

OBSAH

Petr Olšák: Pár poznámek k novému C \TeX u	1
Zdeněk Wagner: L \TeX 2 ϵ v C \TeX u	15
Petr Olšák: Program a2ac – manipulace s fontem na úrovni PostScriptu	23
Martin Pavlík: České třídění Exkurze do systémů zpracování textu	31
Libor Sýkora: Vkládání obrázků do L \TeX u	37
Pavel Herout: PostScriptové fonty pro ty, co o nich moc nevědí . .	43

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`). 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

Disketa musí být formátována pro DOS (Zpravodaj se sází v OS/2). Nezapomeňte přiložit všechny soubory, které dokument načítá (s výjimkou standardních součástí C \TeX u), zejména v případě, kdy vás nemohu kontaktovat e-mailem.

Trocha historie

Původní C_ST_EX vyšel z projektu Oldřicha Ulrycha se záměrem nabídnout uživatelům, kteří problematice (aspoň zpočátku) nechtějí rozumět příliš do hloubky, snadně obsluhovatelný instalační balík. Šlo o to, abychom pro člověka, který jedním dechem položí dvě otázky „co je to T_EX a mohl bych jej vyzkoušet na svém PC?“, měli připravenou nějakou odpověď na otázku první a *snadnou* odpověď na otázku druhou: „tu máš diskety a vyzkoušej si to“.

Hlavní pilíř práce spočíval tedy v sestavení funkční instalace pro DOS z veřejně dostupných PD (in public domain) programů. Původně se jednalo o programy z S_bT_EXu a později o programy z e_mT_EXu. Samozřejmě tam byla zahrnuta též práce programátorů z našich zemí. Bylo totiž potřeba doplnit převzaté programy o další nástroje pro pořizování dokumentů v českém a slovenském jazyce. Také byla snaha pokud možno co nejvíce ulehčit život řadovému uživateli. Znamená to, že uživatel by například neměl být nucen při běžných požadavcích číst mnohdy velmi rozsáhlou, nepřehlednou a nesourodou dokumentaci jednotlivých balíků PD programů, ze kterých cílový balík sestával.

V prehistorické době C_ST_EXu (1990–92) byla situace z jedné strany obtížná, protože plno věcí pro podporu češtiny a slovenštiny bylo teprve ve vývoji. Z druhé strany však byla situace jednodušší, protože pod pojmem „zkoušet T_EX na PC“ se jaksi samozřejmě očekávalo prostředí operačního systému DOS.

V roce 1993 jsem se zapojil do vytvoření nové verze C_ST_EXu, protože jsem považoval za vhodné zařadit do něj svůj program MNU, který umožnil vytvářet modifikovatelné a snadno ovladatelné nabídky pro spouštění jednotlivých T_EXovských úkolů. Program MNU spolupracuje s poměrně rozsáhlými DOSovskými dávkami. Je tedy patrné, že i tato verze instalace se opírala o jednouchvatelský a jednoúlohový operační systém DOS.

Jako autoři instalačního balíku jsme se tehdy domluvili na tom, že budeme každoročně začleňovat změny a opravy. Proto jsme tehdejší verzi $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u nazvali $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ 93 a o rok později s nepatrnými úpravami byl dán k dispozici $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ 94. Očekávalo se, že se v této řadě bude pokračovat...

Dnes máme pro PC různé operační systémy a ten převládající[†]) není shodný s tím, který mají původní autoři $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u v oblíbě. Navíc nemáme jenom PC, ale existují i jiné von Neumannovy strojky, na nich znovu jiné operační systémy a vše se spojuje do heterogenních, mnohdy dosti specifických, sítí. Pro tyto podmínky zatím nebyl $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ stavěný.

Mezi původní cíle $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u je třeba zařadit i snahu po unifikaci řešení problematiky české a slovenské sazby. To má dvě výhody: Za prvé, šetří se síly programátorů, kteří podporu pro národní sazbu vytvářejí, protože na řešení stejného problému nepracují bez vzájemné informovanosti na více místech současně. Za druhé, šetří se nervy uživatelů, kteří si vzájemně vyměňují $\mathcal{T}\mathcal{E}\mathcal{X}$ ovské dokumenty a zkušenosti s $\mathcal{T}\mathcal{E}\mathcal{X}$ em, protože všichni pracují se stejným produktem.

Tohoto cíle se bohužel nepodařilo dosáhnout. Dnes se dá těžko posoudit, kolik procent instalací $\mathcal{T}\mathcal{E}\mathcal{X}$ u, na nichž se pořizuje sazba v našich národních jazycích, tvoří $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$. Uděláme anketu aspoň mezi členy $\mathcal{C}\mathcal{T}\mathcal{U}\mathcal{G}$ u? Pravděpodobně je procento vysoké, ale existují i lokální instalace postavené na jiných principech řešení národní problematiky. Některé takové instalace dokonce přesahují lokální charakter svého působení. Uvedme například instalaci pana Kasprzaka, která je navíc šířena prostřednictvím Internetu.

O názvu

Logo $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ je zkratkou „český a slovenský $\mathcal{T}\mathcal{E}\mathcal{X}$ “, což asi každý ví. Číslo verze za logem odpovídá ročníku, kdy byla verze připravena a šířena. Očekávalo se, že bude existovat $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ 95, protože předcházely verze $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ 93 a $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ 94. Bohužel, pro názorový nesoulad mezi účastníky diskusní skupiny „tzv. tvůrců $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u“ a pro následný absolutní nezájem těchto účastníků přiložit ruku k dílu, když už byla stanovena koncepce a když jsem zveřejnil uvnitř této skupiny alfa verzi, se příprava na verzi 95 protáhla a byla uvedena až v roce 1996. Dokumentace se na současnou

[†])MS Windows

verzi $\zeta\text{T}_{\text{E}}\text{X}$ většinou odvolává pod číslem 95, někde se píše $\zeta\text{T}_{\text{E}}\text{X}95,6$. Uživatele je třeba upozornit, že mezi $\zeta\text{T}_{\text{E}}\text{X}$ em 95 a předchozími verzemi je hodně dosti zásadních rozdílů, viz níže.

Protože od nynější verze platí úmluva, že se jednotlivé moduly v $\zeta\text{T}_{\text{E}}\text{X}$ u budou opravovat průběžně, nebudou vznikat další verze s čísly podle ročníků. Z toho důvodu bude asi jednodušší říkat nejnovější verzi prostě $\zeta\text{T}_{\text{E}}\text{X}$. Předchozí verze by se měly pro odlišení důsledně nazývat $\zeta\text{T}_{\text{E}}\text{X}94$, $\zeta\text{T}_{\text{E}}\text{X}93$ či $\zeta\text{T}_{\text{E}}\text{X}92$. Tuto fintu, a do jisté míry podraz vůči uživatelům, už známe z produktu současného $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ teamu. Produkt se překvapivě jmenuje $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, zatímco to, co jsme byli až dosud zvyklí $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ em nazývat, máme nyní pojmenovávat jako $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2.09$. V dalším textu budeme tedy slovem $\zeta\text{T}_{\text{E}}\text{X}$ označovat verzi $\zeta\text{T}_{\text{E}}\text{X}95,6$.

Žádná jiná varianta počestění $\text{T}_{\text{E}}\text{X}$ u by se neměla nazývat stejným jménem. Pouze ζTUG em oficiálně podporovaná verze má toto jméno. Totéž platí o formátech $\text{T}_{\text{E}}\text{X}$ u, které jsou zahrnuty v $\zeta\text{T}_{\text{E}}\text{X}$ u. V současné době jde o jména `csplain`, `csamstex`, `cslamstex`, `cslatex`, `csamslat` a `cs1t209`. Pokud například najdeme na síti $\text{T}_{\text{E}}\text{X}$ ovský dokument, který bude mít v úvodním komentáři napsáno „zpracovat formátem `csplain`“, mělo by tím být naprosto jednoznačně řečeno, o co jde.

Vymezení pojmu a struktura $\zeta\text{T}_{\text{E}}\text{X}$ u

Název bychom měli, teď si ještě povíme, co to vlastně ten $\zeta\text{T}_{\text{E}}\text{X}$ je. Dříve jsme měli situaci jednodušší, protože jsme většinou tento název ztotožňovali s oněmi devatenácti disketami pro DOS. Ovšem i při takovém pojetí jsme se dopouštěli některých omylů a nepřesností. Třeba tvrzení, že $\zeta\text{T}_{\text{E}}\text{X}$ je $\text{emT}_{\text{E}}\text{X}$, je omyl. Nebo jiné tvrzení, že pod pojmem $\zeta\text{T}_{\text{E}}\text{X}$ si představím určité seskupení nabídek realizovaných programem MNU, také není v pořádku.

V současné době, kdy s $\zeta\text{T}_{\text{E}}\text{X}$ em pracují uživatelé různých operačních systémů (OS), je třeba vymezit tento pojem poněkud opatrněji.

$\zeta\text{T}_{\text{E}}\text{X}$ pro daný OS se skládá z těchto komponent:

1. Základ – $\text{T}_{\text{E}}\text{X}$, makra, fonty, ovladače. Fungující pro anglický jazyk.
2. Prostředky pro podporu č/s sazby závislé na OS.
3. ζ -fonty.

4. Makra pro sazbu v národním prostředí a další podpora nezávislá na OS.

Ad 1. Základ většinou vzniká kompilací Knuthova zdrojového textu `tex.web` do spustitelného programu pro daný OS. Totéž platí pro `META-FONT`. Dále jsou součástí základu Knuthovy fonty ve tvaru `mf` a makra společně s makry dalších autorů. Například makra \LaTeX U. V neposlední řadě základ musí zahrnovat `dvi` ovladač pro zobrazovací jednotku v daném OS, tzv. prohlížeč, a ovladače pro nejběžněji používaná výstupní zařízení.

Nebývá obvyklé, aby tento základ vznikl v dílnách českých autorů. Jako základ se používají PD instalace \TeX U pro daný OS. Například pro DOS a OS/2 se používá jako základ `emTeX`, pro UNIX je použita \TeX ovská instalace Karla Berryho. Základ není vhodné označovat jako součást $\zeta\TeX$ U ve smyslu vykonané práce autorů $\zeta\TeX$ U, ovšem z pohledu skutečnosti, že bez základu nám to nepojede, zřejmě se bez označení „součást $\zeta\TeX$ U“ mnohdy neobejdeme.

Základ určuje způsob balení dalších komponent, tj. např. strukturu použitých adresářů, způsob a typ použitého programu pro vytváření instalačních modulů (například `zip` nebo `tar`), způsob prezentování dokumentace a případně další náležitosti.

Ad 2. Abychom mohli pořizovat česky a slovensky psané dokumenty, potřebujeme některé další prostředky, které jsou závislé na použitém OS. Například je potřeba zodpovědět otázku, jak a zda vůbec řeší OS pořizování zdrojových textů \TeX U s národními abecedami. Jedná se třeba o volbu editoru a pomůcek k pořizování textů včetně nábodeníček. Také zde patří různé konvertory, preprocesory (třeba program `vlnka`), kontroly překlepů apod. V každém OS bude asi tato podpora na jiné úrovni. V některém OS můžeme mít navíc různá „vylepšení“, například program `MNU`.

Ačkoli se některé věci z této oblasti dají převzít z nabídky PD programů, přesto je potřeba udělat v daném OS kus samostatné programátorské práce. Vše, co kdokoli pro svůj OS udělá, je v projektu $\zeta\TeX$ vítáno. Například pro UNIXy se v současné době dají použít fonty v ISO8859-2 zavedené do X aplikace (například editoru). V případě Emacsu se dá vkládání nábodeníček řešit na úrovni `maker` editoru. Jiná možnost je využít přepínání tabulky `Xmodmap` na úrovni X serveru. Po-

kud například tyto softwarové pomůcky nabídnou autoři do $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u pro UNIXy, budeme potěšeni.

Ad 3. V každé instalaci $\mathcal{T}\mathcal{E}\mathcal{X}$ u jsou za základní fonty považovány Knuthovy fonty Computer Modern. Nic na tom nemění skutečnost, že instalace většinou umí pracovat s rozsáhlým množstvím dalších, obvykle PostScriptových, fontů. Jako základní fonty pro českou a slovenskou sazbu jsou v $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u stanoveny $\mathcal{C}\mathcal{S}$ -fonty.

$\mathcal{C}\mathcal{S}$ -fonty jsou k dispozici ve stejném formátu, jako Computer Modern fonty a navazují na ně. Jedná se tedy o formáty `mf` a `tfm`, které jsou přenositelné na libovolný OS.

Ad 4. Makra $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u navazují na makra základu a na $\mathcal{C}\mathcal{S}$ -fonty. Ve všech případech jde o textové soubory přenositelné na libovolný OS. Makra modifikují formáty, převzaté ze základu (například `plain`, `latex`) tak, aby se v nich místo Computer Modern fontů pracovalo s $\mathcal{C}\mathcal{S}$ -fonty. Dále jsou v těchto makrech zahrnuty české a slovenské vzory dělení slov. Výsledné formáty mají před původním názvem předponu `cs`, tj. třeba `csplain`, `cslatex`. Toto jsou oficiální názvy formátů $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u.

Kromě maker existují další pomůcky pro českou a slovenskou sazbu, nezávislé na OS. Uveďme například virtuální fonty v kódování $\mathcal{C}\mathcal{S}$ -fontů, které umožňují sazbu PostScriptovými fonty v národních jazycích.

„ $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ pro jistý OS“ představuje relativně samostatný celek a je nabízen na ftp serverech v samostatných adresářích. Rozličný může být i způsob distribuce a způsob instalace. Tyto věci se většinou podřizují způsobům, jaké jsou použity při instalaci základu.

První dvě komponenty jsou závislé na operačním systému, zatímco další dvě komponenty se v různých OS neliší. Proto třetí a čtvrtou komponentu nazýváme *jádrem* $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u, které je ve všech OS „skoro“ shodné. Slůvko „skoro“ v sobě zahrnuje pouze nutné transformace textových souborů, struktury adresářů a názvů souborů z důvodu specifik daného OS: máme různé konce řádků pro MAC, DOS a UNIX a některé méněcenné OS nerozlišují velikosti písmen v názvech souborů a mají velmi nepříjemné omezení v délce názvu.

V současné době nabízíme $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ pro DOS a OS/2. Základem tohoto $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u je `em $\mathcal{T}\mathcal{E}\mathcal{X}$` , který je společný pro oba OS. Nad touto instalací lze použitím balíku `texwin.zip` vytvořit též podporu pro MS Windows. Dále existuje $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ pro UNIXy, ovšem zatím jen v plenkách. Základem je instalace Karla Berryho. Nabízené TARy nyní obsahují jen převedené

jádro $\zeta\text{T}_{\text{E}}\text{X}$ u zabalené do struktury adresářů podle základu. Zatím pro UNIX chybí druhá komponenta. Po jednání s autory softu spadajícího do druhé komponenty, se pokusím do $\zeta\text{T}_{\text{E}}\text{X}$ u pro UNIX zařadit i jejich práci. Dále rád přivítám jakoukoli spolupráci s uživateli Amigy, MACu, VAXu apod. Nabízím odbornou konzultaci k problematice $\text{T}_{\text{E}}\text{X}$ u a okolí, ovšem o operačních systémech samotných vím velmi málo. Dočkáme se v brzké době rozšíření $\zeta\text{T}_{\text{E}}\text{X}$ u pro další OS?

Výhody proti předchozí verzi $\zeta\text{T}_{\text{E}}\text{X}94$

V následujících dvou odstavcích si shrneme výhody a nevýhody nové verze vzhledem ke staré. Tím odpovíme na otázku, zda má pro uživatele smysl přecházet na novou verzi $\zeta\text{T}_{\text{E}}\text{X}$ u. Nejvýznamnější výhody se dají shrnout do následujících bodů. První dva body se týkají srovnání $\zeta\text{T}_{\text{E}}\text{X}$ u pro DOS, ostatní body mají obecnější platnost.

1. Lepší kompatibilita se současným $\text{emT}_{\text{E}}\text{X}$ em.
2. Končí (snad) problémy s tex386 kontra různé DOSovské správy paměti.
3. Je implicitně zařazeno automatické dogenerování fontů.
4. Je zařazen cslatex jako c/s varianta $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\epsilon}$.
5. Možnost použít PostScriptové fonty i v národní sazbě.
6. Nové vzory dělení slov pro češtinu.
7. Existuje WWW stránka s přehledným rozdělením $\zeta\text{T}_{\text{E}}\text{X}$ u do modulů.

Ad 1. V původním balíku byl základ $\text{emT}_{\text{E}}\text{X}$ přebrán jen částečně a byl kompletně přebalen s ohledem na potřeby instalace $\zeta\text{T}_{\text{E}}\text{X}$ u. Časem se objevily dvě nevýhody. Za prvé to odporovalo požadavku autora $\text{emT}_{\text{E}}\text{X}$ u E. Mattese distribuovat $\text{emT}_{\text{E}}\text{X}$ pouze jako celek. Za druhé, E. Mattes mezitím přešel k novějším verzím a dramaticky změnil adresářovou strukturu i syntaxi konfiguračních souborů^{*)}. Tím se distribuovaný $\zeta\text{T}_{\text{E}}\text{X}$ značně vzdálil od aktuální verze svého základu a nastaly problémy. Když si někdo vzpomněl, že chce tisknout jazykem PCL na 600 dpi tiskárnách, nefungovalo to. Když si sehnal novější verzi ovladače,

^{*)}Takové zásadní změny se v $\text{emT}_{\text{E}}\text{X}$ u odehrávají zhruba s frekvencí dvou let. Není to poprvé, co se s tím setkáváme, bohužel.

kde byla chyba odstraněna, byl zmaten novým jazykem konfiguračního souboru a strukturou adresářů. Teprve za cenu obrovského úsilí spojil starý $\mathcal{C}\text{T}_{\text{E}}\text{X}$ s novým programem do fungujícího celku. Podobná věc platila pro bublinkové tiskárny, nové verze grafických karet monitoru apod. Tyto věci ve starém $\mathcal{C}\text{T}_{\text{E}}\text{X}$ u podporovány nebyly a kombinací nových ovladačů $\text{em}_{\text{T}}\text{E}\text{X}$ u se starým $\mathcal{C}\text{T}_{\text{E}}\text{X}$ em přibývaly potíže.

Protože lze předpokládat, že v budoucnu dojde k dalším změnám v $\text{em}_{\text{T}}\text{E}\text{X}$ u, bude pro snadnější obnovování $\mathcal{C}\text{T}_{\text{E}}\text{X}$ u pro DOS jednodušší převzít základ beze změn s originálně balenými ZIPy.

Ad 2. Potěším všechny zoufalce, kteří chtějí s DOSem obsloužit více než 640 kb paměti pomocí různých berliček, jako je `emm386` a jim podobné. To s původní verzí $\text{em}_{\text{T}}\text{E}\text{X}$ u nešlo moc dohromady. Současná verze $\text{em}_{\text{T}}\text{E}\text{X}$ u už obsahuje příslušné `exe` programy (např. `tex386.exe`) ve „vykuchaném“ tvaru. Znamená to, že tyto programy nemají obsluhu rozšířené paměti zahrnutu přímo v sobě. Místo toho uvedené programy spolupracují s dvěma různými podpůrnými nástroji – `emx` nebo `rsx`. Nástroj `emx` obsahuje původní Mattesovo pojetí práce s rozšířenou pamětí, zatímco `rsx` dokáže spolupracovat s `emm386`. Typ podpůrného nástroje si program vybere automaticky v závislosti na přítomném (nebo nepřítomném) emulátoru paměti. Důsledkem toho je, že se to nepere s Microsoftským pojetím obsluhy rozšířené paměti.

Nedá se předpokládat, že problém paměti v DOSu je navždy vyřešen. Různé novější verze zpětně nekompatibilních obsluhovačů paměti mohou znovu způsobit problémy. Budiž pouze útěchou, že se nemusí kvůli tomu (snad) měnit verze programů `tex386.exe` a jim podobných, ale stačí vyměnit příslušný `rsx`. Problémy tohoto typu budou existovat tak dlouho, jak dlouho bude existovat DOS. Zvrácenost tohoto operačního systému spočívá v tom, že nechává na aplikaci, aby se sama nějak vypořádala se vstupem do rozšířené paměti. Pokud na stejném hardware spustíte nějaký UNIX, bariéra 640 kb přestává existovat a operační systém nabízí aplikaci tolik virtuální paměti, kolik se konfiguruje.

Ad 3. V $\mathcal{C}\text{T}_{\text{E}}\text{X}$ u 94 zabíraly převážné množství distribučních disket bitové mapy \mathcal{C} -fontů pro nejrůznější tiskárny a v nejrůznějších velikostech. Automatické dogenerování bitmap \mathcal{C} -fontů pak nebylo skoro potřeba provádět a bylo implicitně vypnuté. Dnes se používá větší množství typů tiskáren a větší množství fontů. Automatické dogenerování bitmap je proto implicitně zapnuté. To má pouze tu nevýhodu, že administrátor

sítě, který instaluje $\zeta\text{T}_{\text{E}}\text{X}$ do místa zabezpečeného proti zápisu uživatelem, musí pozměnit obsah konfiguračních souborů, aby se nově generované bitmapy vytvářely v místě, kam uživatel zapisovat může. Pokud se vychází z malého množství hotových bitových map, existuje ještě druhá nevýhoda. Tou je skutečnost, že se na fonty zpočátku musí chvíli počkat.

Aby byla trochu kompenzována druhá nevýhoda, jsou v $\zeta\text{T}_{\text{E}}\text{X}$ zařazeny knihovny bitových map pro původní čtyři typy podporovaných tiskáren, ovšem tyto knihovny zahrnují jen nejpoužívanější minimum. Proto nejsou tak rozsáhlé. Vše mimo tyto výchozí knihovny projde automatickým generováním.

Ad 4. V nové verzi $\zeta\text{T}_{\text{E}}\text{X}$ jsme se dočkali převedení $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 2 ϵ do národního prostředí. Tento formát se nyní nazývá `cslatex`. V $\zeta\text{T}_{\text{E}}\text{X}$ pro DOS lze navíc použít původní formát $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 2.09 a lze si snadno z nabídky vybrat, která verze $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ bude použita.

Výhody nového $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 2 ϵ vám neřeknu, protože $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ nepoužívám. Kdo jej používá, zřejmě by našel celou řádku argumentů, proč přejít na $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 2 ϵ .

Ad 5. V novém $\zeta\text{T}_{\text{E}}\text{X}$ máte možnost použít základní skupinu PostScriptových fontů. Jsou zařazeny virtuální popisy těchto fontů ve stejném kódování, jako ζ -fonty. Rovněž jsou k dispozici makra pro `cslatex` i `csplain`, která umožní snadné použití těchto fontů koncovému uživateli.

Základní skupina PostScriptových fontů je instalována v každém PostScriptovém RIPu, takže ji kupujete společně s RIPem v tiskárně. Do $\zeta\text{T}_{\text{E}}\text{X}$ budou zahrnuty v nejbližší době též knihovny bitových map těchto fontů pro nejběžnější tiskárny v obvyklých rozlišeních. Potom bude možné si sazbu s těmito fonty prohlížet a tisknout na levných tiskárnách. V činnosti ale nebude automatické generování nestandardních velikostí, protože popisy kreseb těchto fontů jsou chráněny licencí a není možné je šířit zdarma.

Pokud použijete rastrovací mechanismus těchto fontů z PD instalace Ghostscriptu, tj. místo originálních a licencí chráněných fontů ve formátu `pfb` použijete volně šířené `gsf`, velmi vás prosím, nevystavujte takové dokumenty na veřejnosti. Dělal byste $\text{T}_{\text{E}}\text{X}$ u jen ostudu.

Je též přibalen nástroj pro vytvoření dalších virtuálních popisů libovolných PostScriptových fontů. Je jím program `a2ac`. Pokud si tedy koupíte PostScriptový font ve Standard AdobeEncoding, pak máte možnost jej použít v české sazbě v $\text{T}_{\text{E}}\text{X}$ u. Doporučuje se nekupovat počestěné

varianty těchto fontů, protože velmi často platí, že počestěný font znamená pomršený font. Je daleko lepší využít pro počestění nástrojů z $\zeta\text{T}\text{E}\text{X}$ u.

Ad 6. V novém $\zeta\text{T}\text{E}\text{X}$ u jsou použity nové vzory dělení českých slov od Pavla Ševečka a kol. Tyto vzory jsou podstatně lepší, než vzory dělení z $\zeta\text{T}\text{E}\text{X}$ u 94. Slovenské vzory dělení zatím bohužel zůstaly beze změny. Vzory dělení (české i slovenské) jsou navíc uloženy v souborech nezávislých na volbě kódování, tj. neovlivní případně jinou volbu vnitřního kódování TEX u a jsou snadno přenositelné do jiných OS.

Ad 7. Existuje WWW stránka, která informuje o současném stavu $\zeta\text{T}\text{E}\text{X}$ u a přehledně ukazuje rozdělení jednotlivých instalací do modulů (například instalačních ZIPů). Prostřednictvím WWW lze též zkopírovat přímo do vašeho počítače libovolný modul $\zeta\text{T}\text{E}\text{X}$ u nebo kontaktovat autory modulů elektronickým dopisem.

Nevýhody proti předchozí verzi $\zeta\text{T}\text{E}\text{X}94$

1. Neexistuje snadná obnova předchozí verze na novou.
2. Větší nároky na disk.
3. Méně komfortní instalační procedura.
4. Méně kladen důraz na DOS.
5. Menší spolehlivost na rozhraních kompetencí.
6. Opravy se budou dělat za chodu.

Ad 1. Předchozí verze $\zeta\text{T}\text{E}\text{X}$ u (94 a 93) se od nové velmi výrazně liší v adresářové struktuře. Proto při přechodu na novou verzi je nutné zcela opustit verzi starou. Chcete-li mít obě verze vedle sebe, přejmenujte starý adresář na jiný název než `emtex` a instalujte nový $\zeta\text{T}\text{E}\text{X}$ vedle. Po přejmenování staré verze bude ještě potřeba pozměnit obsah i jména spouštěcích dávek staré verze a dále upravit set pro `TEXDIR` v `cfg\texset.bat`.

Ad 2. Protože jsme do $\zeta\text{T}\text{E}\text{X}$ u zařadili kompletní `emTEX`, nastávají problémy s velikostí instalace na disku. Tvůrce `emTEX`u Eberhard Matthes se tímto systémem intenzívně zabývá, a tak mu asi je jedno, že jeho instalace zabírá 50 MB. Méně jedno to je náhodnému uživateli, který na

svém disku kromě $\text{T}_{\text{E}}\text{X}$ u chce mít plno dalších věcí a $\text{T}_{\text{E}}\text{X}$ považuje jen jako nezávazný experiment.

Mattes balí do svých ZIPů skupiny software, které se paralelně doplňují. Například `exe` programy pro DOS i OS/2 jsou ve společném ZIPu. Nebo dokumentace je jednak anglická a jednak německá. Až poté, co je ZIP rozbalen na disk se tedy může začít šetřit místem vymazáním zbytečného softu. Například vymazání všeho, co se týká OS/2, protože máme zatím jenom DOS, případně vymazání veškeré německé dokumentace, protože raději čteme anglicky.

Jako kompromis byly vytvořeny tři varianty, definující množinu použitých ZIPů pro instalaci. Nejmenší varianta „small“ instaluje funkční minimum. To vyjde na 17 MB. Pak se dá například vymazat vše, co patří OS/2 a není použitelné pro DOS a dále německá dokumentace a vše se zredukuje na 13 MB. Pokud dále problematice velmi rozumíme a víme naprosto přesně, co budeme potřebovat a co ne, dají se ručně dále vymazávat jednotlivosti, až dojdeme obvykle k 8 MB. Pokud odmažeme $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ a necháme jen plain, dostaneme se na hranici 5 MB. Menší to rozhodně nikdy nebude. Protože v průběhu práce se budou generovat formáty a dále fonty, je rozumné pro minimální nároky počítat obvykle s 20 MB. Další varianty instalace „all“ a „big“ zaberou po řadě zhruba 38 MB a 50 MB.

Ad 3. Způsob balení Mattesových ZIPů tedy znemožnil elegantní instalaci jen těch částí $\zeta\text{T}_{\text{E}}\text{X}$ u, které skutečně potřebujeme. Proto nemělo smysl pro novou verzi $\zeta\text{T}_{\text{E}}\text{X}$ u použít analogii k instalačnímu programu, který známe ze starších verzí.

Ad 4. Až dosud byla práce autorů v $\zeta\text{T}_{\text{E}}\text{X}$ u nejprve inspirována jejich vlastními potřebami. Současná verze $\zeta\text{T}_{\text{E}}\text{X}$ u pro DOS je asi prvním balíkem, kdy jsem jako autor balíku nevytvořil toto dílo pro svou vlastní potřebu, ale pro druhé. Cítím totiž závazek vůči uživatelům DOSu, kteří novou verzi $\text{e}\text{T}_{\text{E}}\text{X}$ u zapracovanou do $\zeta\text{T}_{\text{E}}\text{X}$ u potřebují. Skutečně, na své domácí 286 zůstanu u verze z $\zeta\text{T}_{\text{E}}\text{X}$ u 94, která se mi podařila vměstnat včetně starého $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u do 5 MB na disku. Každý, kdo má pouze 40 MB disk ví, co to znamená „disková tíseň“. Tam se prostě nová verze nevejde a ani se nedá zkoušet. V zaměstnání bych asi také nebyl uživateli pochválen, kdybych přešel na novou verzi, na kterou nejsou zvyklí. A já osobně v zaměstnání používám $\text{T}_{\text{E}}\text{X}$ skoro výhradně na UNIXu.

Důsledkem tohoto odklonu autorových potřeb od odvedené práce může být, a pravděpodobně je, méně propracovaný a určitě méně prozkoušený balík.

Ad 5. Jednotlivé části \LaTeX u dělají různí autoři pod různými operačními systémy. Výsledná práce pak při opomenutí nemusí zcela zapadat do koncepce \LaTeX u a vznikají potíže na „rozhraních kompetencí“. Příkladem je současný balík `cslatex`, který vytvořil Z. Wagner a odzkoušel pod svým operačním systémem, kterým je OS/2. Přitom zcela ignoroval skutečnost, že v balíku MNU pro DOS jsou pro `cslatex` připraveny položky v menu a dávky. Tyto dávky ovšem s dodaným balíkem neuměly pracovat. Vznikla otázka, zda změnit dávky nebo balík `cslatex`. Ani já, ani on, jsme neměli možnost ani chuť vyzkoušet, zda naše dílo na sebe navazuje, protože nikdo z nás příliš nepracuje s DOSem. V následující kapitole se proto pokusím vymezit požadavky na autora modulu, co musí udělat, aby byl modul bezproblémově zařazen do \LaTeX u.

Ad 6. Je mi jasné, že pro uživatele nebude příliš snadné sledovat postupný pohyb ve verzích jednotlivých modulů a jejich opravy, přestože každá významnější změna v \LaTeX u bude zveřejněna na listu. V souboru `zmeny.txt` přístupném jednak pomocí ftp a jednak pomocí WWW bude přibývat podrobný záznam o jednotlivých změnách včetně data zanesení změny na serveru. Toto opatření, na rozdíl od skokových změn jednou ročně, umožní autorům provádět opravy modulů v okamžiku, kdy na to mají náladu a chuť a já je nebudu muset nutit ke splnění časových termínů, které podle mých zkušeností stejně neplnili. Pravda, je to poněkud méně uživatelsky přehledné, bohužel.

Jak zařadit nový modul

I vy se můžete stát spoluautory \LaTeX u. Když vytvoříte instalační modul (například ZIP), který je v souladu se stávající strukturou \LaTeX u pro daný OS a je jednoduše začlenitelný do stávajícího stavu, je to jedině vítáno. Je ovšem bezpodmínečně potřeba, abyste si zkusmo instalovali současnou verzi \LaTeX u pro daný OS a na něm provedli experiment začlenění vašeho modulu. Při tomto začlenění je potřeba zaznamenat úkony a změny, které je nutné udělat v souborech pocházejících z rukou jiných autorů. Tyto změny je pak potřeba podrobně popsat a zaslat

těmto autorům. Až je autoři začlenění do svých modulů, je teprve možné nový modul začlenit do $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u.

Příklad. Budete chtít přidat modul Ghostscriptu `gs.zip` pro $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ pro DOS. Provedete následující:

1. Instalujete si *současnou* verzi $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u pro DOS.
2. Instalujete svůj balík.
3. Provedete zásahy do stávajících souborů (třeba přidání položky v MNU).
4. Odzkoušíte funkčnost instalace včetně vašeho balíku.
5. Zašlete autorovi změněných souborů soupis změn (například mě zašlete instrukci o začlenění nové položky do `cfg.mnu` a do instalačních dávek).
6. Já provedu potřebné změny a vystavím obnovený modul `mnu.zip` a váš modul `gs.zip`. Také provedu změny v instalačních dávkách, případně v dokumentaci. Nepočítejte s tím, že to budu znovu zkoušet!
7. Vy dáte anonci o novém modulu na listu `cstex` a budete na tomto listu odpovídat na případné problémy a dotazy uživatelů.

Samozřejmostí dobře připraveného modulu je nejen správná struktura modulu, ale také dobrý a instruktivní dokumentační soubor v češtině nebo ve slovenštině. Také je třeba (například v dokumentaci) rozlišit vaši práci od práce převzaté ve formě převzatých PD programů a modul by neměl být v rozporu s požadavky autorů převzatých PD programů.

Je to dost hodně práce, ale když už takovou věc jednou uděláte pro sebe, asi stojí za to ji zveřejnit pro druhé pod hlavičkou $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u. Ušetříte tím práci mnoha dalším lidem, kteří bohužel, jak plyne z mých zkušeností, vám za to ani nebudou příliš vděční.

Zařazení některých nových modulů může být nejprve předmětem diskuse s autory konkurenčních produktů. Například zařadit do $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u pro DOS alternativní systém k systému MNU může vyžadovat delší diskusi. Netvrdím, že bych byl z důvodu své ješitnosti zásadně proti. Asi by záleželo na okolnostech a na projektu výsledného uspořádání modulů.

Co s pátou disketou

Kromě PD programů se v některých komponentách \LaTeX u mohou vyskytovat i programy vázané licenci. Takové programy pak \LaTeX UG rozšiřuje podle podmínek stanovených smlouvou mezi autorem, resp. distributorem a \LaTeX UGem.

V roce 1992 \LaTeX UG koupil editor \LaTeX Ed a korektor překlepů, obojí pro DOS. Od té doby k žádným dalším nákupům ani přechodům na novější verze nedošlo. Současné vedení \LaTeX UGu neinklinuje k příliš rozsáhlé podpoře nákupu podpůrných programů pro \TeX , zvláště pokud se jedná o software závislý na použitém OS.

Zmíněný editor \LaTeX Ed a korektor překlepů je podle smluv s autory možné bezplatně šířit jen pro členy \LaTeX UGu a není dovoleno jej distribuovat mimo řady členské základny. Tento software je zařazen na páté disketě v \LaTeX U pro DOS. Zatímco celý \LaTeX je k dispozici na anonymních ftp serverech, obsah této páté diskety z uvedených důvodů přístupný není.

Kolektivní členové \LaTeX UGu dostali v roce 1993 celý \LaTeX 93 na disketách, včetně páté diskety. Individuální členové si pak mohli obsah páté diskety kopírovat od kolektivních členů. Další možností je distribuence obsahu páté diskety prostřednictvím Internetu za použití osobního identifikačního čísla (PIN), které na požádání přidělí členovi \LaTeX UGu Martin Bílý.

V novém \LaTeX U je obsah páté diskety z pohledu zastoupených souborů zcela stejný jako v \LaTeX U94, ovšem ZIPy jsou zabalené s nepatrně odlišnou strukturou. Uvnitř ZIPu je přidán hlavní adresář `emtex`, aby struktura těchto ZIPů souhlasila se strukturou ostatních ZIPů v instalaci. Máte-li pátou disketu z \LaTeX U 94 nebo 93, pak stačí přebalit ZIPy do nové struktury a můžete disketu používat i v současném \LaTeX U. Pro přebalení je v \LaTeX U připravena dávka `newcs5.bat`, kterou je možno na „obnovení“ diskety použít. Upozornění: Provedte nejprve zálohu původní diskety.

Kde najdeme novou instalaci?

Instalace \LaTeX U je k dispozici na ftp serveru s anonymním přístupem `math.feld.cvut.cz` v adresáři `/pub/cstex`. Tam jsou v současné době

adresáře `msdos`, `os2` a `unix`. Přitom první dva adresáře jsou zcela totožné (UNIXový link), protože distribuce pro DOS a OS/2 vychází ze stejného základu `emTEX`.

Distribuce pro MS Windows je realizována jedním doplňkovým modulem `texwin.zip`, který je součástí `CSTEX` pro DOS.

Dále jsou k dispozici soubory `diskuse.txt` a `zmeny.txt`. V prvním je souhrn nejpodstatnějších diskusních příspěvků, týkajících se `CSTEX` na diskusní skupině `cstex`. V druhém je podrobný seznam změn včetně data provedení změny v jednotlivých modulech. Důvody, proč byla změna provedena, je mnohdy možné vyčíst ze souboru `diskuse.txt`.

Při zkopírování `CSTEX` z ftp serveru doporučuji uchovat si soubor `zmeny.txt`, neboť tento soubor vlastně charakterizuje verzi `CSTEX`, kterou jste si zrovna zkopírovali. Protože jsou změny prováděny „za chodu“, můžete v budoucnu porovnat svůj soubor `zmeny.txt` s vystavovaným a máte přehled o rozdílech mezi vaší verzí `CSTEX` a vystavovanou.

`CSTEX` je rovněž přístupný prostřednictvím WWW na

<http://math.feld.cvut.cz/olsak/cstex>.

Server `math.feld.cvut.cz` je primárním serverem `CSTEX`. Ostatní ftp servery jej mohou zrcadlit.

`CSTEX` pro daný OS má většinou specifický způsob instalace a mnohdy též distribuce. Například `CSTEX` pro DOS a OS/2 může být šířen též na disketách 1.44MB. Proto je zde připravena dávka `makeinst.bat`, která z adresáře, v němž jsou všechny moduly pohromadě, překopíruje instalační soubory na jednotlivé diskety s označením CS-1 až CS-5, EM-1 až EM-7 a PK-1 až PK-3. Tím vzniká instalační sada disket použitelná na počítačích s DOSem nebo OS/2.

Pro UNIX zatím podobná věc není udělána, protože disketové médium pro UNIX nebývá zcela obvyklé a instalační TARy základu daleko přesahují kapacitu jedné diskety.

*Petr Olšák
5. února 1996*

V tomto článku bude ukázáno, jak je v C_ST_EXu implementován L^AT_EX 2_ε. Zaměříme se zejména na problematiku národní podpory T_EX, a tudíž i L^AT_EX, je postaven tak, aby pracoval stejným způsobem na všech operačních systémech. Přesto se jednotlivé implementace pro různé operační systémy poněkud liší. Tento článek se tudíž více dotkne té implementace, kterou používá autor, a to je emT_EX pro OS/2.

Instalační kontroverze

Při instalaci se mírně protichůdně stýkají čtyři různé koncepce. První z nich je koncepce autorů L^AT_EXu, druhou koncepcí je přístup autora emT_EXu – Eberharda Mattesa, další proud tvoří názor Petra Olšáka jako hlavního tvůrce C_ST_EXu a závěrem jsou to rozhodnutí spoluautora národní podpory.

L^AT_EX 3 team poskytuje dokumentovaný zdrojový kód. Ten se rozbalí T_EXovým programem DOCSTRIP a výsledné soubory se mají přesunout „tam, kde je T_EX očekává“. Pak se spustí iniT_EX a formát je hotov. Rozbalovací fáze trvá na méně výkonných počítačích dlouho, proto se L^AT_EX dodává i v rozbalené podobě.

E. Mattes poskytuje původní i rozbalenou verzi. Dokumentovaný zdrojový kód je v adresáři `emtex\dist\latex\base\`, všechny soubory, které vyprodukoval DOCSTRIP, jsou v `emtex\texinput\latex2e\`. Tam také `tex*.exe` hledá vstupní soubory. Po vytvoření formátu však zůstává v pracovních adresářích řada souborů, které již nejsou potřebné a zbytečně zabírají místo těm, kteří mají na svém disku těsno. L^AT_EX ovšem používá řadu souborů interně. Běžný uživatel, který nechce číst dlouhou dokumentaci, tedy neví, co může beztržně smazat.

Petr Olšák správně tvrdí, že uživatel by měl vytvořit formáty, načež může smazat celý strom `emtex\dist\`, aby mu na disku nepřekážely soubory, které nepotřebuje. Vyžaduje ovšem, aby soubor `cslatex.ini` a další konfigurační soubory byly přítomny v adresáři

`emtex\texinput\latex2e\cslatex\`. Tam také zůstanou po vytvoření formátu a zvláště v DOSu na FAT blokují tyto kraťoučké soubory zbytečně mnoho místa. Z hlediska šetření místa by jejich umístění v `emtex\dist\cslat2e\` bylo logičtější, ale to by vyžadovalo nemalé zásahy do struktury $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u.

Současná instalace české a slovenské verze $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u musí splňovat řadu částečně protichůdných požadavků. Předně musí být slučitelná se zbytkem $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u. Musí proto vyhovovat přání Petra Olšáka. Musí být schopna koexistence s $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ em 2.09, takže je nutno dodržet adresářovou strukturu, kterou vytvořil Eberhard Mattes. Navíc se zde uplatňuje to, co v úvodním článku tohoto čísla napsal Petr Olšák – autoři vytvářejí tento produkt v jiném systému než je MS DOS. Původní verzi národní podpory napsal Jiří Zlatuška a podle jmen souborů je zřejmé, že byla vytvářena na UNIXu. Ode mě pocházejí úpravy, jejichž cílem je přiblížení celé implementace potřebám $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u. Přitom se vychází z mých vlastních potřeb.

Národní podpora je implementována pomocí konfiguračních souborů. Ty vytvořil Jiří Zlatuška podle svých požadavků. Já jsem je dále modifikoval, takže i tyto konfigurační soubory jsou dále konfigurovatelné. Pokud tyto konfigurační soubory „druhého řádu“ nejsou přítomny, vytvoří se stejný formát, jaký má Jiří Zlatuška. Tím se snad přispěje k lepší unifikaci.

Moje verze běhá pod OS/2, kde nejsou problémy s omezením paměti na 640 kB (velikost virtuální paměti je omezena pouze volným místem na pevném disku), přičemž požadují vzory dělení pro řadu neobvyklých jazyků, jako je dánština apod. Takový formát je pro běžné české a slovenské uživatele nepotřebný a svým rozsahem dokonce může až obtěžovat. Je tedy účelné vytvořit pro $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ formát jednodušší, který pro kompatibilitu se světem bude obsahovat vzory dělení pro americkou angličtinu a pro nás vzory dělení české a slovenské. To přináší rozdíl mezi instalací „oficiální“ a soukromou, což může být zdrojem chyb. Proto je při instalaci prohlížen další adresář, kam uživatel (hlavně já) může podstrčit svůj konfigurační soubor a vytvořit si tak jinou verzi s jinými vzory dělení. Pak ovšem nelze použít instalační dávku `inibat` z $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u, ale musí se použít postup, který bude uveden později. Takový konfigurační soubor musí být v adresáři `emtex\dist\cslatex.cfg\`. Výhoda umístění uživatelských konfiguračních souborů mimo standardní větvu spočívá v tom, že se při instalaci nové verze $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u nepřemažou. Tímto mechanismem

se v autorově dílně vytváří jak formát soukromý, tak formát oficiální, a díky vlastním dávkám v OS/2, které byly nyní v nouzi úspěšně převedeny i do DOSu, lze oba formáty provozovat současně. Přitom dlouhá jména souborů nejsou naštěstí problémem. Při rozzipování pod DOSem se překousnou podle konvence 8+3 a emTeX si umí zkrátit jména uvedená v makru `\input` tak, že se soubory správně najdou.

Proč používat L^AT_EX 2_ε?

Tato otázka sice patří úplně do jiného dokumentu, ale přesto ji zodpovíme zde. L^AT_EX 2_ε totiž při instalaci způsoboval jisté obtíže a mnohým nezasvěceným uživatelům by se mohlo zdát, že jejich překonávání za tu námahu nestojí a mohou si klidně vystačit se starou verzí 2.09. To ovšem platí pouze pro ty uživatele, kteří se spokojí s nenáročnými texty.

Základním problémem L^AT_EXu jsou fonty. V původní verzi s `old font selection scheme` jsou dosti pevně zadrátovány fonty `computer modern`. Jednotlivý font lze zavést makrem `\newfont` i primitivem `\font`. Pak ale nefungují přepínače `\bf`, `\small`, apod. Plnohodnotné zavedení nových fontů do L^AT_EXu 2.09 není triviální záležitostí. Tento problém, mimo jiné, řeší `new font selection scheme`. Nový font se zavádí pomocí jednoduchých deklarací. Ortogonální přepínače umožňují použití *tučně kurzívy* prostým uvedením `\it\bf`. Za tuto výhodu však platíme nekompatibilitou. Řada dokumentů, psaných pro NFSS, nelze s OFSS zpracovat, dokumenty psané pro OFSS poskytují nesprávný výstup v NFSS, zejména v matematickém režimu. A právě to řeší L^AT_EX 2_ε.

Všichni víme, že L^AT_EX 2_ε se trvale mění a dvakrát ročně je nová verze. Je to tím, že konečným cílem je L^AT_EX 3 a L^AT_EX 2_ε je vlastně testovací formát pro nové koncepce. Kromě kompatibility starých a nových dokumentů je zde pohodlně podporována vícejazyčná sazba, kdy lze současně používat fonty v různých kódováních. Dokonce i pro jeden jazyk lze použít v jednom dokumentu různé kódované fonty, přičemž dělení funguje ve všech kódováních správně. Logickou nevýhodou jsou značné nároky na paměť a výrazně pomalejší překlad. Uživatelé s předpotopným PC pak mohou neúspěšné pokusy o vytištění i nejjednodušších dokumentů odradit. Pro urychlení existují programy, s jejichž pomocí lze rychle přelatexovat označenou část zdrojového textu a zobrazit v prohlížeči. Na současných počítačích to však není nutné. Na 66 MHz 80486

v OS/2 pracuje `tex386.exe` tak rychle, že jsem při překladu celé knihy rád, když mi počítač dopřeje pár sekund odpočinku.

Z toho vyplývá následující doporučení: pokud chcete vážně pracovat s \TeX em, chcete využívat PostScriptové fonty, sázíte vícejazyčné dokumenty, máte fonty z různých zdrojů s různým kódováním a nechcete ztrácet čas vytvářením virtuálních skriptů pro překódování, zapomeňte na DOS a Windows, pořídte si pořádný operační systém (OS/2 nebo UNIX) a používejte $\LaTeX 2_{\epsilon}$. Pokud snad chcete zůstat u procesoru 80286 či horšího a přejdete na $\LaTeX 2_{\epsilon}$, nezbyde vám nic jiného než nečinně přihlížet, jak vaše stránky zvolna přežvykuje big \TeX .

Vstupní a výstupní kódování

\TeX byl v původní verzi sedmibitový a zpracovával jen první polovinu ASCII tabulky. Akcentované znaky se musely zadávat pomocí maker, což je pro české i slovenské texty velmi nepohodlné. Naštěstí Don Knuth vytvořil osmibitovou verzi, při jejímž použití lze psát texty v editoru s českou či slovenskou klávesnicí.

Původní cm-fonty však nemají akcentované znaky, obsahují pouze samostatná diakritická znaménka a česká i slovenská písmenka se z nich dají složit. To se ale dosahuje různými triky. Zdrojový text se může překonvertovat filtrem tak, aby obsahoval pouze sedmibitové znaky a makra a tento soubor se pak předloží \TeX u. Tím ale trpí kvalita dělení. Proto Oldřich Ulrych podstrčil \TeX u upravené metriky a programem `dvi2dvi` provedl příslušné změny až v DVI-souboru. Vše by vyřešily osmibitové fonty, ale původní dc-fonty nerespektovaly zcela česká typografická pravidla. Proto byly vytvořeny ζ -fonty.

Železná opona kdysi vytvořila kódovou džungli. Snad každý začal na nějakém PC používat češtinu Kamenických, která ale z technických důvodů nešla použít na nejstarších laserových tiskárnách. IBM, Hewlett-Packard a MicroSoft vytvořily společně kódovou stránku 852, kterou oficiálně používá DOS i OS/2. V RVHP byla normalizována KOI-8-ČS, která se ale téměř nepoužívá. Na UNIXových strojích zvítězilo kódování ISO 8859-2. Windows, aby to nebylo tak jednoduché, používají kódovou stránku 1250, která se bezdůvodně liší od ISO 8859-2 v pěti znacích. Windows '95 jsou samostatná kapitola a raději se o nich rozepisovat nebudu. $\zeta\TeX$ tuto džungli bohužel ještě prohloubil. Při tvorbě $\zeta\TeX$ u '94

se mylně předpokládalo, že v UNIXu musí být kódování fontů shodné se vstupním kódováním. \mathcal{C} -fonty mají proto kódování IL2, což znamená, že spodní polovina má stejné kódování jako původní cm-fonty, horní polovina je kódována podle ISO 8859-2. \mathcal{C} -fonty lze tedy bez obav použít místo cm-fontů, ale jinak nejsou kompatibilní s ničím. Pokud chce uživatel společně s \mathcal{C} -fonty použít dc-fonty nebo jiná písma, dostupná pravděpodobně v kódování T1 (kód dc-fontů), musí se s tím nějak vypořádat.

První řešení je čistě \TeX ovské. Lze vytvořit virtuální skript, jímž se font překóduje. Práci sice usnadní Mattesův `qdTeXvp1`, ale nějaký čas to stejně zabere. Navíc se to s každým novým fontem musí provádět znova.

Jinou možnost nabízí standardní \LaTeX 2_{ϵ} . Příkazy pro změnu fontů totiž zahrnují i přepínač pro změnu kódování. Pak lze akcentované znaky zadávat pomocí maker a během sazby je \LaTeX 2_{ϵ} převede na osmibitový kód aktuálního fontu. Dělení tedy funguje spolehlivě, ale pro tvorbu českých a slovenských dokumentů opět potřebujeme filtr. Chceme-li se filtru vyhnout, musí být akcentované znaky aktivní a musí se expandovat na příslušná makra, která pak \LaTeX 2_{ϵ} zkonvertuje. Toto řešení vede k téměř 100% přenositelnosti zdrojových textů, ovšem nefunguje správně `\uppercase`.

UNIXový \TeX na rozdíl od `emTeXu` neumí tcp-tabulky, takže následující vlastnost v něm nelze zařídit snadno (vyžaduje překlad s jiným změnovým souborem). V tcp-tabulce `emTeXu` lze totiž specifikovat, že akcentované znaky se mají při čtení nahradit \TeX ovskými makry. Pak lze míchat fonty s různým kódováním a funguje i `\uppercase`. Bez problému mohou napsat i `ñ` a `ç`, což umí moje ósedvojková klávesnice, a \LaTeX 2_{ϵ} vše vysází správně, přestože v \mathcal{C} -fontech tyto znaky nejsou. Každá výhoda však má i nevýhody. Pokud si zobrazíte český text makrem `\typeout`, uvidíte místo českých písmenek na obrazovce \TeX ovská makra. To lze ještě přelustit, daleko horší jsou hlášení o přetečených a podtečených boxech. Zde nám \TeX nabídne pouze hexadecimální výstup se zobáky, což se čte dosti těžko. To ale pořád není to nejhorší.

Představte si, že máte starý dokument, který jste kdysi vysadili v \LaTeX u 2.09 a nyní jej sázíte znovu. Použijete \LaTeX 2_{ϵ} , tcp-tabulka převede akcentované znaky na makra, spustíte prohlížeč a s hrůzou zjistíte, že prostředí `tabbing` se zcela rozpadlo. Tam totiž má `\` zcela jiný význam. Naštěstí se `tabbing` vždycky dá přepsat pomocí balíku `longtable`. Pro `verbatim` však taková jednoduchá pomoc neexistuje.

Podobný problém přinesou i aktivní akcentované znaky. Lze sice napsat chytrější makra, která by principiálně ošetřila `tabbing` i `verbatim`, ale nejsem si jist, zda to již současná verze dělá.

Jednoduché řešení tedy neexistuje. autor používá více různých formátů podle okamžitých potřeb. Podle počtu použitých vzorů dělení zabírá formát \LaTeX u 700–800kB, což si lze celkem dovolit i na malém disku.

Instalace

Základní instalace v $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u spočívá v tom, že se rozzipují příslušné soubory a spustí se dávka `inibat`. Ta vytvoří standardní formát `cslatex`. Ten obsahuje vzory dělení pro americkou angličtinu, češtinu v kódování IL2 i T1 a slovenštinu v kódování IL2. Více se do paměti malého $\mathcal{T}\mathcal{E}\mathcal{X}$ u nevejde. Standardní `tcp`-tabulka převádí vstup z kódu Kamenických do IL2. Formát lze vygenerovat i malým $\mathcal{T}\mathcal{E}\mathcal{X}$ em, ale řadu dokumentů, zejména v režimu kompatibility s $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 2.09, nelze malým $\mathcal{T}\mathcal{E}\mathcal{X}$ em zpracovat.

Pokud máte jiné požadavky na konverzi vstupních znaků nebo chcete instalovat vzory dělení pro jiné jazyky, musíte vygenerovat jiný formát. Je vhodné nazvat jej jinak, i když bude kompatibilní s formátem `cslatex` a běžné české texty se v obou formátech vysází stejně. Bohužel nevím, jak upravit $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$, aby vygeneroval jiný formát. Lze to však učinit použitím Rexxové dávky `cslatex.cmd` pro OS/2 nebo `cslatex.bat` pro DOS. Obě dávky jsou v adresáři `emtex\dist\cs1t2e\`. Jsou dosti složité a jejich přesný popis by zabral příliš mnoho místa. V následujícím textu bude pouze naznačeno, jak si lze $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 2 ϵ překonfigurovat. Podrobnější údaje jsou pro zájemce uvedeny v dokumentaci. V OS/2 si můžete přečíst hypertextový dokument `emtex\book\czech\cs1txins.inf`, pro ostatní je k dispozici „lineární“ návod `emtex\doc\czech\cs1txins.*`, kde přípona specifikuje kódování textu.

Instalační dávky plní řadu úkolů, v OS/2 poměrně efektivně, neboť Rexx je dosti silný programovací nástroj, v DOSu hloupě a vyžaduje velký environment. Dokud jsem ještě používal DOS, musel jsem nastavit prostor pro environment na 2048, aby mé dávky fungovaly. To je první věc, kterou musíte zajistit, když se rozhodnete pro použití instalační dávky `cslatex.bat`. V OS/2 tento problém odpadá.

Mé spojení na Internet je pomalé. Proto se mi někdy nevyplácí čekání, až se mi konečně natáhne rozbalená forma a raději si pustím DOCSTRIP sám. Instalační dávka tedy sama testuje, zda je L^AT_EX 2_ε již rozbalen. Podobně se zachází i s podporou češtiny a slovenštiny; zde je však důvod jiný. Drobná potíž je i s dc-fonty. Nový L^AT_EX 2_ε podporuje nové dc-fonty. Pokud má někdo staré dc-fonty, musí rozbalit další soubor. I to kontrolují mé instalační dávky a vše provedou automaticky.

Jak již bylo uvedeno dříve, L^AT_EX 2_ε je konfigurovatelný prostřednictvím konfiguračních souborů. Ty standardní najdete převážně v adresáři `emtex\texinput\latex2e\cslatex\`. Pokud je smažete, pokusí se in_iT_EX vytvořit stejný formát, jako má Jiří Zlatuška, a při tom T_EX v DOSu i v OS/2 zhavaruje. Pro svůj formát musíte tudíž napsat vlastní konfigurační soubory. Pokud přepíšete standardní konfigurační soubory, pak můžete o své změny při instalaci nové verze C_ST_EXu přijít. Proto instalační dávky prohlížejí přednostně adresář `emtex\dist\... \...cslatex.cfg\`, do něhož žádná distribuce nic nezapisuje. Standardní dávka `inibat` se ovšem do tohoto adresáře nekouká. Pokud si měníte konfiguraci, nezbyvá vám nic jiného, než pročíst instalační návod a použít mé instalační dávky.

Ještě je nutno zmínit se o vzorech dělení. C_ST_EX používá parametr `/mt25000`, aby i malý T_EX měl dost paměti pro ostatní objekty. Pokud budete přidávat další jazyky, musíte také něco vyhodit. Bez zvětšení hodnoty parametru `/mt` se obejdete jen stěží. Pak ovšem bude nutné zasáhnout do dávek C_ST_EXu, jinak si bude T_EX stěžovat, že nemůže za běhu zvětšit trie size.

Ještě jednou kódování

Současným standardem jsou cm-fonty s kódováním OT1. C_S-fonty jsou jejich rozšířením a jsou standardní součástí L^AT_EXu v C_ST_EXu. C_STUG bude, doufáme, i nadále podporovat C_S-fonty, zatímco svět přejde na dc-fonty s kódováním T1. Někteří uživatelé udělali tento krok již nyní a L^AT_EX 2_ε z C_ST_EXu jim to umožňuje. Stačí změna konfiguračního souboru. Na jednu stranu je to výhodné, ale přináší to problém. Snadno se totiž v makrech pro českou a slovenskou podporu něco přehlédne a L^AT_EX pak nefunguje tak, jak má. Programování také není jednoduché, když není předem dáno, jaké kódování bude použito standardně. Navíc se

L^AT_EX 2_ε neustále mění. Uživatel to sice nepozná, ale programátor vnitřních maker musí s každou novou verzí poměrně dost předělávat. Tím vznikají různé problémy a byl bych rád, kdyby postupem času ubývaly.

Některé soubory chybí

Podobně jako vlastní L^AT_EX 2_ε, dodává se řada dalších rozšiřujících nástrojů ve formě dokumentovaných zdrojových kódů, které si uživatel musí sám rozbalit programem DOCSTRIP. Tyto nástroje již nejsou tak objemné, takže se na CTAN rozbalené nenajdou. Snad proto je E. Matthes dodává v původním tvaru a najdou se v adresáři `entex\dist\... \latex\tools\`. Již několik uživatelů přišlo na to, že při pokusu o tisk dokumentace se nenajdou soubory `verbatim.sty` a `multicol.sty`. Uvažujeme o tom, že užitečné nástroje rozbalíme a připravíme do vhodných adresářů v C_ST_EXu. Také jsem byl upozorněn na to, že jsem do C_ST_EXu zapomněl vložit PSNFSS, takže PostScriptové fonty může nyní použít jen expert. Předpokládám, že v době, kdy čtete tento článek, jsou již uvedené problémy odstraněny.

Jak naložit s chybami?

Přes všechnu snahu chyby byly, jsou a budou. Pokud nějakou najdete, spojte se s autorem tohoto článku. Některé problémy lze dočasně napravit nějakou náplastí. Vždy však pomůžete sobě i ostatním, protože se můžete trefit do něčeho, co autor nepoužívá a chybu by tudíž nenašel. Další příčinou může být, že autor má nějaké soubory doma, všechno funguje, ale zapomněl je zabalit do distribuce. E-mailovou adresu najdete v podpisu i ve všech distribucích.

Zdeněk Wagner
wagner@mbbox.cesnet.cz

Program a2ac – manipulace s fontem na úrovni PostScriptu

PETR OLŠÁK

V tomto článku je popsán autorův program **a2ac**. Program umožňuje sázet texty PostScriptovým fontem v jazycích, ve kterých se používají akcentované znaky. Výchozí fonty nemusejí obsahovat celou abecedu daného jazyka; stačí, když jsou přítomny jednotlivé akcenty samostatně. Konfigurační soubory programu **a2ac** jsou nezávislé na kódování PostScriptového fontu i na kódování, které je použito v sázecím systému. Program může být použit na přípravu fontů pro libovolný sázecí systém, ovšem testován byl pouze **T_EX**. Použití **a2ac** pro **T_EX** vytváří alternativu k programu **fontinst** od Alana Jeffreyho. Balík **a2ac** je volně k dispozici na CTANu včetně zdrojového textu v jazyce C. Program byl autorem testován na systémech SUN OS, Linux a dále v DOSu¹.

Program **a2ac** (Afm To Afm add Composites) konvertuje **afm** soubory (Adobe Font Metrics) na nové soubory rovněž ve formátu **afm**. Při své práci načítá tzv. definiční soubor, v němž je stanoveno, jaké změny se mají v **afm** formátu provést. Především se přidávají nové kompozitní znaky a rozšiřuje se tabulka kernů o nové kerningové údaje podle masek.

V metrice **afm** jsou informace o znacích vyjádřeny pomocí symbolických názvů znaků (např. **Scaron** znamená Š, **Uring** je Ů a **eacute** je é). Každý použitý název může být podchycen v metrice dvěma způsoby. Buď je vázán na konkrétní pozici ve fontu (kód a PostScriptová procedura na vykreslení znaku), nebo má kód nastaven na -1 a je popsán v metrice jako složenina jiných znaků. Popis složených (kompozitních) znaků se přitom opírá o symbolické názvy, takže není závislý na kódování fontu.

Základní myšlenka programu **a2ac** je založena na tom, že v definičním souboru popíšeme všechny potřebné složeniny pomocí symbolických znaků a program **a2ac** je do metriky doplní. Protože program pracuje

¹V redakci byl úspěšně použit v OS/2 Warp, pro překlad kompilátorem IBM VisualAge C++ nebylo nutné provádět žádné změny ve zdrojovém kódu. Pozn. red.

na úrovni symbolických názvů znaků, jeho činnost je naprosto nezávislá na kódování jak PostScriptového fontu, tak vnitřního kódování sázecího systému, pro který se font připravuje.

Program `a2ac` je při čtení definičního souboru vybaven nenulovou inteligencí. Umožňuje deklarovat a používat proměnné, psát metrické a kompozitní informace pomocí jednoduchých výrazů a pomocí masky kernů doplňovat rozsáhlé množství nových kerningových informací pro nové znaky.

Následuje ukázka formátu definičního souboru. Příklad zahrnuje informace o nových písmenech `É`, `Ě`, `é` a `ě`. Nejprve jsou definovány znaky `Ecaron` a `Eacute` jako kompozitní (viz řádky `NC`), pak jsou vytvořeny pro tyto znaky nové kerningové informace. Kerny budou stejné, jako kerny s písmenem `E` (viz řádek `NK`). Dále jsou definovány znaky `ecaron` a `eacute` jako kompozitní a tyto znaky ponesou stejné kerningové informace, jako by se jednalo o písmeno `e` (viz řádky `NC` a `NK`). Nakonec jsou uvedeny některé výjimky. Dvojice `Pě`, `Tě` a `Vě` nebudeme kernovat, zatímco dvojice `Pe`, `Te` a `Ve` většinou mají záporný kern. Podobně pro dvojice `Pé` a `Té`.

```
NC Ecaron 2 ; PCC E 0 0 ; PAT caron 0 Carontop ;
NC Eacute 2 ; PCC E 0 0 ; PAT acute Acuteshift Acutetop ;
NK (Ecaron,Eacute) : E
NC ecaron 2 ; PCC e 0 0 ; PAC caron 0 0 ;
NC eacute 2 ; PCC e 0 0 ; PAC acute acuteshift vshift ;
NK (ecaron,eacute) : e
RK (P,T,V) ecaron 0
RK (P,T) eacute 0
```

Údaje o umístění akcentů jsou zde uvedeny pomocí proměnných (např. `acuteshift`), které jsou v definičním souboru deklarovány dříve. Program usazuje akcenty podle různých hledisek (viz `PCC`, `PAT` nebo `PAC`) navíc s přihlédnutím ke sklonu písma. Podrobnější informace o formátu definičního souboru najde čtenář v dokumentaci `a2ac-cz.doc`.

Do definičního souboru nelze zanášet informace o ligaturách. Ukážeme, že to nevádí. Pokud font ligatury má, pak ve vstupním `afm` souboru se na konci některých řádků s prefixem `C` vyskytuje údaj o ligatuře. Například:

```
C 102 ; WX 333 ; N f ; B 20 0 383 683 ; L i fi ; L l fl ;
```

Tato informace se přepíše do výstupního `afm` souboru a pro vytvoření záznamu v tabulce ligatur pro sázecí systémy je dostačující. V případě `TeXu` čte tyto údaje program `afm2tfm` a vytváří odpovídající záznam v ligační tabulce pro formát `tfm`. Navíc se dají prostřednictvím souborů typu `enc` definovat nové, speciálně `TeXovské`, ligatury. Například řádek v souboru `xl2.enc`

```
% LIGKERN hyphen hyphen =: endash ; endash hyphen =: emdash ;
```

definuje známé ligatury pro „--“ a „---“.

Příprava fontu pro `TeX`

Po transformaci `afm` metriky programem `a2ac` se může použít program `afm2tfm`, který čte upravený soubor `afm`. Kromě toho program čte soubor `enc`, podle něhož provádí překódování fontu do kódování, které je použito v `TeXu`. Výsledkem je virtuální font, který obsahuje nejen informace o překódování fontu podle souboru `enc`, ale obsahuje též mechanismy na sestavení všech kompozitních znaků. Tyto mechanismy byly přitom definovány pomocí popisu kompozitních znaků v souboru `afm`. O to se postaral program `a2ac` prostřednictvím svého definičního souboru.

Začněme nyní s přípravou fontu pro `TeX`, který podporuje českou a slovenskou sazbu. Nejprve spustíme program `a2ac`. Ten se volá se třemi parametry:

```
a2ac input.afm corr.tab output.afm
```

První parametr je název vstupního (`afm`) souboru, druhý parametr označuje definiční soubor obsahující definice změn, které se mají provést. Třetí parametr je název výstupního souboru. Přípony souborů (`.afm`, `.tab`) je nutno uvést, program si je nedomyslí.

Příkazový řádek volání programu bývá obvykle součástí nějakého scriptu nebo dávkového souboru. Například, pro přípravu fontu z `afm` formátu do `tfm` můžeme použít jednoduchý UNIXový script, který vypadá takto:

```
a2ac $1.afm cscorr.tab c$2.afm
afm2tfm c$2.afm -t xl2.enc -v c$2 r$2
vptovf c$2.vpl c$2.vf c$2.tfm
```

Pokud se script jmenuje třeba `mkfnt`, pak nový font připravíme například tímto příkazem:

```
mkfnt Times-Roman ptmr
```

K balíku programu je přidán definiční soubor `cscorr.tab`, který obsahuje definice korekcí, pomocí nichž lze vytvořit font obsahující znaky české a slovenské abecedy a vhodný kerning. Při použití tohoto souboru se pouze vyžaduje, aby vstupní `afm` soubor obsahoval všechny akcenty pro české a slovenské znaky. To splňují např. všechny standardní Adobe fonty. Výstupem je `afm`, obsahující navíc kompozitní znaky a rozšířený kerning.

Soubor byl beze změny použit na všechny základní PostScriptové fonty (32 fontů v každém RIPu) s velmi slušným výsledkem. Chceme-li docílit ještě lepšího výsledku, je možné pro různé fonty soubor `cscorr.tab` nepatrně obměňovat.

Po použití programu `a2ac` jsou další kroky pro přípravu fontu standardní. Například výše uvedený UNIXový script připravuje \TeX ovský font postavený na kódování \mathcal{C} -fontů. Výstupem scriptu je `c*.tfm` (vstup pro \TeX), a dále `r*.tfm` a `c*.vf` (vstup pro `dvips`). Jediné, co je třeba udělat, je umístit tyto soubory do odpovídajících adresářů, nebo upravit script tak, aby to dělal za nás. Dále je potřeba přidat jeden řádek do souboru `psfonts.map` (konfigurační soubor programu `dvips`). Řádek vypadá takto:

```
rfont PostScript-Font
```

Pokud není PostScriptový font instalován v RIPu výstupního zařízení, musíme jej uložit ve formátu `pfb` (`pfa`) do našeho počítače. Pak řádek v souboru `psfont.map` bude mít tvar:

```
rfont PostScript-Font </cesta/k/souboru.pfb
```

nebo (pro DOS):

```
rfont PostScript-Font <d:\cesta\k\souboru.pfb
```

Nyní můžeme zavést font do \TeX u pomocí těchto příkazů:

```
\font\newfont=cfont
```

```
{\newfont zde je použit osmibitový text
```

```
ve stejném kódování, jako je definováno v~souboru  
xl2.enc. Tento text bude vysázen v~novém fontu.}
```

Po \TeX ování a zpracování programem `dvips` dostaneme text vysázený novým fontem.

Kódování \TeX u

Není nutné, aby se kódování vstupního souboru shodovalo s kódováním \TeX u. Je možné provést transformace na úrovni `tcp` tabulek v `em \TeX` u nebo změnami vektorů `xord/xchr` v souboru `tex.ch` (změny zdrojového textu \TeX u ve formátu `web`) nebo nastavením aktivních znaků. Soubor `enc` určuje kódování virtuálního fontu, které se musí shodovat s kódováním tabulek dělení slov. Tomuto kódování říkáme vnitřní kódování \TeX u.

V našich zemích se asi nejčastěji používá vnitřní kódování \TeX u podle \mathcal{C} -fontů (tzv. \mathcal{C} -kódování). Tyto fonty jsou přirozenou součástí pravděpodobně nejrozšířenější instalace \TeX u u nás s názvem $\mathcal{C}\TeX$. \mathcal{C} -fonty v tomto balíčku zcela nahrazují textové Computer Modern fonty. \mathcal{C} -kódování je nadmnožinou kódování textových fontů Computer Modern, přičemž rozšíření zahrnuje akcentované znaky české a slovenské abecedy podle ISO 8859-2.

Výhody tohoto kódování (ve srovnání například s mezinárodně používaným kódováním T1) jsou tyto:

- Fonty obsahují jen ty znaky, které jsou nutně potřeba pro českou a slovenskou sazbu. Příprava sazby tedy není zatížena zbytečnostími, což šetří čas i místo na disku. Například při generování z `METAFONT`ových zdrojů se pracuje jen se znaky, které se vyskytují v české nebo slovenské sazbě a soubory `pk` jsou menší. Výjimky (třeba znaky v cizích jménech) lze vždy obejít použitím makra z `plain \TeX` u nebo primitivem `\accent`.
- Protože je toto kódování nadmnožinou kódování textových fontů Computer Modern, je možno v instalaci zcela nahradit Computer Modern fonty \mathcal{C} -fonty bez dalších zásahů do \TeX ovských maker. \TeX se pouze prostřednictvím `\catcode` seznámí s novými písmeny (kódy větší než 128) a pomocí `\lccode`, `\uccode` se naučí dvojice malé/velké písmeno. V instalaci $\mathcal{C}\TeX$ je například formát `csplain`, který je zcela shodný s formátem `plain` s jedinou výjimkou. Textové fonty CM jsou nahrazeny \mathcal{C} a formát je přizpůsoben pro osmibitový vstup v kódování ISO 8859-2.

- Protože jsou akcentované znaky umístěny podle ISO 8859-2, je možno v UNIXových instalacích rovnou použít osmibitový český a slovenský text bez vstupního překódování. V jednoduchosti je síla! Na druhé straně v DOSu není pro naše jazyky ustálen jediný standard kódování textových souborů. Proto je zde potřeba použít tabulku `tcp`.
- Kerning a další náležitosti fontu můžeme připravit s ohledem na českou a slovenskou sazbu a nemusíme čekat, až nám to někdo daleko za našimi hranicemi udělá. Principiálně ani není možné, aby jediný font vyhovoval tradicím většího množství národů současně.

V balíku programu `a2ac` jsou zařazeny soubory `x12.enc` a `xt2.enc`. Tyto soubory definují kódování, které je nadmnožinou \mathcal{C} -kódování. Rozšíření zahrnuje některé znaky, které nejsou obsaženy v české a slovenské abecedě (tj. v \mathcal{C} -fontu), nicméně jsou zahrnuty v materiálu PostScriptových fontů kódovaných podle Adobe StandardEncoding.

Protože textové Computer Modern fonty nejsou jednoznačně kódované, je třeba řešit několik drobných problémů. Pro některé pozice existují dvě alternativy: rozlišují se fonty typu `\rm` a `\tt` (např. ligatura `fi`, nebo šipka dolů) a fonty typu `\rm` a `\it` (dolar nebo libra na stejné pozici).

Je zřejmé, že konzervativní rozšíření CM fontů (například \mathcal{C} -fonty) také nemůže být jednoznačné. Máme-li vnitřní kódování \TeX u postavené na bázi \mathcal{C} -fontů, je třeba použít dva kódovací soubory typu `enc`: soubor `x12.enc` pro fonty typu `\rm`, `\it` a soubor `xt2.enc` pro fonty typu `\tt`. Dolar je ponechán na pozici 36 ve všech fontech a libra je umístěna na pozici 132. Po zavedení PostScriptového fontu typu `\it` je tedy potřeba vědět, že libra není dosažitelná standardním plain \TeX ovským špinavým trikem: `{\it\$}`, ale je potřeba použít makro, které volá `\char132`.

Pozice 32 není v souboru `x12.enc` definována, protože škrťátko pro polské `L` a `l` není v Adobe StandardEncoding zahrnuto. Místo toho jsou na pozicích 163 a 179 samostatné znaky `L` a `l`. Pokud tyto znaky chceme používat v plain \TeX u s PostScriptovým fontem, je vhodné předefinovat makra `\l` a `\L`.

Všimneme si dále, že v Adobe StandardEncoding tabulce chybí trojnásobné ligatury a velká řecká písmena. Nepřítomnost trojnásobných ligatur v PostScriptovém fontu nám příliš nevádí, protože tyto ligatury se v českém a slovenském jazyce nevyskytují. Pokud navíc máme Post-

Scriptový font v kódování ExpertEncoding, můžeme trojnásobné ligatury přidat na úrovni `vpl` souborů.

Nepřítomnost řeckých znaků je jedna z příčin (bohužel ne jediná), proč nelze snadno použít nový PostScriptový font v matematické sazbě. Velké množství plain \TeX ovských maker a vestavěných algoritmů \TeX u se opírá o jisté speciální vlastnosti matematických fontů. Mám na mysli třeba rozsáhlé množství parametrů `\fontdimen` pro matematický font nebo různé plain \TeX ovské triky skládání dlouhých šipek pomocí segmentů složených ze znaku „mínus“ nebo „rovná se“. Aby vše vyšlo i pro PostScriptový font, je potřeba nejen zařadit na úrovni `vpl` do fontu písma řecké abecedy (obvykle z fontu Symbol), ale je potřeba též obohatit font o plno parametrů `\fontdimen` a vše sladit s \TeX ovskými makry. To bohužel není jednoduchá záležitost a už vůbec se to nedá automatizovat. Odladit matematickou sazbu v novém fontu vyžaduje cit výtvarníka a hluboké zkušenosti s \TeX em. Proto se často od sazby matematiky v PostScriptovém fontu upouští² a nové fonty se zavádějí jen do textové části sazby. Pokud pro zavedení fontu použijeme pouze primitiv `\font` a nebudeme se pouštět do zavádění nových matematických skupin pomocí `\textfont`, `\scriptfont` a `\scriptscriptfont`, zůstane při použití plain \TeX u matematika ve fontu Computer Modern.

Náhradní fonty

Pro prohlížeč a tiskárny nevybavené PostScriptem je většinou potřeba vytvořit náhradní font. Buď se spokojíme s přibližnou náhradou pomocí standardních CM (CS) fontů nebo použijeme program `ps2pk`. Druhá cesta je možná jen za předpokladu, že font máme k dispozici ve formátu `pfb` nebo `pfa` v počítači. To nemusí být pravidlem – často je font instalován jen v RIPu výstupního zařízení a nejsme schopni jej odtud „vydolat“. Proto popíšeme cestu, jak je možné pracovat s náhradními fonty.

Náhrada běžnými \mathcal{G} -fonty se provede pomocí následujících kroků. Například pro nahrazení fontu `cptmr` použijeme font `csr10`.

```
tftopl csr10.tfm csr10.pl
```

provede konverzi metriky do formátu `pl`.

²Existují PostScriptové fonty, které jsou vybaveny podporou sazby \TeX ovské matematiky, ale je jich málo.

Dále je třeba do `csr10.pl` doplnit příkazy `MAPFONT` a `VTITLE` (viz [1]). Na začátek souboru napíšeme

```
(VTITLE Substitute font cptmr -> csr10)
(MAPFONT D 0
  (FONTNAME csr10)
  (FONTCHECKSUM D 0)
  (FONTAT R 1.0)
)
```

Nyní přejmenujeme pozměněný spouboř `pl` na `vpl`:

```
mv csr10.pl ptrm.vpl
```

(v DOSu: `ren`), a konečně vytvoříme „nahrazovací“ virtuální popis:

```
vptovf ptrm.vpl ptrm.vf ptrm.tfm
```

Soubor `tfm` zahodíme a soubor `vf` předložíme ovladači, který má použít náhradní font. Pozor! Nikoli pro finální výstup, tedy nikoli pro ovladač `dvips`.

O názvech akcentovaných znaků

Pro znaky `đ`, `ť` by se správně měly použít symbolické názvy `dquoteright` a `tquoteright`. Místo nich byly ale použity symbolické názvy `dcaron` a `tcaron`. Důvod je ten, že tyto znaky mají svůj protějšek ve verzálkách nazvaný správně `Dcaron` a `Tcaron`. Kdybychom nazvali písmeno `đ` jako `dquoteright`, pak by použití parametru `-V` v programu `afm2tfm` pro vytvoření fontu „Capitals and Small Capitals“ nedalo správný výsledek. Abychom nezaváděli nová symbolická označení pro `Ĺ` a `ĺ`, zůstali jsme i zde u názvu `Lcaron` a `lcaron`.

Reference

- [1] Donald Knuth: Virtual fonts, a more fun for grand wizards. TUGboat 11(1):13–23, April 1990.
- [2] Petr Olšák: Typografický systém \TeX . CSTUG 1995, 270 stran. ISBN 80-901950-0-8.
- [3] Petr Olšák: Úvaha o fontech v \LaTeX u. \TeX bulletin 3/93 (121–131).

Následující dva texty vznikly jako semestrální práce z předmětu „Publikační systém T_EX“, který přednáší Petr Olšák na Elektrotechnické fakultě ČVUT. Po drobných, zejména jazykových, úpravách jsou zařazeny do našeho bulletinu, přičemž studenti (autoři jednotlivých příspěvků) s touto formou zveřejnění souhlasí.

České třídění

Exkurze do systémů zpracování textu

MARTIN PAVLÍK

Evoluce pokračuje – rodí se počítač

Člověk je tvor společenský a zřejmě z nedostatku osobního kontaktu s ostatními lidmi si vymyslel počítač, aby se necítil tak osamělý. Mnoho programátorů po celém světě tráví dlouhé noci před obrazovkami svých nových přátel a ve snaze vdechnout trochu života elektronickému miláčkoví ho zahrnují kilobajty, v posledních letech spíše megabajty naklofaného textu¹ očekávají kloudný výsledek. V zápalu nadšení však opominají být lidmi a sami se pomalu stávají tvory digitálními bez velké dávky estetického cítění. Toto se projevuje zejména v uživatelské nepřátelskosti některých programů, které jsou svými stvořiteli vybaveny mnoha neužitečnými funkcemi, a postrádají to, co je opravdu potřeba.

Jazykový babylon

Abychom si vzájemně *lépe* rozuměli, mluvíme na Zemi mnoha různými jazyky. Co národ, to jiná řeč a odlišné zvyklosti. V posledním desetiletí se stále více množí nový digitální tvor a s ním se rodí i nová kultura a nový jazyk. Počítače, bohužel, nejsou tak flexibilní jako lidé a nejsou schopny se přizpůsobit řeči lidské, proto se člověk pozvolna přizpůsobuje počítačům. V našich zeměpisných šířkách vzniká nový jazyk, velmi podobný

¹Slovo *text* se nečte *techt!*

češtině, avšak s mnoha odlišnostmi. Toto *nářečí* se jmenuje „cestina“ a některá jeho pravidla přiblíží následující pojednání. Jak dále uvidíte, „cestina“ není jen jedna, ale liší se počítač od počítače a program od programu.

Znakové sady

Kolébku počítačů jsou Spojené Státy a protože v této části světa řeč a písmo neprodělaly změny způsobené prací našich jazykotvůrců, jsou i počítače ve své původní podstatě ušetřeny všelijakých dlouhých a krátkých nabodeniček. To je jistě pro polovinu světa velmi příznivé, avšak při expanzi výpočetní $\text{T}_{\text{E}}\text{X}$ niky do naší malé země to s sebou přináší jisté komplikace co se týče znakových sad a ostatních pravopisných odlišností. Na neštěstí problému české znakové sady se ujalo hned několik pohotových tvůrců, každý si šel svou vlastní cestou (zaručeně tou nejlepší) a jako výsledek zde máme několik vzájemně neslučitelných výsledků. V tomto zmatku není mnohdy snadné přesvědčit vlastní tiskárnu aby tiskla česky a ne třeba hebrejsky. Osobně se přiznám, že na počítači píši raději vše anglicky, než abych se potýkal cestinou.

Třídění dle abecedy

Bohužel, znakovou sadou problémy s implementací češtiny zdaleka nekončí. V textových editorech, tabulkových procesorech a jiných produktech je jako jedna z nepostradatelných funkcí třídění dle abecedy a vytváření různých rejstříků a seznamů. Znaky s akcenty jsou umístěny v horní polovině znakové sady a pořadí znaků v takovéto sadě tedy neodpovídá abecednímu řazení. Třídí-li program jednotlivá hesla podle pořadí znaku ve znakové sadě, dopouští se tak chyb v řazení, které jsou nepřipustné.

České abecední řazení je specifikováno normou ČSN 01 0181, kde kromě pořadí jednotlivých znaků v abecedě jsou zakotvena také ostatní pravidla, týkající se pořadí velkých a malých písmen, číslic, ostatních znaků, a pravidla pro řazení víceslovných hesel. V této normě je také několik nejasností nebo sporných bodů, které nabízejí více vzájemně odlišných interpretací. Jako nejasné se jeví například třídění téměř shodných slov, lišících se pouze ve znacích s akcenty, nebo shodných slov odlišných pouze v počtu nebo umístění velkých a malých písmen. Norma zde

rozdílí je pořadí na základě počtu velkých nebo akcentovaných písmen, případně podle jejich umístění ve slově. Uvedená pravidla jsou nejasná a často i protichůdná.

Podle mého názoru je vhodné tuto normu přepracovat a použít trochu více zdravého rozumu. Pro snazší orientaci a vyhledávání je vhodné třídit jinak shodná slova postupným porovnáváním znak po znaku zleva doprava, kdy velké písmeno stojí v pořadí za písmenem malým a znak s akcentem za znakem bez akcentu.

Test vybraných programů

Třídící algoritmy v textových editorech a tabulkových procesorech se liší program od programu a jsou více či méně špatné či dobré. Namátkou jsem vybral několik programových produktů, jejichž třídící algoritmy jsem otestoval a dosažené výsledky předkládám v následujících řádcích.

Byly testovány programy Microsoft Word 5.0 a Claris Works 2.0 počítači Apple, dále český Word Perfect 5.1 a příkaz Sort operačního systému MS-DOS 6.0. U testu jednotlivých programů jsou v případě (vcelku) správného řazení uváděny pouze odlišnosti.

Pravidla správného řazení

abc nástrojaře	≺ ABC nástrojaře	(protože abc ≺ ABC)
abc	≺ ABC	(protože malá písmena ≺ velká)
ABC	≺ abc frézaře	(protože prázdné slovo je dříve než slovo frézaře)
abc frézaře	≺ abc nástrojaře	(podle druhého slova)
abc nástrojaře	≺ ABC nástrojaře	(protože abc ≺ ABC)
lalálá	≺ lálalá	(protože la ≺ lá)
nadivá	≺ nádiva	(protože na ≺ ná)
plagiát	≺ plachta	(protože g ≺ ch)
pláně	≺ plánička	(protože é ≺ i)
plánička	≺ Plánička	(protože p ≺ P)
pláňka	≺ plankton	(protože a ≺ t)
plášť	≺ plat	(protože š ≺ t)

plat	← plát	(protože a < á)
platno	← plátno	(protože a < á)
plátno	← platnost	(protože platno < platnost)
sténá	← stěna	(protože é < ě)

Vzor správného řazení

abc
 ABC
 abc frézaře
 abc nástrojaře
 ABC nástrojaře
 lalálá
 lálalá
 nadívá
 nádiva
 plagiát
 plachta
 pláně
 plánička
 Plánička
 plaňka
 plankton
 plášť
 plat
 plát
 platno
 plátno
 platnost
 sténá
 stěna

abc nástrojaře
 ABC
 ABC nástrojaře

*Řazení programem
 Claris Works 2.0*

abc
 ABC
 abc frézaře
 abc nástrojaře
 ABC nástrojaře

*Řazení programem
 Word Perfect 5.1 (česká verze)*

abc nástrojaře
 abc frézaře
 abc
 ABC nástrojaře
 ABC

*Řazení programem
 Word Perfect 6.0 (anglická verze)*

*Řazení programem
 Microsoft Word 5.0*

abc
 abc frézaře

abc nástrojaře
 abc
 abc frézaře
 ABC

ABC nástrojaře
plaňka
plachta
plášť
plagiát
plánička
Plánička
plankton
pláně
plat

Řazení programem
Sort DOS 6.0 (COUNTRY=042)

ABC
abc

abc frézaře
abc nástrojaře
ABC nástrojaře
plaňka
plachta
plagiát
plankton
plat
platno
platnost
plášť
plánička
Plánička
pláně
plát
plátno

Závěry testu

Program CSR [1] řadí podle uvedeného vzoru. Textový editor Microsoft Word 5.0 uspěl vcelku dobře, jen při rozlišování pořadí velkých a malých písmen ve víceslovných heslech měl problémy. Shodná hesla jsou tříděna nejprve podle prvního slova, kde malá písmena mají správně přednost před velkými. Tabulkový procesor Claris Works seřadil všechna hesla bez jediné chyby, i víceslovná hesla jsou seřazena správně nejprve podle abecedy a až poté podle velkých a malých písmen. Textový editor Word Perfect 5.1 pod DOSem řadil hesla správně, jen víceslovná hesla jsou seřazena poněkud podivně. Zdá se, jakoby algoritmus řadil podle druhého slova v heslu spíše sestupně než vzestupně. Řazení probíhá nejprve podle velkých písmen v prvním hesle, až poté podle druhého hesla. Textový editor Word Perfect 6.0 v anglické verzi není upraven pro třídění českých textů, výsledek tomu odpovídá. Pouze se zdá nepochopitelné, jak se slovo plaňka dostalo před slovo plachta. Je také patrné, že třídění probíhá pouze podle prvního slova v heslu. Příkaz Sort operačního systému DOS je velmi neschopný. Nedokáže rozeznat velká a malá písmena a jeho práce s akcenty je též na nic. Možná by se choval lépe, kdyby bylo použito kódování znaků podle MicroSoftu namísto

Kamenických. Toto kódování však pro jeho nedostatky téměř nikdo nepoužívá².

Názor autora

Drobné nedostatky v řazení, jako třeba prioritní řazení podle velkých písmen prvního slova před druhým slovem hesla, se dají v některých případech opomenout nebo obejít (lze si na ně zvyknout). Nesprávné řazení akcentů je však nepříjemné a proto nepřijatelné.

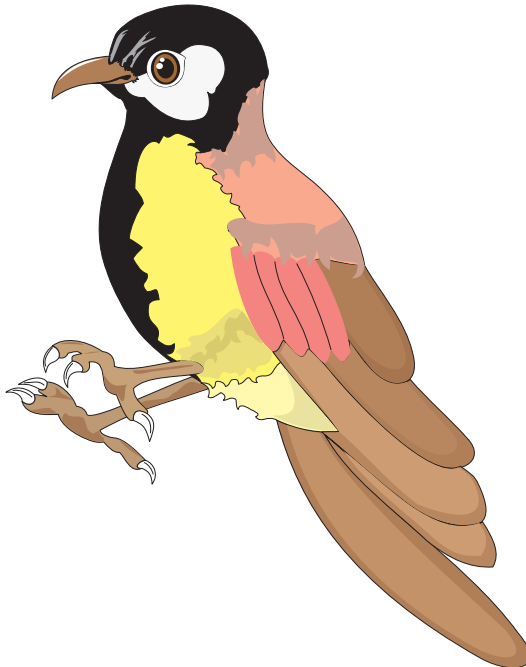
Je třeba si však uvědomit, že vlastní algoritmus řazení není vše, pokud nejsou uživateli poskytnuty ostatní potřebné funkce jako vytváření indexů, rejstříků, a jejich vzájemná provázanost. Program CSR řadí správně, ale je neohrabaný a chybí jeho provázanost s ostatními funkcemi. Velmi nepříjemné je omezení řazení na takzvané volné řádky vstupního souboru.³ Bylo by přinejmenším vhodné implementovat klíčová slova pro zahájení a ukončení bloku textu, ve kterém je třídění požadováno, např. `\csrbegin` a `\csrend`. Parametr třídění, tj. rozsah sloupců podle kterých se třídí, by měl také být redefinovatelný např. příkazem `\csrpara.....`

Literatura

[1] Petr Olšák, *Program csr (Czech Sort) – abecední řazení podle normy*. Zpravodaj ČSTUGu, 3:126–139, 1994.

²Kódová stránka 852 podle MicroSoftu a IBM je jediným standardem pro češtinu v OS/2, žádný jiný kód se zde nepoužívá. Pozn. red.

³Program CSR rozlišuje pevné a volné řádky vstupního souboru. Pevné řádky jsou všechny komentářové řádky od začátku souboru, dále následuje pole volných řádků a první následující komentářový řádek pole volných řádků opět uzavře.



1. Úvod


Na začátku mé práce bylo mnoho otázek, které se týkaly hlavně systému, který zvolím jako základ pro svoje seznámení s T_EXem. Po zralé úvaze jsem se rozhodl pro DOS. Bylo to rozhodnutí, které se samo nabízelo, a to hlavně proto, že SW pro DOS byl v tu chvíli pro mne dostupnější a zároveň jsem předpokládal, že s ním budou menší problémy při instalaci.

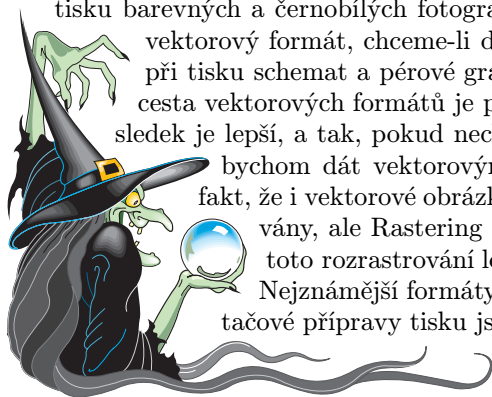
Jak hluboce jsem se mýlil, jsem poznal velmi záhy. $\text{T}_{\text{E}}\text{X}$ pro DOS je dostupný a snadno instalovatelný, ale to je vše, co se dá v dobrém o této kombinaci prohlásit. Velmi záhy jsem zjistil, že kombinace DOS a $\text{T}_{\text{E}}\text{X}$ není to „pravé ořechové“. Pracovat se s takovou kombinací sice dá, ale rozhodně ne pohodlně. Rozhodl jsem se vyzkoušet **UNIX**.



Vzhledem k tomu, že na počítači mi vedle plazícího se DOSu běhá ještě *Linux*, nainstaloval jsem $\text{T}_{\text{E}}\text{X}$ pro *Linux*.



2. Formáty obrázků


ormáty obrázků lze rozdělit na dvě základní skupiny. Do jedné patří formáty vektorové popisující obrázky matematicky a do druhé formáty orientovné bodově. Každý z formátů se liší oblastmi svého použití. Zatímco rastrové formáty jsou nepostradatelné při tisku barevných a černobílých fotografií, je nanejvýš vhodné použít vektorový formát, chceme-li dosáhnout maximálního účinku při tisku schémat a pérové grafiky. Všeobecně lze ale říci, že cesta vektorových formátů je průchodnější. Samozřejmě i výsledek je lepší, a tak, pokud nechceme tisknout fotografii, měli bychom dát vektorovým formátům přednost. Je sice fakt, že i vektorové obrázky musí být před tiskem rastrovány, ale Rastering Image Processor (RIP) zvládne toto rozrastrování lépe a v požadovaném rozlišení. Nejznámější formáty a jejich použití v oblasti počítačové přípravy tisku jsou v tabulce 1.



Formát	Použití
 Rastrové formáty	
PCX, BMP, GIF	Na počítačích třídy PC velmi oblíbené rastrové formáty, které však nejsou příliš vhodné pro barevnou grafiku.
TIFF, TGA	Velmi komplexní formáty pro práci s barevnou rastrovou grafikou.
XPM, XBM	Formáty rastrových obrázků používané v systému X Windows. Jsou vhodné zejména pro B&W grafiku.
 Vektorové formáty	
EPS, PS, AI	Tyto formáty jsou vlastně jen různá zapouzdření a slovníky formátu PostScript firmy Adobe. Jsou ve spojení s DVIPS maximálně vhodné pro vkládání obrázků do \TeX u.
WMF, CGM, DXF	Standardy z oblasti počítačů PC. Pro vkládání do \TeX u nejsou moc vhodné.

Tabulka 1: Grafické formáty

3. em \TeX


 kládání obrázků do \TeX u je záležitost poněkud delikátní povahy, a to už z toho důvodu, že se samotný \TeX o obrázky nijak nestará¹. Nechává jejich obsluhu na programech pro zpracování DVI souborů. Tyto programy využijí informace z DVI souboru a příslušný grafický soubor adekvátně zpracují. A tady už záleží hodně na vlastní distribuci \TeX u. Existují komplexní distribuce, které mají podporu obrázků již v sobě zahrnutou, ale existují i takové, které obrázky nijak nepodporují. Mezi distribuce, které obrázky podporují, patří třeba em \TeX .

Varianta \TeX u – em \TeX – je pro vkládání obrázků poměrně dobře vybavena. Umožňuje vkládat jak rastrové obrázky, tak i obrázky vektorové.

¹Samozřejmě kromě obrázků vytvářených v prostředí *PICTURE*


V em \TeX u se jedná o rozšíření programů pro zpracování DVI souborů. Zároveň musí být rozšířen i program `dvips`, který převádí DVI soubor do PostScriptu.

3.1. Vkládání bitmap

 vkládání bitmap se provádí pomocí speciálního příkazu pro DVI ovladač. Příkaz se vkládá většinou přes prostředí PICTURE \LaTeX u. Jeho tvar je `\special{em:graph jméno}`, kde jméno je jméno rastrového obrázku. em \TeX umožňuje vkládání nejběžnějších bitmap jako jsou BMP a PCX. Pokud vkládáme obrázky přes prostředí PICTURE, musíme znát jejich přesnou velikost v bodech, jinak se nám bude obrázek překrývat s textem². Zároveň si musíme uvědomit, že obrázky nebudou nijak zpracovávány, a proto je nutné vkládat je již jako bitmapy, které mají jen dvě úrovně jasu³. Proto je nutné vkládat obrázek přesně v tom rozlišení, v jakém ho chceme tisknout. To je poněkud problém při tisku na více zařízeních a při prohlížení, protože musíme měnit obrázek podle zařízení.

Obecně lze tedy prohlásit, že rastrové obrázky sice začlenit jdou, ale že je s nimi mnoho starostí, mají-li vypadat skutečně pěkně.

3.2. Vkládání vektorových obrázků

 ektorové obrázky jsou v \TeX u vlastně dvojího druhu. Za prvé jsou to obrázky vytvořené PostScriptem a za druhé obrázky tvořené v prostředí PICTURE (ev. v \TeX Cadu)⁴.

\TeX Cadem lze interaktivně kreslit jednoduché vektorové obrázky, které se převedou do posloupnosti příkazů prostředí PICTURE. Použijeme-li styly `emlines` a `bezier`, můžeme kreslit linky libovolného sklonu a Bézierovy křivky.

²Mezeru můžeme vložit i jiným vhodným příkazem.

³DVI-ovladač nedokáže rozptylovat.

⁴O používání Postscriptových souborů se zatím nebudu zmiňovat, protože o tom pojednává samostatná kapitola, která nemá s em \TeX em nic společného.

4. Another Fine T_EX



nebo další rozšíření T_EXu. Existuje jich několik a liší se možnostmi využití. Většinou se zaměřují na vytváření obrázků. Mezi taková rozšíření patří makro EPIC. Toto makro rozšiřuje možnosti prostředí PICTURE o další prvky a prostředí:

- `\multiputlist`, `\matrixput`, `\grid`, `\picsquare`, `\dottedline`,
`\dashline`, `\drawline`, `\jput`, `\putfile`,
- `dottedjoin`, `dashjoin`, `drawjoin`

Tyto funkce jsou přirozeným rozšířením prostředí PICTURE a jejich realizace se opírá přímo o funkce v T_EXu již implementované. Kromě něj však existuje ještě celá řada dalších rozšíření jako například TPIC, což je vlastně již programové rozšíření. Mezi funkce, které tato rozšíření nabízejí, patří kreslení grafů, tištění funkcí, napojení na další kreslicí programy, stínování, přidávání textur a celá řada dalších funkcí.

Tato rozšíření fungují buď samostatně jako EPIC nebo za podpory dalších programů jako v emT_EXu. Zpravidla také mění a rozšiřují již implementované příkazy prostředí PICTURE.

5. PostScript




d té doby, co firma Adobe přišla se svým jazykem pro popis stránky, uběhla už nějaká doba a z PostScriptu se stal standard pro tiskárny a někdy i pro obrazovky. Jeho použití je jednoduché, je poměrně snadno modifikovatelný a snad každý trochu slušný program pro práci s textem či obrazem ho umí zpracovat. Můj osobní názor je, že i do T_EXu je vkládání obrázků s použitím PostScriptu velmi efektivní.

Pokud použijeme makro EPSF, můžeme pomocí příkazu `\epsffile` vložit postscriptový soubor, který zachovává konvenci Encapsulated PostScript File. Kromě toho lze tento soubor zmenšovat a deformovat podle potřeby. Je to snad jediný rozumný způsob, jak s co nejmenším úsilím vložit skutečně vektorový obrázek. Zároveň PS dovoluje zapouzdřit bitmapové obrázky. To má navíc tu výhodu, že takto zapouzdřené bitmapy lze zmenšovat a deformovat. Navíc se nemusíme starat o rozlišení výsledného výstupního zařízení. Musíme pouze zajistit, aby hodnota DPI obrázku byla alespoň dvakrát tak velká, jako je nejvyšší hodnota LPI v reprodukčním řetězci.

Mírnou nevýhodou je, že ne každý ovladač DVI souborů je schopen tisknout PS informaci⁵. Nicméně toto lze obejít pomocí DVIPS, které by u kvalitního tisku mělo tak jako tak po T_EXu následovat.

Pro masochisty je zde navíc možnost vkládat do T_EXu přímo části programů PostScriptu. Zároveň je zde možnost použít PS soubor pro přidání barvy do T_EXu. Barva se ovšem dá aplikovat jen na text a nikoliv na obrázky. Nenalezl jsem makro pro barevné obrázky, ale domnívám se, že by bylo možné uzpůsobit EPSF pro vkládání barevných obrázků a jejich tištění T_EXem⁶.

6. Fine Art of T_EX Pictures

 azval bych tak práci s makrem FLOATFIG. Podle mého názoru je toto makro pro každého, kdo chce používat T_EX pro publikování, prostě nepostradatelné. Vkládní obrázků je zvláště v oboru počítačové grafiky velmi potřebná a frekventovaná činnost. Makro FLOATFIG umožňuje obrázky obtékat a zároveň je také samo umísťuje na obrázovky. FLOATFIG se rozhodne, na jakou polovinu stránky obrázky přijdou. To se liší podle sudých a lichých stran.

Prostředí `floatingfigures` dovoluje stejná rozšíření jako prostředí `figure`. To znamená, že obrázky jsou automaticky číslovány a uváděny v seznamu obrázků a že mohou mít popis. Jediným parametrem je šířka zabraná obrázkem. Rozdělení obrázků po stránce volí makro samo podle několika pravidel, ale zpravidla není třeba se o rozložení obrázků příliš starat. Já jsem si makro trochu upravil, abych měl větší kontrolu nad tím, jak jsou obrázky umísťovány na sudých a lichých stranách a jaké je jejich zarovnání vůči textu.

Pomocí makra `floatfig` jsem vytvořil i iniciály na začátcích kapitol. Právě kvůli nim jsem musel makro upravit, i když tato úprava se mi osvědčila i při normální práci s T_EXem.

⁵Což zřejmě neplatí o XDVI.

⁶Práci s PostScriptem v T_EXu velmi usnadňuje například balík PStricks Timothyho Van Zandta. Barevné obrázky lze vkládat i makrem EPSF. Při výstupu na ofset se musí provést barevné separace. Lze to provést na úrovni PostScriptových maker přímo v programu DVIPS, ale existují i osvitové jednotky, které barevnou separaci provádějí. Pozn. red.

7. Závěr

Jak shrnout, co jsem při své práci s $\text{T}_{\text{E}}\text{X}$ em a obrázky zjistil? Je to rozsáhlá problematika a pro vkládání obrázků existuje celá řada nástrojů, z nichž většina potřebuje další podpůrné programy. Stejně tak je na nástroje bohatá oblast tvoření vlastních obrázků, kde můžeme nalézt vše, co můžeme při vytváření obrázků potřebovat.

Já osobně vidím jako nejuniverzálnější prostředek PS soubory, a to jak pro jejich výhody při práci s obrázky v $\text{T}_{\text{E}}\text{X}$ u, tak i pro jejich značnou univerzálnost a meziplatformovou stabilitu. Zároveň také osvitové jednotky a lepší tiskárny stejně vyžadují výstup ve formátu PostScript, a tak se převodu DVI souboru stejně nevyhneme.

Rovněž bez makra `FLOATFIG` je vkládání obrázků takřka nemyslitelné.

Je vidět, že ač není $\text{T}_{\text{E}}\text{X}$ pro vkládání obrázků přímo určen, dokáže se s nimi velmi obstojně vypořádat.



Uff, to byla ale práce!

PostScriptové fonty pro ty, co o nich moc nevědí

PAVEL HEROUT

1. Upozornění

- Tento článek má sloužit pro ty, kteří mají o dané problematice jen mlhavé znalosti, ale rádi by se dozvěděli, jak lze **jednoduše** v $\text{T}_{\text{E}}\text{X}$ u využít PostScriptových fontů, případně jak to vše funguje a jak lze začít používat nové dosud neznámé fonty. Pokud vás začne daná

problematika více zajímat, doporučuji studium literatury uvedené na konci.

- Popis bude co nejkonkrétnější, z čehož vyplývá, že nemůže postihnout všechny problémy. Uvedená řešení nemusí být nutně nejelegantnější, ale to, co je zde popsáno, je funkční.
- Vše je popisováno pro instalaci „a11“¹ $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u '96 pro MS-DOS² na lokálním disku D: – základní adresář je tedy D:\EMTEX.
- Zde popisované použití PostScriptových fontů je vyzkoušené v plain $\mathcal{T}\mathcal{E}\mathcal{X}$ u i v $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u 2.09 – tento článek je sázen v $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u 2.09.

V případě $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ u 2.09 je ale nutné zdůraznit, že je to řešení „přes ruku“, což – doufám – vyplyne z následujících ukázek. Jako mnohem lepší variantu doporučuji přejít na $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ 2_ε, kde pan Wagner již vše potřebné provedl – viz [2].

- Pokud by po přečtení tohoto článku vznikl názor, že jsem cokoliv ze zmiňovaných programů nebo souborů vytvořil, pak je to názor chybný. Je potřeba zdůraznit, že **okamžitě po správné instalaci**³ $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u '96 je možné používat minimálně základní množinu 35 PostScriptových fontů.

2. Výhody a nevýhody

Protože na světě platí pravidlo, že je vždy „něco za něco“, je třeba se hned na začátku dozvědět, co nám PostScriptové fonty mohou přinést a co nás to bude stát.

Výhody

- Je možné využívat písem, která jsou všeobecně uznávaná i mimo $\mathcal{T}\mathcal{E}\mathcal{X}$ ovskou komunitu, čímž např. odpadají problémy s množením textů méně kvalitní rozmnožovací technikou, kde CM fonty nebyly příliš výhodné.

¹Viz [1] a <ftp://math.feld.cvut.cz/pub/cstex/msdos/@ctimne.txt>. Téměř vše je ale také zahrnuto do instalace „mini“, nikoliv však „small“.

²Viz <ftp://math.feld.cvut.cz/pub/cstex/msdos>

³Provedené **po** 19.3.1996 – viz:

<http://math.feld.cvut.cz/olsak/cstex/zmeny>

- Tato písma lze – je-li pro to dobrý důvod – zcela volně míchat s fonty T_EXu.
- Výstup lze bez problémů tisknout v optimální kvalitě na libovolném PostScriptovém zařízení. Odpadá tedy nutnost vytvářet různé výstupní soubory pro tiskárnu 300 DPI, 600 DPI, popř. připravovat objemné bitové mapy (* .pk) pro jiné zařízení, např. osvitovou jednotku.
- Výsledné tiskové soubory budou nezanedbatelně menší⁴, což může být v mnoha případech jedno z rozhodujících kritérií.

Nevýhody

- Nevýhodou, na kterou brzy narazíte, je, že při přípravě dokumentu nelze beze zbytku použít standardní zobrazovač .dvi souboru – **dvistr**. To znamená, že jsou vygenerovány bitmapy nejběžnějších PostScriptových fontů v nejběžnějších rozlišeních. Pohybujete-li se mimo oblast „*nejběžnější*“, pak **dvistr** některé znaky nezobrazí.
- Řešením předchozího problému je tisk do PostScriptu a následné prohlížení výsledného .ps souboru. Bohužel ale ve standardní instalaci C_ST_EXu '96 pod MS-DOSem dosud není zobrazovač PostScriptových dokumentů – nejčastěji **Ghostscript** s **Ghostview**⁵, takže dokument nelze po konverzi do PostScriptu jednoduše v MS-DOSu na obrazovce prohlížet.
- Budete-li si výsledný PostScriptový soubor prohlížet pomocí zmíněného **Ghostview**, zjistíte často, že výsledek není optimální. Písmena jsou různě přes sebe atp. Naštěstí je to problém pouze náhradních fontů použitých tímto zobrazovačem a výsledný dokument na tiskárně bude správně, ovšem spolehnout se na „*co vidím, to dostanu*“, není možné.⁶
- Za určitou nevýhodu lze také považovat to, že použitím PostScriptových fontů dostáváme k dispozici desítky dalších písem, které můžeme libovolně míchat asi tak, jako když pejsek a kočička vařili dort.

⁴PostScriptový soubor s tímto článkem pro tiskárnu 600 DPI by měl při použití **jen interních** PS fontů jen asi 140 kB.

⁵Viz např.:

<ftp://ftp.muni.cz/pub/tex/CTAN/support/ghostscript/aladdin/g353dos.zip>

⁶Máme-li ovšem k dispozici originální PostScriptové fonty, lze **Ghostscript** nakonfigurovat tak, že je bude umět použít, a pak bude náš dokument zobrazen korektně.

Vynucené použití jen Computer Modern fontů dosud způsobovalo, že T_EXem vyrobené dokumenty byly „decentní“⁷, čímž míním fakt, že v jednoduchosti je krása. Použitím PostScriptových fontů nastává možnost pomocí T_EXu generovat stejně „řhavé“ tiskoviny, které jsou sázeny jinými programy za použití základní ideje: „Čím více fontů, tím lépe.“

3. Jak začít

Jako naprosto první krok na začátku vřele doporučuji zkopírovat si někam do svého pracovního adresáře soubor `test8z.tex` z adresáře `D:\EMTEX\DOC\A2AC`. V něm je testována většina standardních fontů. Tento soubor přeložte plainT_EXem, zkonvertujte jej do souboru `test8z.ps` a vytiskněte na svojí PostScriptové tiskárně. Pokud jakákoliv z těchto akcí neproběhne dobře, máte zřejmě chybnou instalaci C_ST_EXu '96 a výrazně doporučuji nepouštět se do žádných dalších pokusů s PostScriptem a pokusit se o novou instalaci minimálně souborů ze `cspfont.zip`.

Pracujeme-li v plainT_EXu a nemáme-li žádné další zvláštní požadavky, pak stačí jen napsat

```
\input cavantga
```

A od této chvíle sázíme českým⁸ fontem Avant Garde Book. Dokonce fungují i všechny obvyklé přepínače, takže po `\bf za-`
`čneme psát tučně, po \it to bude skloněné písmo a po \rm`
se dostaneme znovu do základního písma. Přepínač `\tt` zapne neproporcionální písmo.

Totéž analogicky platí i pro další rodiny písem⁹, dle následující tabulky:

⁷I přesto, že je dostupných asi 25 českých fontů z Computer Modern.

⁸Přesněji „počeštěným“.

⁹Ukázky těchto písem viz v tabulce na str. 50.

<i>soubor</i>	<i>rodina písma</i>
cavantga	Avant Garde Book
cbookman	Bookman
chelvet	Helvetica
cncent	New Century Schoolbook
cpalatin	Palatino
ctimes	Times Roman

Tento postup funguje i v L^AT_EXu 2.09, ovšem pouze při použití desetibodového písma. To znamená, že v `\documentstyle[]` nesmí být velikost 11pt nebo 12pt. Například:

```
\documentstyle[12pt]{article}
```

znemožní použití tohoto jednoduchého přepnutí do jiného písma.

Budeme-li blíže zkoumat, jak je to zařízeno, zjistíme, že v adresáři:

```
D:\EMTEX\TEXINPUT\CSPLAIN
```

jsou uloženy soubory `cavantga.tex` atd. například s tímto obsahem:

```
\font\tenrm=pagk8z at 10pt
\font\tenbf=pagd8z at 10pt
\font\tenit=pagko8z at 10pt
\font\tentt=pcrr8t at 10pt
\let\tenzl=\tenit
\tenrm
```

Informace pro nás podstatná je v tomto okamžiku: `at 10pt`

Z tohoto nastavení velikosti je možné odvodit nevýhody uvedeného jednoduchého postupu:

- Není možné okamžitě vysadit zvoleným fontem větší ani menší písmo – to nás v plainT_EXu ale nepřekvapí. Nepříjemné je to však v L^AT_EXu, kde jsme zvyklí na přepínače typu `\large`, `\small`, ...
- Není možné použít fonty typu „*polotučný skloněný*“, které jsou samozřejmě součástí rodin PostScriptových písem.
- Jako další nevýhodu lze uvést, že není možné použít ostatních interních PostScriptových fontů – např. symbol uvozující tyto výčtové položky, ani množství existujících externích fontů.¹⁰

¹⁰Vysvětlení pojmů „interní“ a „externí“ viz dále.

Z těchto nevýhod lze vyvodit, že uvedený jednoduchý postup je možné bez dalšího úsilí použít jen pro texty sázené desetibodovým písmem, kde vystačíme s přepínači `\rm`, `\bf` a `\it`.

Na druhé straně není dobré tento způsob zcela zavrhnout, protože ho lze bez problémů kombinovat s dále uvedenými způsoby.

Poznámka:

Není-li velikost 10pt vhodná, lze v plain \TeX u použít triku:

```
\magnification\magstep1
\input cavantga
```

Kdy dostaneme základní písmo o velikosti 12 bodů.

4. Jaké máme dostupné PostScriptové fonty a jak se značí

Fonty se často sdružují do *rodin písem*¹¹, což je skupina – často čtyř – fontů podobného jména a stejných kresebných vlastností. Vždy jeden font je základní – v terminologii \TeX u bychom řekli `\rm`. Další tři fonty jsou *vyznačovací*. Jedná se o *skloněné písmo* (`\it`), **tučné písmo** (`\bf`) a **skloněné tučné písmo**, které nemá v \TeX u přímý přepínač.

Máme-li současné PostScriptové zařízení – nejčastěji laserovou tiskárnu – pak bude s vysokou pravděpodobností splňovat vlastnosti PostScriptu **Level 2**. Toto označení – mimo mnoha jiných podstatnějších rozdílů od **Level 1** – znamená, že uvnitř tiskárny je uloženo 35 interních PostScriptových fontů.¹²

S koupí tiskárny pak oficiálně vlastníme tyto fonty a můžeme je využívat. Protože fonty jsou uloženy v tiskárně, není třeba uvádět jejich popis do výsledného tiskového souboru a tento soubor tak bude podstatně kratší.

Zdůrazněme ještě jednou, že všechny tyto fonty – respektive jejich počestěné varianty – je možné po instalaci $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u okamžitě využít.

Kromě zmíněných 35 fontů existuje celá spousta dalších fontů, kterým se říká externí, protože je nutné popis těchto fontů před tiskem do

¹¹V \TeX u jsme dosud používali rodinu písma Computer Modern.

¹²V laserových tiskárnách s PostScriptem **Level 1** jich bylo jen 13. Počet fontů ale není pro určení **Level** x vždy směrodatný. Pozn. red.: rozdíl je zejména v tom, že Level 2 má více příkazů než Level 1.

tiskárny zavést. Co je nutné učinit, potřebujeme-li v $\text{T}_{\text{E}}\text{X}$ u využít i tyto fonty, bude popsáno na str. 59.

Poznámka:

Při práci s PostScriptovými fonty se často setkáme i s jejich jiným označením – formát/font Adobe Type 1 nebo jen Type 1. Toto označení znamená, že je font popsán soustavou matematických křivek.

Na rozdíl od Type 1 existuje formát nazvaný Type 3, který popisuje daný font jako PostScriptovou proceduru schopnou realizovat vykreslení znaků fontu buď z bitmapy nebo z matematického popisu.¹³ Formátem Type 3 se nebudeme dále zabývat.

Kromě těchto dvou nejznámějších formátů rozeznává PostScript **Level 2** ještě množství dalších, vždy označených Type x , např. Type 0 jsou kompozitní fonty¹⁴, Type 42 jsou TrueType fonty používané v MS Windows atd.

Již zmíněných 35 interních fontů se dělí do osmi rodin po čtyřech fontech a dále pak tři samostatné fonty, které netvoří rodinu s žádným dalším písmem. Zkratky použitelné v $\text{T}_{\text{E}}\text{X}$ u, jména a ukázky těchto fontů jsou uvedeny v následující tabulce.

Poznámka:

V tabulce nejsou uvedeny všechny ukázky všech PostScriptových fontů, které lze po instalaci $\mathcal{C}\mathcal{S}\text{T}_{\text{E}}\text{X}$ u použít. Jsou například vynechány všechny verze *Small-Caps* (kapitálek¹⁵). Uvedena jsou jen písma, která se pravděpodobně zobrazí na PostScriptové tiskárně – po určité manipulaci s ovládacími prvky – jako *PostScript Typeface List*. Libovolné jiné dostupné fonty můžete také používat – viz str. 59.

Co znamenají ty „tajemné“ zkratky v označení písma? Jedná se o konvenci v označování fontů tak, aby jejich název nepřesáhl osm

¹³Tohoto formátu využívá např. **dvips**, který konvertuje sazbu CM fonty do PostScriptu.

¹⁴Odpovídající přibližně $\text{T}_{\text{E}}\text{X}$ ovským virtuálním fontům.

¹⁵„Velkých písmen ve velikosti malých písmen“ – typografové promiňte! Příklad: ČESKÉ BUDĚJOVICE.

JJ	jméno fontu
D	tloušťka fontu – duktus (<i>weight</i>) – tlustý, tenký, ...
V	varianta fontu – skloněný, patkový, bezpatkový, ...
R	rozpětí fontu (<i>expansion</i>) – zúžený, rozšířený, ...

Každá položka má množství možností, které zde nemá cenu vyjmenovávat (podrobnosti viz [3]). Pokusíme se jen ukázat na příkladech několik typických jmen pro ukázkou systému pojmenování.

Názvy z předchozí tabulky začínají vždy písmenem „P“, což znamená PostScript, a většinou končí znaky „8z“, což znamená rozšíření PostScriptových *roman* fontů podle kódování ISO 8859-2, tj. prakticky \mathcal{C} Sfonty. U rodiny Courier jsou znaky „8t“, představující PostScriptové *typewriter* fonty. Tyto znaky tedy nebudeme dále uvádět.

Poznámka:

Nekončí-li označení fontu „8z“ nebo „8t“ – např. pzcmi¹⁷ – nelze jej použít pro sazbu češtiny zde popisovaným způsobem.¹⁸

Co je nutné ještě vědět, je skutečnost, že u některých fontů se některé písmeno v názvu může někdy vynechat, považuje-li se za samozřejmé.

agk	Avant Garde Book
hvr	Helvetica r=normal (bezpatkové písmo)
hvrn	Helvetica r=normal Narrow (úzký)
tmr	Times Roman r=normal (patkové písmo)
agd	Avant Garde Demi (polotučné písmo)
hvb	Helvetica Bold
agko	Avant Garde Oblique (skloněné písmo)
tmri	Times Italic
agdo	Avant Garde Demi Oblique

¹⁷Redakčním zásahem byl do tabulky dosazen počestěný Zapf Chancery Medium Italic (pzcmi8z)

¹⁸Existují totiž virtuální fonty, které nemají na konci znaky „8z“ či „8t“ a jsou v kódování Cork – viz [4] a [5]. Tyto fonty lze použít pro sazbu češtiny, ovšem v \mathcal{C} S \TeX u pro to není jednoduchá uživatelská podpora. Není to však případ fontů uváděných v předchozí tabulce.

hvbo **Helvetica Bold Oblique**
tmbi **Times Bold Italic**

Striktně vzato – Helvetica by měla mít označení hvrrrr, což představuje: **Helvetica** normální tloušťka (**r**=regular) bezpatkové písmo (**r**=normal) normální šíře (**r**=regular).

Pragmaticky vzato – většinou si nebudeme vybírat písmo podle tohoto označení, ale naopak – z tabulky na str. 50 si podle vzorku písma zjistíme použité označení.

5. Praktické použití PostScriptových fontů v plain \TeX u

Chceme např. sázet v plain \TeX u desetibodovým písmem z rodiny Avant Garde.

Z tabulky na str. 46 zjistíme příslušný název pomocného souboru pro tuto rodinu písma a napíšeme `\input cavantga`

Dále budeme chtít používat tutéž rodinu pro nadpisy větší než 10pt a například poznámky menší než 10pt. Celý soubor by vypadal například takto:

```
\input cavantga
\font\velkynadpis=pagd8z at 17pt % hlavni nadpis
\font\nadpis=pagd8z at 14pt % nadpis
\font\pozn=pagk8z at 8pt % poznamka
\font\poznit=pagko8z at 8pt % poznamka italika
```

```
{\velkynadpis Fonty v~PostScriptu} \par
{\nadpis Bezpatkové fonty} \par
Jako základní {\it bezpatkový\}
font je použita rodina {\bf Helvetica}.
```

```
{\nadpis Patkové fonty} \par
Jako základní {\it patkový\} font
je použita rodina {\bf Times Roman}.%
\footnote{$^1$}{\pozn Obě rodiny
```

```
představují {\poznit proporcionální\/}  
písmo.}}  
\bye
```

Poznámka:

Číslo ¹, uvozující poznámku pod čarou (`\footnote{ $\$^1\$$ }`), bude vždy v Computer Modern fontech, protože veškerá matematika zůstává v režii sazby těmito fonty. Máme-li ovšem vhodný PostScriptový font pro sazbu matematiky, lze i toto odstranit – viz [7].

Situace se maličko zkomplikuje, budeme-li chtít použít jinou velikost základního písma, než 10pt. Příklad je pro základní písmo 12pt.

```
\font\rm=pagk8z at 12pt  
\font\bf=pagd8z at 12pt  
\font\it=pagko8z at 12pt  
\rm  
\font\velkynadpis=pagd8z at 21pt % hlavni nadpis  
\font\nadpis=pagd8z at 17pt % nadpis  
\font\pozn=pagk8z at 10pt % poznamka  
\font\poznit=pagko8z at 10pt % poznamka italika  
  
{\velkynadpis Fonty v~PostScriptu} \par  
...  
\bye
```

6. Praktické použití PostScriptových fontů v \LaTeX u 2.09

Poznámka:

Jak uvidíme z obou ukázek, ztrácíme použitím PostScriptových fontů jednu z důležitých výhod \LaTeX u – automatické změny velikosti písma v závislosti na kontextu.¹⁹ Pokud hodláme PostScriptové fonty více využívat, je vhodnější přejít na $\LaTeX 2_{\epsilon}$, kde je toto všechno zařízeno automaticky.

¹⁹Nápravy lze dosáhnout nepříliš jednoduše např. pokusy se soubory `art10.sty`, `lfontscs.tex`, ale není to způsob hodný doporučení.

Ukázka pro základní písmo 10pt.

```
\documentstyle[czech]{article}
\input cavantga
\font\velkynadpis=pagd8z at 17pt % hlavni nadpis
\font\nadpis=pagd8z at 14pt      % nadpis
\font\pozn=pagk8z at 8pt        % poznamka
\font\poznit=pagko8z at 8pt     % poznamka italika
```

```
\begin{document}
{\velkynadpis Fonty v~PostScriptu} \par
{\nadpis Bezpatkové fonty} \par
Jako základní {\it bezpatkový\} font
je použita rodina {\bf Helvetica}.
```

```
{\nadpis Patkové fonty} \par
Jako základní {\it patkový\} font
je použita rodina {\bf Times Roman}.%
\footnote{{\pozn Obě rodiny představují
{\poznit proporcionální\} písmo.}}
\end{document}
```

Ukázka pro základní písmo 12pt.

```
\documentstyle[czech,12pt]{article}
\font\rm=pagk8z at 12pt
\font\bf=pagd8z at 12pt
\font\it=pagko8z at 12pt
\rm
\font\velkynadpis=pagd8z at 21pt % hlavni nadpis
\font\nadpis=pagd8z at 17pt     % nadpis
\font\pozn=pagk8z at 10pt       % poznamka
\font\poznit=pagko8z at 10pt   % poznamka italika
```

```
\begin{document}
{\velkynadpis Fonty v~PostScriptu} \par
...
\end{document}
```


7. Použití speciálních symbolů

Z tabulky na str. 50 je vidět, že PostScriptové fonty také zahrnují fonty Symbol a Zapf Dingbats²⁰. Naskytá se otázka, jak použít nějaký symbol, když není přístupný z klávesnice. Řešení je jednoduché – v následujících tabulkách je nutné požadovaný symbol nalézt, zjistit jeho pořadové číslo a to pak použít jako parametr primitivu `\char`.

Například symbol ♣ byl vysazen příkazem:

```
{\zpd\char37}
```

kterému předcházelo natažení fontu: `\font\zpd=pzdr at 10pt`

8. Jak to všechno funguje

Pokud se chcete dozvědět detailně, jak to všechno funguje, přečtěte si podkapitulu 9.4 v knize **Typografický systém T_EX** ([4]). Zde se omezíme jen na základní informace a souvislosti.

Mějme soubor `pokus.tex` s příkazem:

```
\font\muj=pagk8z at 30pt
```

a pak po příkazu `\muj` napíšeme nějaký text. Při překladu souboru `pokus.tex` T_EXem se vyhledá v adresáři `D:\EMTEX\TFM\PS.XL2` soubor `pagk8z.tfm`. Jedná se o metriku fontu **AvantGarde Book** v kódování dle ISO 8859-2.²¹

Údaje získané z tohoto souboru jsou pro T_EX naprosto dostačující – T_EX nezajímá, jak jednotlivé znaky vypadají, ale jen jak jsou velké. Výsledkem je soubor `pokus.dvi`.

Po překladu se pokusíme spustit běžně používaný zobrazovač **dviscr**, tedy prohlédnout si `pokus.dvi`. Zde dojde k problému, protože **dviscr** nezná bitové mapy fontu **AvantGarde Book** ve velikosti 30pt v požadovaném rozlišení a nabídne nám tedy možnost dodatečného

²⁰Občas jsou zmiňovány jako „Pi fonty“.

²¹Jak byla tato metrika získána se dozvíme později.

Tabulka znaků fontu Symbol – psyr

	0	1	2	3	4	5	6	7	8	9
0										
10										
20										
30				!	∇	#	∃	%	&	ə
40	()	*	+	,	-	.	/	0	1
50	2	3	4	5	6	7	8	9	:	;
60	<	=	>	?	≡	A	B	X	Δ	E
70	Φ	Γ	H	I	∅	K	Λ	M	N	O
80	Π	Θ	P	Σ	T	Υ	ζ	Ω	Ξ	Ψ
90	Z	[∴]	⊥	_		α	β	χ
100	δ	ε	φ	γ	η	ι	φ	κ	λ	μ
110	ν	ο	π	θ	ρ	σ	τ	υ	ω	ω
120	ξ	ψ	ζ	{		}	~			
130										
140										
150										
160		Υ	'	≤	/	∞	f	♣	♦	♥
170	♠	↔	←	↑	→	↓	°	±	"	≥
180	×	∞	∂	•	÷	≠	≡	≈	...	
190	—	⌋	⌘	Ⓔ	Ⓕ	Ⓖ	⊗	⊕	∅	∩
200	∪	⊃	⊇	⊘	⊂	⊆	∈	∉	∠	∇
210	®	©	™	∏	√	·	¬	∧	∨	↔
220	⇐	↑	⇒	↓	◇	⟨	®	©	™	Σ
230	(({		
240		>	∫	∫		J)		J	
250		J		{	J					

generování těchto bitmap. Přijmeme-li jeho „šlechtnou“ nabídku²², zdržíme se o několik vteřin, a protože implicitní nastavení $\text{\texttt{CSTEX}}$ neumožňuje automatické generování bitmap z PostScriptových fontů,²³ **dvicr**

²²Jak jsme zvyklí při generování chybějících bitmap CM fontů.

²³Důvod viz str. 8 v **Pár poznámek...** – [1].

Tabulka znaků fontu Zapf Dingbats – pzdr

	0	1	2	3	4	5	6	7	8	9
0										
10										
20										
30										
40										
50										
60										
70										
80										
90										
100										
110										
120				‘	’	“	”			
130										
140										
150										
160										
170			①	②	③	④	⑤	⑥	⑦	⑧
180	⑨	⑩	①	②	③	④	⑤	⑥	⑦	⑧
190	⑨	⑩	①	②	③	④	⑤	⑥	⑦	⑧
200	⑨	⑩	①	②	③	④	⑤	⑥	⑦	⑧
210	⑨	⑩								
220										
230										
240										
250										

nám po chvíli neužitečné činnosti opět nabídne tutěž službu.

Po odpovědi, že nechceme generovat bitmapy, zobrazí **dviscr** co může. Fráze „co může“ znamená, že uvidíme všechny znaky z *C*fontů (byly-li nějaké použity) a prázdné šedě orámované boxy, které nahrazují znaky, od kterých nejsou bitmapy k dispozici. Jedná se o text, který jsme vysázeli fontem *AvantGarde Book*. Toto prohlédnutí si souboru

pokus.dvi je dobré pro první ověření, že se nám něco podařilo vysázet, samozřejmě však nedostačuje.

Pro získání viditelného výsledku je vhodné použít příkazu z menu: Print/Output to pokus.ps, který vyvolá program **dvips**.²⁴

Dvips se ze souboru pokus.dvi dozví, že byl použit font s metrikou pagk8z.tfm. V adresáři D:\EMTEX\FONTS\VF\A2AC²⁵ k němu hledá jeho odpovídající virtuální font pagk8z.vf. Oba fonty musí mít stejné jméno (pagk8z) a liší se jen příponami (.tfm a .vf). V souboru pagk8z.vf si **dvips** přečte, že existuje soubor rpagk.tfm v adresáři D:\EMTEX\TFM\DVIPS.²⁶

Všimněte si, že v názvu fontu s metrikou přibylo na začátku písmeno „r“. To znamená, že se jedná o *raw* font, tedy skutečný – ne virtuální – font v původním kódování PostScriptových fontů – ne tedy v kódování ISO 8859-2. Z tohoto důvodu také na konci jména fontu zmizely znaky „8z“. Metriky z tohoto souboru v daném kódování použije **dvips**.

V souboru pagk8z.vf je pro **dvips**, mimo informace o existenci rpagk.tfm, také uvedeno, že vlastní popis znaků fontu je svázán s názvem rpagk, který je součástí souboru:

```
D:\EMTEX\DATA\DVIPS\psfonts.map.
```

Podle odkazu rpagk se **dvips** z psfonts.map na téže řádce dozví, že skutečné PostScriptové jméno fontu je AvantGarde-Book.

Protože v psfonts.map není na zmíněné řádce uvedena žádná další informace, je to pro **dvips** znamení, že se jedná o interní font tiskárny²⁷. **Dvips** tedy jen do výsledného výstupního souboru pokus.ps zapíše, že je použit font AvantGarde-Book a text vysázený tímto fontem uloží do pokus.ps. Vlastní digitalizace použitých znaků v textu je pak prováděna až při tisku prostředky tiskárny.

Získaný soubor pokus.ps můžeme buď vytisknout na PostScriptové tiskárně nebo zobrazit na obrazovce pomocí některého PostScriptového prohlížeče – např. již zmiňovaného **Ghostscript** s **Ghostview**, máme-li ho ovšem někde nainstalován.

²⁴Předtím ale musíme zvolit tiskárnu PS_Laser, nemáme-li ji implicitní.

²⁵Proč zrovna v tomto adresáři viz dále.

²⁶Správně by měl ležet v adresáři D:\EMTEX\TFM\PS.RAW, ale takto je **dvips** distribuován.

²⁷Jeho matematický popis je uložen v ROM tiskárny.

9. Co si počít, chceme-li více fontů

Jsou tři možné odpovědi na tuto otázku:

❶ **Nestačí nám náhodou standardní fonty?**

Je vhodné prohlédnout si tabulku na str. 50, případně přeložit a vytisknout soubor `D:\EMTEX\DOC\A2AC\test8z.tex`, kde jsou větší ukázky písem uvedených ve zmíněné tabulce. Pro naprostou většinu požadavků by měly stačit.

❷ **Použít to, co už je hotovo.**

Jsme-li nezvratně přesvědčeni, že naše dílo musí být vysázeno ještě jinými, než dosud popsanými fonty, pak je vhodné provést průzkum adresářů: `D:\EMTEX\TFM\PS.XL2` a `D:\EMTEX\TFM\DVIPS`. Metriky zde nalezené – například `pagkc8z.tfm`, tj. **AvantGarde-Book Small Caps** – vyzkoušet na jednoduchém příkladě a pak použít.

Výhoda tohoto postupu je, že nemusíme zasahovat do konfigurace $\zeta\text{T}_{\text{E}}\text{X}_{\text{u}}'96$.

❸ **Generovat nové fonty.**

Před touto aktivitou musím z vlastní neblahé zkušenosti důrazně varovat. Určitě totiž budete mít potíže vzniklé tím, že při přípravě někde uděláte chybu a pak tím, že tato činnost s kompletním testováním nových fontů zabere mnoho času. Neberete-li toto varování vážně, či je potřeba nových fontů tak nutková, pak čtete dále a obrňte se trpělivostí.

Dalším problémem je, že pro tyto fonty nejsou vygenerovány bitmapy (`.pk`), takže **dvips** znaky fontu nezobrazí.²⁸

První a nezbytnou podmínkou je nutnost mít požadovaný PostScriptový font. Tyto fonty jsou licencované a musí se za ně platit, to znamená, že nejsou²⁹ volně přístupné ve veřejných archivech.

Není ovšem hned třeba opouštět pracoviště s průhlednou záminkou nákupu nových fontů. Můžeme mít totiž ve své organizaci tyto fonty legálně zakoupené aniž o tom víme. Začteme-li se do popisu **dvips** – viz

²⁸Řešením pak je použít program **ps2pk** pro přípravu bitmap – viz:

`ftp://ftp.muni.cz/pub/tex/CTAN/fonts/utilities/ps2pk/ps2pk15.zip`

²⁹Pokud je tam naleznete, vězte, že existuje mnoho fontů **ve formátu Type 1**, ale žádnou firmou **nelicencovaných**. Může se jednat o kvalitní písma, stejně tak jako o písma, s nimiž budou potíže.

[3], najdeme tipy, kde lze takové soubory nalézt³⁰, protože jsou placenou součástí např. X-Windows systému.

Dejme tomu, že jsme se zhlédli v písmu *Souvenir Light*. Pro zdárnou činnost potřebujeme tento font popsáný dvěma soubory. Prvním je metrika s příponou `.afm` – *Adobe Font Metric*. Druhým je matematický popis tohoto fontu. Zde máme dvě možnosti – buď je k dispozici soubor `.pfb` nebo `.pfa`. První označuje *PostScript Font Binary* a druhý *PostScript Font ASCII*, přičemž použít můžeme libovolný z nich.

Poznámka:

Ve skutečnosti ale potřebujeme celkem **osm** souborů, protože určité budeme chtít i **tučnou** verzi, *italiku* a když už to všechno děláme, tak proč ne i *tučnou italiku*.

První problém nastává se jmény souborů. Získáme totiž pravděpodobně soubory `Souvenir-Light.afm`, `Souvenir-Demi.afm`, `Souvenir-LightItalic.afm`, `Souvenir-DemiItalic.afm` (plus tytéž soubory s příponou `.pfa`).

Jména souborů jsou krásně významová, ale MS-DOS má nepříjemné omezení délky jména na osm znaků. Je tedy nutné prostudovat způsob tvoření krátkých jmen fontů – buď z tabulky na str. 50 a následného popisu, nebo lépe z popisu **dvips** (viz [3]).

Soubory tedy přejmenujeme na `psol.afm`, `psod.afm`, `psoli.afm` a `psodi.afm`³¹ (plus tytéž soubory s příponou `.pfa`). Dále je dobré zkontrolovat, zda v adresáři `D:\EMTEX\TFM\PS.XL2` už nejsou tato jména použita pro jiný font a pokud ne, přeneseme všech osm souborů na své PC do nově vytvořeného adresáře `D:\PRIPRAVA`.

Dalším problémem je, že jsme získali PostScriptové fonty, ale nejsou počestěné³².

Pro tento účel připravil pan Olšák program **a2ac**³³, který je uložen v `D:\EMTEX\BIN`. K němu potřebujeme ještě korekční tabulku `cscorr.tab` a dva soubory pro kódování `xl2.enc` a `xt2.enc` – všechny tři nalezneme v adresáři `D:\EMTEX\DATA\A2AC`.

³⁰Například `/usr/lib/X11/fonts/Type1Adobe`

³¹**Pozor** – jméno by nemělo být delší než šest písmen!

³²Máme-li náhodou počestěné, přeskakujeme následující popis programu **a2ac**.

³³Dokumentace k programu je v `D:\EMTEX\DOC\A2AC\A2ac-cz.doc`

Všechny čtyři soubory doporučuji překopírovat do pomocného adresáře `D:\PRIPRAVA`, aby byly co nejmenší problémy. Činností **a2ac** dostaneme počestěnou metriku, tedy např. soubor `psol8z.afm`.

Způsob spuštění programu viz dále, zde jen pragmaticky, proč musíme mít dva soubory pro kódování, které navíc budou potřebné až pro program **afm2tfm**.

Potřebujeme-li mít v českých textech znak zpětného lomítka (*backslash* `\`) ve stejném fontu, použijeme kódovací soubor `xt2.enc` a výsledný soubor bude mít název `psol8t.ttf`. Běžnější je ale použití kódovacího souboru `xl2.enc`, čímž získáme soubor s názvem `psol8z.afm`, obsahující ligatury, atd. Není dokonce vyloučeno, abychom připravili a dále používali oba soubory, tedy `psol8t.afm` a `psol8z.afm`.

Jinak řečeno – soubory `*8t` jsou kódovány podle Computer Modern *typewriter* fontů + ISO 8859-2 a soubory `*8z` podle Computer Modern *roman* fontů + ISO 8859-2. Přesný rozdíl mezi nimi viz **Dodatek F** ve [4].

Další, co je nutné vyřešit, je výroba metriky *raw* fontu (například `rpsol.ttf`), kterou je schopen rozeznat **dvips**. Dále pak příprava virtuálního popisu počestěného fontu (např. `psol8z.vpl`), ze které se připraví virtuální font. Pro obě tyto činnosti použijeme program **afm2tfm** z adresáře `D:\EMTEX\BIN`³⁴ Tento soubor opět zkopírujeme do adresáře `D:\PRIPRAVA`

Práce nekončí – dále je nutné z virtuálního popisu fontu vytvořit metriku, kterou rozeznává \TeX (např. `psol8z.ttf`) a virtuální font pro **dvips** (např. `psol8z.vf`). K tomu nám poslouží program **vptovf** z adresáře `D:\EMTEX\BIN`, který opět zkopírujeme do `D:\PRIPRAVA`.

Protože přípravu fontů budeme opakovat čtyřikrát, je vhodné připravit si dávkový soubor `konv.bat`, s tímto obsahem:

```
@echo off
a2ac %1.afm cscorr.tab %18z.afm > nul
afm2tfm %18z.afm -t xl2.enc -v %18z r%1 >>fonty.txt
vptovf %18z.vpl %18z.vf %18z.ttf > nul
```

Tento dávkový soubor implicitně využívá kódovací soubor `xl2.enc`, takže výsledné metriky budou v kódování *roman* \mathcal{S} fontů.

³⁴Dokumentace k němu je v popisu **dvips**.

Nyní již stačí spustit čtyřikrát dávkový soubor `konv.bat`, tedy:

```
D:\PRIPRAVA>konv psol
D:\PRIPRAVA>konv psod
D:\PRIPRAVA>konv psoli
D:\PRIPRAVA>konv psodi
```

Kromě všech metrik a virtuálních souborů dostaneme také soubor `fonty.txt`, ve kterém jsou skutečné názvy použitých fontů a budeme ho potřebovat pro modifikaci souboru `psfonts.map`.

Nyní je nutné přesunout vzniklé soubory na správná místa takto:

```
p*8z.tfm → D:\EMTEX\TFM\PS.XL2
rp*.tfm → D:\EMTEX\TFM\PS.RAW
p*8z.vf → D:\EMTEX\FONTS\VF\A2AC
```

Soubory `p*.vpl` a `p*.afm` můžeme vymazat.

Poznámka:

Název adresáře `D:\EMTEX\TFM\PS.XL2` vznikl z *Extended ISO Latin 2*, který zahrnuje jak kódy „8z“ tak i kódy „8t“.

Adresář `D:\EMTEX\FONTS\VF\A2AC` je použit proto, že virtuální fonty byly počestěny pomocí programu **a2ac**.

Po tomto přesunu zbudou v adresáři `D:\PRIPRAVA` ještě soubory `.pfa`. Pro ně vytvoříme adresář `D:\EMTEX\PSFONTS` a do něho je všechny přesuneme.

Nyní nám v adresáři `D:\PRIPRAVA` zůstává jen soubor `fonty.txt`, který má momentálně tento obsah:

```
rpsol Souvenir-Light
rpsod Souvenir-Demi
rpsoli Souvenir-LightItalic
rpsodi Souvenir-DemiItalic
```

Tento soubor editorem upravíme do tvaru:

```
rpsol Souvenir-Light < d:\emtex\psfonts\psol.pfa
rpsod Souvenir-Demi < d:\emtex\psfonts\psod.pfa
rpsoli Souvenir-LightItalic < d:\emtex\psfonts\psoli.pfa
rpsodi Souvenir-DemiItalic < d:\emtex\psfonts\psodi.pfa
```

a jeho obsah přidáme na konec souboru `psfonts.map`, který se nachází v adresáři `D:\EMTEX\DATA\DVIPS`. Tyto řádky říkájí programu

dvips, že fonty nejsou interní, ale že jejich matematický popis má hledat v příslušných souborech adresáře `D:\EMTEX\PSFONTS`

Zde je *malá ukázka*, že od **této chvíle** jsou fonty rodiny Souvenir **připraveny k použití**.

Dále zkopírujeme soubor `D:\EMTEX\DOC\A2AC\test8z.tex` do svého adresáře a podle něj připravíme soubor `pokus.tex`, kde použijeme všechny námi vytvořené fonty. Tento soubor přeložíme, vytiskneme a výsledek pozorně prohlédneme. Teprve nyní je možné nově generované fonty používat.

10. Problémy a dobré rady

Problémů může nastat spousta. Zde budou zmíněny některé, se kterými jsem se setkal a – většinou díky laskavé pomoci a radě pánů Olšáka a Wagnera³⁵ – vyřešil nebo alespoň našel vysvětlení.

❶ Neměňte konfigurační soubory

Popsaná instalace PostScriptových fontů a spolupracujících programů v $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u '96 je funkční. **Nevylepšujte ji metodou pokusu a omylu!**

❷ Něco zdánlivě chybí

Budete-li pátrat, kde jsou uloženy některé soubory – např. *raw* metricky od `ptmr8z.tfm` – nenajdete je. Nepochybně ale panice, tyto soubory neexistují, protože při vytváření programem **afm2tfm** je vytvořen odkaz na původní nepočesťený soubor – např. `rptmr.tfm`.

❸ Není možné vysázet zpětné lomítko \

V některých fontech skutečně tento znak není, respektive měl by být jen ve fontech `p*8t.tfm`. Použijte zpětné lomítko z $\mathcal{C}\mathcal{S}$ fontů nebo z PostScriptových fontů z rodiny Courier. Potřebujete-li bezpodmínečně zpětné lomítko z téhož fontu – jako v tomto nadpise, použijte přímo *raw* font: `\font\rnbc=rpnbc at 12pt a {\rnbc\char92}`

³⁵Kterým touto cestou ještě jednou děkuji.

④ Program **afm2tfm** nefunguje

Objevíte-li při generování fontů, že se nevytvořily požadované soubory, podívejte se do souboru `D:\EMTEX\fonty.txt`. Obsahuje-li místo jména fontu `text: emx not found`, znamená to, že **afm2tfm** nemá vhodné podmínky pro práci. Nejjednodušším řešením je spustit $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$, pak příkazem `File/DOS ...` vyvolejte Norton Commander (nebo jiný podobný program) a celé generování fontů proveďte znovu. Poté se vraťte zpět do prostředí $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u.

⑤ Pozor na velká a malá písmena

Toto varování je v MS-DOSu poněkud kuriózní, ale při generování nových PostScriptových fontů je třeba používat jen malá písmena. Chybou je tedy např.: `konv PSOL`, kdy jména souborů v adresáři `D:\PRIPRAVA` získáme pomocí: `dir *.afm > a.txt` a následnou editací souboru `a.txt`.

MS-DOS malá a velká písmena nerozeznává, **dvips** ano.

⑥ Chybějící akcentovaná písmena

Při zkoušce nově vygenerovaných PostScriptových fontů můžeme zjistit, že některá akcentovaná písmena chybí – často např. „Á“. Problém je ve složitosti popisu tohoto fontu a řešení vede na změnu souboru `cscorr.tab`. Celkové řešení není často triviální – dobrá rada je: *Nepoužívejte takovýto font.*

Pokud chcete přesto provést několik pokusů, které ale nemusí vést k cíli, zkopírujte soubor `cscorr.tab`. V novém souboru nahraďte u chybějících akcentovaných znaků prefix `NC` prefixem `RC` a znovu opakujte celý postup generování fontu s tímto upraveným souborem.

⑦ Některé akcenty nejsou správně posazeny

Toto je opět problém vzniklý při generování nových PostScriptových fontů. U skloněných verzí (*italic*) některých fontů nejsou akcenty posazeny správně. Tento problém a jeho řešení je popsán v [2] – vede opět ke úpravě kopie `cscorr.tab` (podrobně viz [6]). Doporučuji opět nepoužívat tento font, dokud nezískáme jeho lepší variantu připravenou odborníkem.

Literatura

- [1] Olšák, P.: Pár poznámek k novému $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u. Zpravodaj Československého sdružení uživatelů $\mathcal{T}\mathcal{E}\mathcal{X}$ u 1/1996, 1–14,
též: <http://math.feld.cvut.cz/ftp/cstex/parpozn.ps>
- [2] Wagner, Z.: Instalace českých PostScriptových fontů. Součást souboru:
<ftp://math.feld.cvut.cz/pub/cstex/msdos/cspsfont.zip>
po rozbalení soubor `D:\EMTEX\DIST\cspsfont.drv`
- [3] Rokicki, T.: DVIPS: A $\mathcal{T}\mathcal{E}\mathcal{X}$ Driver. Součást souboru:
<ftp://math.feld.cvut.cz/pub/cstex/msdos/dvipsexex.zip>
po rozbalení soubor `D:\EMTEX\TEXINPUT\DVIPS\dvips.tex`
- [4] Olšák, P.: Typografický systém $\mathcal{T}\mathcal{E}\mathcal{X}$. $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}$, 1995.
- [5] Olšák, P.: Úvaha o fontech v $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u.
<http://math.feld.cvut.cz/ftp/olsak/bulletin/uvaha.ps>
- [6] Olšák, P.: Program `a2ac`. Zpravodaj Československého sdružení uživatelů $\mathcal{T}\mathcal{E}\mathcal{X}$ u 1/1996, 23–30, též součást souboru:
<ftp://math.feld.cvut.cz/pub/cstex/msdos/cspsfont.zip>
po rozbalení soubor `D:\EMTEX\DOC\A2AC\a2ac-cz.doc`
- [7] Olšák, P.: Dvojité hranaté závorky v matematice.
<http://math.feld.cvut.cz/ftp/olsak/bulletin/bbold.ps>

Vydalo: Československé sdružení uživatelů $\mathcal{T}\mathcal{E}\mathcal{X}$ u
vlastním nákladem jako interní publikaci

Obálka: Bohumil Bednář

Počet výtisků: 650

Uzávěrka: 4. dubna 1996

Odpovědný redaktor: Zdeněk Wagner

Tisk a distribuce: KOPP, Šumavská 3,
37001 České Budějovice

Adresa: $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}$, c/o FI MU, Botanická 68a, 602 00 Brno

fax: 05–412 125 68

e-mail: cstug@cstug.cz

Zřízené poštovní aliasy sdružení $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}$:
bulletin@cstug.cz, zpravodaj@cstug.cz
korespondence ohledně Zpravodaje sdružení

board@cstug.cz

korespondence členům výboru

cstug@cstug.cz, president@cstug.cz

korespondence předsedovi sdružení

cstug-members@cstug.cz

korespondence členům sdružení

cstug-faq@cstug.cz

řešené otázky s odpověďmi navrhované k zařazení do dokumentu

CSFAQ

secretary@cstug.cz, orders@cstug.cz

korespondence administrativní síle sdružení, objednávky CD-ROM

bookorders@cstug.cz

objednávky tištěné T_EXové literatury na dobírku

ftp server sdružení:

ftp://ftp.cstug.cz/

www server sdružení:

http://www.cstug.cz/

Podávání novinových zásilek povoleno Oblastní správou pošt

v Českých Budějovicích, j. zn. P 3.202/94 ze dne 19. července 1994