám  $\LaTeX 2_{\epsilon}$ , zaměřím se ve svém popisu právě na něj.

Při vytváření brožury musíme vyřešit dva hlavní problémy: vložit vhodné příkazy do preambule dokumentu (tj. před `\begin{document}`) a rovněž nakonfigurovat prohlížeč tak, abychom si mohli výsledky své práce prohlédnout (a samozřejmě rovněž vytisknout) v reálné podobě. Moje požadavky na vytvářenou brožuru byly následující:

- tisk čtyř stran A5 na list papíru A4 (oboustranně)
- pořadí stran tak, aby se z potíštěných listů papíru dala přímo složit brožura
- na každé straně musí být podtržené záhlaví, ve kterém bude vždy číslo strany a dále střídavě názvy kapitol a podkapitol
- bude počítáno s korekcí tloušťky papíru (jinak by po složení a ořezu brožury byl levý okraj textu na každé straně opticky posunutý)

### Příkazy v preambuli dokumentu

Přestože pro vytváření běžné brožury vystačíme s prostředky standardní instalace  $\LaTeX$ u, je dle mých zkušeností lépe využít styl *fancyheadings*<sup>1</sup> od Pieta van

---

<sup>1</sup>Například `ftp.cstug.cz/pub/tex/CTAN/macros/latex209/contrib/fancyheadings`

Oostruma z utrechtské univerzity. Pokud totiž nevyužijeme tento nebo nějaký podobný styl, narazíme na některá omezení:

- nejde vytvořit víceřádkové záhlaví (na začátku každé strany)
- záhlaví nelze podtrhnout
- název kapitoly nemůžeme umístit zcela vpravo

Pomocí *fancyheadings* lze provádět samozřejmě mnoho dalších věcí, a to nejen pro záhlaví, ale i pro patičku. Tyto další možnosti jsem nezkoušel. Podrobnosti jsou však k nalezení v dokumentaci v souboru *fancyhea.dvi*.

Pro brožuru, která má strany formátu A5, sázíme samozřejmě užší sloupec textu, než pro (jednosloupcovou) sazbu na formát A4. Z toho vyplývá častější výskyt overflow boxů a podobných problémů při překladu. Velké části z nich předejdeme zařazením příkazu `\sloppy`. Tento příkaz způsobí, že požadavky na velikost mezer jsou méně přísné<sup>2</sup>.

Nyní bych si dovilil uvést příklad zdrojového textu. Tam, kde je řádek doplněn komentářem `[fh]`, nelze daný příkaz použít bez doplňkového stylu *fancyheadings*.

```
\documentclass[twoside,a5paper]{book}
% dvě strany A5 na stranu A4

\usepackage{czech,fancyhea}

\pagestyle{fancy} % [fh]

\setlength{\headrulewidth}{1pt}
% tloušťka čáry podtrhující záhlaví [fh]

\lhead[\rm\thepage]{\rm\rightmark}
% hlavička na každé levé straně [fh]

\rhead[\rm\leftmark \hspace{-1.5ex}]{\rm\thepage}
% hlavička na každé pravé straně [fh]

\cfoot{} % číslo strany není dole uprostřed [fh]

\sloppy % méně přísné požadavky na mezery
```

---

<sup>2</sup>Do běžného režimu sazby odstavce se můžeme vrátit příkazem `\fussy` nebo lépe používat příkaz `\sloppypar` pro jednotlivé odstavce.

## Konfigurace ovladače tiskárny

Jak už jsem uvedl výše, pro správné prohlédnutí a tisk dokumentu musíme správně nastavit ovladače (nejčastěji dvidrv E. Mattese)<sup>3</sup>. Pokud používáme `TeX-menu-system` dr. Olšáka ze standardní instalace `CTeX`, je nejvýhodnější vytvořit si pomocí `Print\Edit Spec. cnf` konfigurační soubor, nazvaný např. `broz.cnf` a tento soubor použít pomocí nabídky `Print\Spec. cnf`. Tím je zaručeno, že nemusíme potřebné parametry zadávat vždy při volání ovladače.

*Upozornění:* Hotový text je nutné prohlédnout si pomocí `Print\Preview` a nikoliv pomocí běžného `TeXView`; volba `View` vyvolá pouze ovladač obrazovky `dviselect` bez možnosti vložení parametrů, a tedy nelze využít ani konfiguračního souboru.

Pro znalost všech parametrů, a to nejen využitelných pro tisk brožury, doporučuji prostudovat si dokumentaci<sup>4</sup> k ovladačům `dvidrv`.

Nyní bych uvedl obsah svého konfiguračního souboru `broz.cnf`. Upozorňuji, že lomítka uvedená v tomto souboru jsou správná, neboť se jedná o parametry, nikoli o příkazy `TeX`.

```
%
% Soubor "broz.cnf":
%
/tr1 % rotace o 90 stupňů,
%     aby se nám vešly dvě strany vedle sebe
/ar1 % rows; počet tiskových stran pod sebou, nyní 1
/ac2 % columns; počet tiskových stran vedle sebe, nyní 2
/an1 % section-count; nastavení čítače
/as0.1575mm % korekce na tloušťku papíru
/ah210mm % výška formátu A5
/aw148.5mm % šířka formátu A5
/d:s %   oboustranný tisk (jen pro některé typy tiskáren);
%       obvykle však místo toho použijeme:
% /d:o % pro tisk lichých stran a
% /d:e % pro tisk sudých stran
```

Jak je zřejmé z konfiguračního souboru, problémem je též oboustranný potisk listů A4 (tj. abychom na jeden list A4 dostali čtyři strany formátu A5). Zde máme buď možnost použít parametr `/d:s` (`double-sided: short`), což lze jen

---

<sup>3</sup>Uživatele jiných instalací a operačních systémů možno odkázat na balík univerzálních programů `dvibook`, `dviselect`, `...`, který je k dispozici na CTAN. Tímto způsobem lze pracovat i v případě, že ovladač výstupního zařízení, který máme po ruce, nemá podporu ekvivalentní Mattesovu `dvidrv`.

<sup>4</sup>Součást `CTeX`: `/EMTEX/DOC/ENGLISH/DVIDRV.DOC`

u některých typů tiskáren, anebo pro většinu běžných typů tento parametr vymažeme. Dále nastavíme nejprve /d:o (odd) pro tisk lichých stran a následně pak /d:e (even) pro dotisk sudých stran.

Parametr /as slouží ke korekci na tloušťku papíru. Podle doporučení v dokumentaci DVIDRV.DOC by měla být u tohoto parametru uvedena hodnota, která je asi 1,5-násobkem tloušťky listu. Tloušťku listu jsem zjišťoval měřením více listů papíru posuvným měřidlem. Pro papír  $80 \text{ g m}^{-2}$  bylo naměřeno 0,105 mm. Tedy hodnota v parametru bude 0,1575 mm. S touto hodnotou je však třeba experimentovat a upravit ji podle skutečně dosaženého výsledku při skládání brožury.

*Josef Barák*  
*xbarak00@kpve.fee.vutbr.cz*

---

---

## Announcement of the 18th Annual Meeting of the TeX Users Group

---

*July 28–August 1, 1997*  
*Lone Mountain Conference Center*  
*San Francisco, California*

We would like to extend an invitation to TeX users around the world to join us in one of the most beautiful and exciting cities in the world.

The theme of this year's meeting is "TeX Comes Home". Papers have been submitted in the following general areas:

- Picture this: the TeXie approach to graphical illustration
- Tooling up: where are we with TeX?
- TeX and Scientific Publishing on the Internet
- TeX behind the scenes: what is our relationship to SGML?
- TeX and the real world
- L<sup>A</sup>TeX—state of the art
- Multilingual typography without boundaries

There are still timeslots available for additional papers—If you were considering submitting a paper, please contact the program committee at <mailto:tug97-1@mail.tug.org>.

We also welcome exhibitors and those who wish to conduct workshops in TeX-related topics.



Besides the formal presentations, you will also enjoy the commercial exhibits, posters, workshops, and “Birds-of-a-Feather” sessions.

We hope to arrange bursary funds for support of students and those participants who demonstrate need. For bursary requests, send email to `mailto:bursary@mail.tug.org` or by postal mail to

TUG'97 Bursary Committee  
135 Center Hill Road  
Plymouth, MA 02360 USA

## Contact Information

For complete details, please see our Web page at `http://www.tug.org/tug97` or request an email package from `mailto:tug97-1@mail.tug.org`.

If you are interested in attending and wish to be kept up to date with the latest conference information including program, cost, etc., please send email to `mailto:tug97-1@mail.tug.org`.

---

---

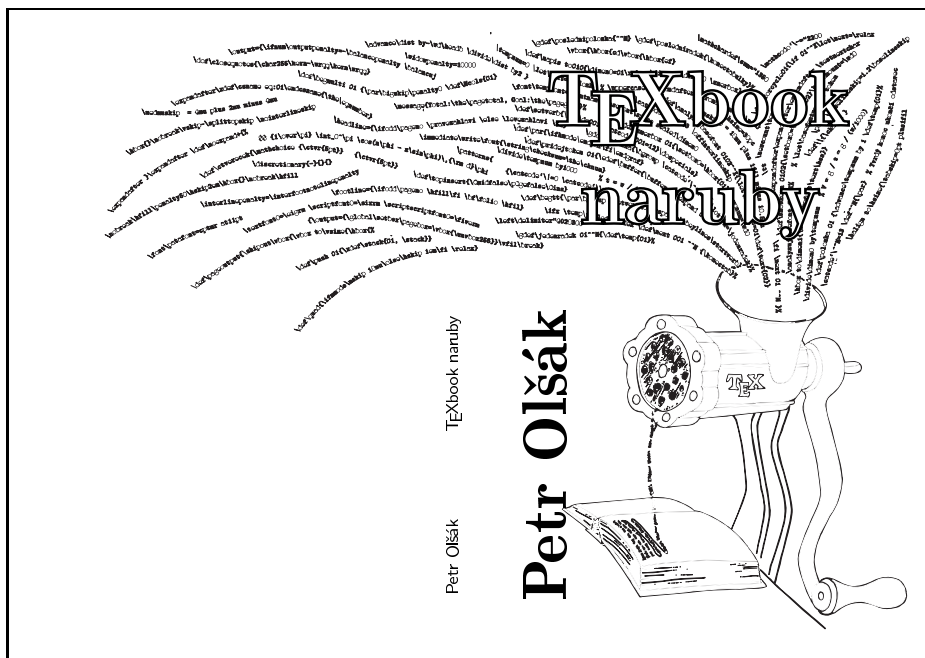
## Recenze knihy $\TeX$ book naruby

JAN PAZDZIORA

Dne 10. února oznámil pan Olšák českým příznivcům  $\TeX$ u prostřednictvím konference členů  $\zeta$ TUGu a newsové skupiny `cz.comp.cstex`, že dává k dispozici text své knihy  *$\TeX$ book naruby*. Kniha je dostupná na Internetu jako DVI, PDF a PS soubor, její URL je `http://math.feld.cvut.cz/olsak/tbn.html`.

Stejně jako v autorově knize *Typografický systém  $\TeX$*  se i zde jedná o výlet do světa tohoto sázecího systému. Tentokrát je ale zaměřen přímo na samotné jádro věci, program  $\TeX$ . Zabývá se jeho algoritmy, primitivními makry i makry formátu `plain` a jejich použitím, vysvětluje pravidla pro jejich správnou činnost a vše ilustruje na příkladech.

Hlavní část knihy se jmenuje *Algoritmy* a v sedmi kapitolách nás seznámí s hlavními částmi programu, kterými dokument při zpracování  $\TeX$ em prochází. Od vstupního procesoru přes expanzi maker, hlavní procesor (střeva programu), tvorbu tabulek a sazbu matematiky až po lámání a výstupní rutiny, s několika bonbónky navíc. Každá z těchto částí se podrobně věnuje jednomu kusu programu a vysvětlí jak jeho práci, tak souvislost s ostatními algoritmy, které  $\TeX$



používá. Vše je doprovázeno příklady – téměř na každé stránce najdeme kousek  $\TeX$ ovského kódu, na kterém autor prakticky předvede fakta uvedená v textu.

Druhá, menší část, má název *Reference* a ten již sám říká, co je jejím obsahem: abecedně uspořádaný přehled definic a syntaktických pravidel (například definice *glue* spolu možnými tvary zápisu), slovníček primitivních a plainovských maker, velmi užitečný seznam příkladů z knihy, literatura a rejstřík. I zde jsou definice často prokládány příklady typického použití, navíc u každého makra najdeme odkaz jak na podrobnější výklad v první části knihy či na jeho použití, tak i čísla stránek v *TeXbooku*, kde se o daném hesle hovoří.

Tím se dostáváme k často kladené otázce: pro koho je kniha určena a jaký rozdíl mezi *TeXbookem naruby* a jeho originálním bratříčkem? Pokusím se zde na tuto otázku odpovědět, protože by to mohlo dát čtenáři představu, zda si ji chce ze sítě stáhnout, když už přece jednu (či dvě?) knihy o  $\TeX$ u doma má a potřebu číst další zatím nepociťoval.

Pravděpodobně nezaujme začínající uživatele  $\TeX$ u. Autor předpokládá, že čtenář  $\TeX$  víceméně běžně používá a nabízí zde pomoc pro případ, že už typické *Začínáme s* nebo *Jemný úvod do* nestačí. Případ, kdy se program z nepochopitelného důvodu začne chovat jinak, než bychom si přáli, nebo kdy bychom si chtěli usnadnit práci napsáním makra, které ale přesáhne naše

uživatelské schopnosti. V takovém okamžiku již nestačí vědět, jak co (v ideálním případě) udělat, mnohem důležitější je pak informace, proč se program chová právě tak, jak se chová, jaký má k tomu důvod. A právě o těchto důvodech a souvislostech je tato kniha. S její pomocí se můžeme snadněji dostat ze situace, kdy tušíme, že by to tak nějak mělo fungovat, blíže k pozici, kdy víme, proč je odpověď  $\TeX$ u právě taková, jaká je.

Knih je velmi dobrou referencí, na rozdíl od originálního *TeXbooku*. Tím, že u definice makra najdeme jednak jeho krátký popis a vysvětlující příklady, jednak odkaz do první části knihy, kde se dozvíme o kontextu a souvislostech jeho použití, se knížka stala velmi přehlednou a užitečnou, pokud hledáme rychlou odpověď. V *TeXbooku* je informace o makru či funkci programu většinou rozprostřena po celé knize a její hledání je často dosti strastiplné. Vše se tam komplikuje také velmi beletristickým stylem výkladu i tím, že se prof. Knuth často dopouští svých malých „lží“, které uvádí na pravou míru třeba až několik kapitol později. V *TeXbooku naruby* se autor naopak snaží dát na jednom místě co možná kompletní a nejhutnější odpověď, přičemž možná tvrdost takového postupu je vyvážena praktickými příklady a ukázkami. A pro hodně českých a slovenských čtenářů může být dosti důležitým argumentem stále i jazyková stránka.

Pro toho, kdo to s  $\TeX$ em myslí jen trošku vážně a nechce se na něj dívat jako na černou skříňku, se *TeXbook naruby* může stát užitečným průvodcem a zdrojem informací. Mohla by zaujmout místo druhé knihy o  $\TeX$ u v rodině, hned za uživatelskými manuály. Je zde totiž shromážděno velké množství „low level“ informací, proložených praktickými příklady.

Jediným problémem zůstává neexistence její tištěné verze. I když je Internet stále dostupnějším médiem, je pravděpodobně stále hodně těch, pro které bude elektronická podoba překážkou. Doufejme, že tento stav je pouze dočasný a že se s *TeXbookem naruby* setkáme brzy i v papírovém provedení.

Jeho náhradou se může alespoň prozatím stát elektronická verze ve formátu PDF – Portable Document Format firmy Adobe. Ta byla vytvořena programem `tex2pdf`<sup>1</sup> a její hlavní výhodou je přenositelnost a „klikatelné“ odkazy v textu, které nás přenesou přímo na tu stranu a místo v knize, na které ukazují. Pohyb po knize dlouhé 468 stran se tak velmi zjednodušuje, což je u díla referenčního typu určitě přínosem.

*Jan Pazdziora*  
*Fakulta informatiky*  
*Centrum výpočetní techniky FI*  
*Botanická 68a*  
*602 00 Brno*  
*adelton@informatics.muni.cz*

---

<sup>1</sup>Han The Thanh: Alternativní výstup programu  $\TeX$  – PDF. Zpravodaj Československého sdružení uživatelů  $\TeX$ u, 6 (2), 69–85 (1996).

---

---

## Závazná objednávka knihy TeXbook naruby

---

Vzhledem k nízkému počtu zájemců o knihu TeXbook naruby není možné tuto knihu vydat tradičním způsobem. Na druhé straně je nesmyslné, aby si knihu tiskl každý zájemce sám. Jednak se tím zbytečně drancují společné prostředky (tiskárny, toner, papíry) akademických (případně) ústavů, a to není fér, jednak je stejně potřeba nechat takový výtisk svázat (nebo si ho svázat sám). MFF UK společně s  $\zeta$ TUGem se proto rozhodli udělat pro tuto věc maximum. Kniha bude vydána v počtu výtisků, který bude odpovídat počtu závazných objednávek na tuto knihu. Protože se předpokládá, že knihu si objedná více než jeden zájemce, lze prohlásit, že cena knihy bude rozhodně nižší, než cena, kterou by zaplatil jednotlivý zájemce za vlastními silami vyrobený výtisk.

Objednat knihu si může člen i nečlen  $\zeta$ TUGu. Je zde však cenový rozdíl 100 Kč. Členství pro studenty stojí v současné době rovněž 100 Kč, pro ostatní 200 Kč, nicméně všichni členové  $\zeta$ TUGu dostávají bulletin (4× ročně) a jsou jim poskytovány výrazné slevy (nejméně 10%) pro knihy a produkty, které se prostřednictvím organizace  $\zeta$ TUG distribuují. Členství se rozhodně vyplatí. Individuální členové  $\zeta$ TUGu si mohou objednat maximálně 1 výtisk knihy se slevou, kolektivní členové maximálně 6 výtisků.

V následující tabulce jsou uvedeny různé ceny knihy. Platit bude cena, která bude odpovídat celkové poptávce po knize. Samozřejmě čím vyšší poptávka, tím nižší cena. Celková poptávka ještě není známa. Tato závazná objednávka je tedy zároveň marketingovým průzkumem trhu. Každý zájemce závazně vyplní v následující tabulce počet kusů, které odebere při dané ceně knihy. Knihy lze objednávat i hromadně pro více lidí. V případě nárokování slevy pro členy  $\zeta$ TUGu je potřeba uvést seznam těchto lidí. Za zaplacení objednávky ručí objednavatel.

Příklad: 10 členů  $\zeta$ TUGu se domluví, že jeden z nich, pan Novák, hromadně objedná několik výtisků knihy TBN. Při ceně do 400 Kč si tato skupina objedná 5 výtisků, při ceně do 300 Kč si objedná 7 výtisků, při ceně do 200 Kč si objedná 10 výtisků a ještě 5 výtisků, na které již nelze uplatnit slevu. Pan Novák tedy vyplní tabulku následujícím způsobem:

Cena maximálně	Počet kusů pro členy	Počet dalších kusů
400 Kč	5	0
300 Kč	7	0
200 Kč	10	5

Cena bude nakonec stanovena na 250 Kč. Pan Novák tedy obdrží 7 výtisků knihy TBN za cenu 250 Kč za kus.

Objednávku pošlete pokud možno obratem, protože cenová kalkulace bude provedena k počtu objednávek na konci srpna.

Objednávka z přirozených důvodů není na  $\zeta$ TUGu vymahatelná, neboť kniha doposud nenašla vydavatele.  $\zeta$ TUG pomáhá vydání knihy mimo jiné její propagací (viz recenze v tomto čísle a článek o klikací knize autora), shromažďováním objednávek a v případě vydání pak odkoupením části nákladu a distribucí svým členům. V případě, že ztratíte vložený objednávací lístek, můžete si objednávku stáhnout a vytisknout ze serveru sdružení  $\zeta$ TUG <http://www.cstug.cz/>

---

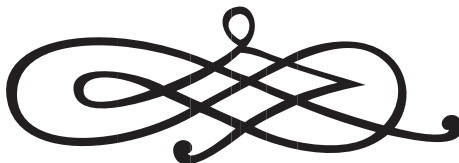
---

## TUGboat 17 (4) December 1996

---

<b>Addresses</b>	335	
<b>TUG'97 Update</b>	336	
<b>General Delivery</b>	337	<i>Michel Goossens</i> : From the President
	340	<i>Barbara Beeton</i> : Editorial Comments
		A historical perspective; The Kyoto Prize for Knuth; Recommended reading: Why things bite back; Why does a lion represent $\TeX$ ?
	342	Amsterdam, 13 March 1996 – Knuth meets NTG members
	355	$\zeta$ TUG, Charles University, Prague, March 1996 – Questions and answers with Prof. Donald E. Knuth
<b>Survey</b>	367	<i>Philip Taylor</i> : Computer typesetting or electronic publishing? New trends in scientific publication
<b>Software &amp; Tools</b>	382	<i>S. A. Strelkov and G. R. Epshtein</i> : Making indices for VINITI's "Mathematics" abstract journal
<b>Philology</b>	385	<i>Mikhail Ivanovich Grinchuk</i> : $\TeX$ and Russian traditions of typesetting

<b>Humanities</b>	388	<i>Christina Thiele: T<sub>E</sub>X and the Humanities</i>
<b>Book Review</b>	393	<i>Jonathan Fine: The Printing Revolution in Early Modern Europe, by Elizabeth L. Eisenstein</i>
<b>Queries</b>	396	<i>George Graetzer: Suggestions wanted for new edition of Math into L<sup>A</sup>T<sub>E</sub>X</i>
	396	<i>Dirk Kuypers: P<sub>P</sub>C<sub>H</sub>T<sub>E</sub>X molecule templates</i>
<b>Abstracts</b>	397	<i>Les Cahiers GUTenberg, Nos. 21–24</i>
<b>News &amp;</b>	403	<i>Calendar</i>
<b>Announcements</b>		
	405	<i>Mimi Burbank: Production notes</i>
	405	<i>Future issues</i>
<b>TUG Business</b>	406	<i>Mimi Burbank: Facts and Figures</i>
	409	<i>Institutional members</i>
<b>Forms</b>	410	<i>TUG membership application</i>
<b>Advertisements</b>	411	<i>T<sub>E</sub>X consulting and production services</i>
	411	<i>Index of advertisers</i>



## Zpravodaj Československého sdružení uživatelů T<sub>E</sub>Xu

ISSN 1211-6661

Vydalo: Československé sdružení uživatelů T<sub>E</sub>Xu  
vlastním nákladem jako interní publikaci

Obálka: Bohumil Bednář

Počet výtisků: 750

Uzávěrka: 13. června 1997

Odpovědný redaktor: Zdeněk Wagner

Tisk a distribuce: KONVOJ, spol. s r. o., Berkova 22, 612 00 Brno,  
tel. 05-740233

Adresa: ČSTUG, c/o FI MU, Botanická 68a, 602 00 Brno

fax: 05-412 125 68

e-mail: [cstug@cstug.cz](mailto:cstug@cstug.cz)

Zřízené poštovní aliasy sdružení ČSTUG:

[bulletin@cstug.cz](mailto:bulletin@cstug.cz), [zpravodaj@cstug.cz](mailto:zpravodaj@cstug.cz)

korespondence ohledně Zpravodaje sdružení

[board@cstug.cz](mailto:board@cstug.cz)

korespondence členům výboru

[cstug@cstug.cz](mailto:cstug@cstug.cz), [president@cstug.cz](mailto:president@cstug.cz)

korespondence předsedovi sdružení

[cstug-members@cstug.cz](mailto:cstug-members@cstug.cz)

korespondence členům sdružení

[cstug-faq@cstug.cz](mailto:cstug-faq@cstug.cz)

řešené otázky s odpověďmi navrhované k zařazení do dokumentu ČSFAQ

[secretary@cstug.cz](mailto:secretary@cstug.cz), [orders@cstug.cz](mailto:orders@cstug.cz)

korespondence administrativní síle sdružení, objednávky CD-ROM

[bookorders@cstug.cz](mailto:bookorders@cstug.cz)

objednávky tištěné T<sub>E</sub>Xové literatury na dobírku

ftp server sdružení:

<ftp://ftp.cstug.cz/>

www server sdružení:

<http://www.cstug.cz/>

Podávání novinových zásilek povoleno Českou poštou, s. p. OZJM Ředitelství  
v Brně č. j. P/2-1183/97 ze dne 11. 3. 1997.