Úvaha o fontech v CSTEXu

Při rozhovorech s nejrůznějšími příznivci TEXu o nové instalaci CSTEXu se často zavede řeč na otázku fontů, jejich názvů, kódování a mechanismů, s jakými CSTEX s těmito fonty pracuje. Myslím si, že to je záležitost velice důležitá a proto jsem se rozhodl uvést čtenáře do „zákulisí“, v němž nová koncepce fontů vznikala a zmínit mnohá pro a proti, se kterými jsme se potýkali při rozhodování o tom, jak to udělat.

Sám nepatřím mezi ty tvůrce CSTEXu, kteří v otázce fontů prosazovali jednoznačně jednu určitou myšlenku – většinou jsem se při těchto diskusích zdržel hlasování, protože jsem se nepovažoval v této věci za odborníka, který má znalosti o tom, jak se to dělá v jiných zemích. Nevím tedy, co je v této oblasti nejlepší. Tím samozřejmě nechci ze sebe smést veškerou vinu za nedostatky, které CS fonty mají. Na druhé straně si myslím, že bych mohl právě proto podat možná nejobjektivnější pohled na věc. Přitom asi pohled dostatečně zasvěcený, protože jsem „byl při tom“, tj. zúčastnil jsem se všech schůzek, na nichž nová instalace vznikala.

Původní verze

Mnohého čtenáře zajímají především změny, které s sebou přechod z původní instalace na novou přináší. Podívejme se na to podrobněji z technického hlediska. Začneme tím, jak to bylo zařízeno v původním CSTEXu. Pomineme úplně pravěkou verzi, kde

```
jedin\'y mo\v{z}n\'y zp\accent23usob,
jak n\v{e}co napsat \v{c}esky
```

vypadal takto. Prvním krůčkem k pohodlnějšímu psaní bylo vytvoření preprocesoru, který text napsaný například v kódování Kamenických konvertoval do textu obsahujícího příslušné `\accent23` a podobné klenoty a takto předpracovaný soubor se teprve nabízel ke čtení TEXem. Ani toto řešení nebylo pro národní jazyk dostačující, protože neumožňovalo implementovat vzory dělení slov.

Zastavme se až u osmibitové verze TEXu a u docela nedávné verze CSTEXu, která byla ještě před rokem šířená naším sdružením a která už velmi dobře umožňovala pořizovat české a slovenské dokumenty. Mattesův emTEX je schopen pomocí tzv. `tcp` tabulek řídit činnost preprocesoru, který je přímo zabudovaný do TEXu. Tyto tabulky se načítají jednou provždy při vytváření formátů. Byly vytvořeny tak, aby byl emTEX schopen číst například zdrojový text v „Kameničtině" a přitom vnitřně pracoval v poněkud jiném kódování – v případě původní verze CSTEXu šlo o kódování „skoro" podle Corku (o tomto kódování se rozepíšu za chvíli, protože kolem něho se to všechno točí). Tím stará verze vytvořila soubor `dvi`, který v sobě obsahoval odvolávky na akcentovaná písmena (říkejme mu pracovně osmibitový `dvi` soubor). Přitom tyto odvolávky neměly svůj protějšek v bitmapách (`pk`), takže se uvedený soubor `dvi` nedal použít přímo pro zpracování ovladači. Proto Oldřich Ulrych vytvořil konverzní rutinu `dvi2dvi`, která z osmibitového `dvi` souboru udělala jiný a delší soubor stejného jména, kde všechny odkazy na akcentovaná písmena byly nahrazeny řadou DVI-příkazů stejných, jako kdyby byl použit příkaz `\accent`. Tato rutina byla zařazena do dávky hned za volání TEXu a „průměrný" uživatel o její činnosti nic nemusel vědět. Výsledkem byl nakonec sedmibitový `dvi` soubor, který se dal zpracovat jakýmikoli ovladači, pracujícími s Computer Modern fonty. To byla veliká přednost z hlediska kompatibility.

Toto řešení mělo ale i nedostatky. Především háček v CM fontech jako samostatný znak, který se příkazem `\accent` posazuje nad písmena, vůbec neodpovídá českým typografickým představám a navíc jej TEX umisťuje na geometrickou osu písmene a nikoli podle „výtvarného citu". Také zde nebyla dodržena Knuthova zásada o neměnnosti CM fontů. Soubory `tfm` se totiž jmenovaly CM a přitom byly osmibitové – tedy TEX byl schopen například font zavedený jako

```
\font\titul=cmbx10 scaled\magstep4
```

zpracovat i s háčky a čárkami! Pro uživatele AMSTEXu a plainu, kde se bez příkazu `\font` nelze obejít, to bylo pohodlné řešení. Ovšem tito uživatelé si možná neuvědomovali, že pak jejich zdrojové texty nejsou mezinárodně přenositelné, protože standardní instalace TEXu s CM fonty má správně na výskyt akcentovaného znaku v takto zavedeném fontu reagovat v `logu` zprávou

```
Missing character .. in font..
```

a znak zcela ignorovat. Je sice pravda, že české texty asi nikdo nebude chtít mít mezinárodně přenositelné, ale nezasahovat do CM fontů je požadavek samotného autora TEXu a CM fontů a má k tomu asi dobré důvody. Ze staré verze si tedy uživatelé osvojili určitý zlozvyk v používání příkazu `\font`, jehož odnaučení bude činit značné potíže. Já sám patřím mezi takové uživatele.

Je třeba si uvědomit, že tato věc *vůbec nesouvisí se zvoleným typem kódování fontu*, takže kritici použitého kódování, kteří argumentují, že není možno použít příkaz `\font` „přirozeným způsobem“, jsou vedle jak ta jedle.

Mechanismus práce s fonty v nové verzi

Uveďme nejprve základní cíle, které byly kladeny na nově vytvořenou instalaci. Jsou tři:

- 100% kompatibilita s CM světem
- Zařadit osmibitové bitmapy fontů
- Nemít bitmapu žádné abecedy zbytečně dvakrát, tj. šetřit diskovou kapacitu

K dispozici byly osmibitové fonty Petra Nováka, které vycházely z CM fontů, ale navíc obsahovaly další znaky odpovídající písmenům s pevnými akcenty, používanými v české a slovenské sazbě. METAFONTové zdrojové texty těchto fontů dal Petr Novák plně k dispozici CSTUGu a to byl vlastně hlavní impuls k vytvoření naší nové instalace. Než se tyto fonty dostaly do nového balíku, byly výrazně upraveny Láďou Lhotkou a především Karlem Horákem.

Osmibitové fonty jsme nazvali shodně jako CM fonty, ovšem s tou výjimkou, že místo prvních dvou písmen `cm` jsou použita písmena `cs`. V instalaci jsou na úrovni metrik vedle sebe obě skupiny fontů, jak CM, tak CS. Najdete zde tedy jak soubor `cmr10.tfm`, tak `csr10.tfm`. Pouze pro matematické fonty alternativa CS neexistuje. CM metriky jsou nyní *originální* Knuthovy metriky nikterak neupravované. CS metriky obsahují i akcentované znaky a přitom v „dolní“ polovině tabulky (to je ta část s kódy menšími než 128) se CS fonty shodují s CM fonty.

Formát, který se automaticky generuje v CSTEXu při inicializaci má předponu CS, tedy např. `csplain, cslatex` apod. Při inicializaci `csplainu` a `csamstexu` je na chvíli předefinován primitivní příkaz `\font` (viz soubory `csplain.ini`, `csamstex.ini` a `csfonts.tex`). Tím pádem všechny formátem definované příkazy pro práci s fonty (např. `\bf`, `\it` apod.) automaticky pracují s fonty CS místo CM fontů. Protože na konci inicializace je příkazu `\font` vrácen původní význam (nedělalo by to dobrotu, kdyby tomu tak nebylo), je nutno pro zavedení uživatelského fontu použít *výslovně* CS font, tedy například

```
\font\titul=csbx10 scaled\magstep4
```

Formáty LaTeXu a AMSLaTeXu jsou řešeny trochu jinak. Zde není příkaz `\font` vůbec v průběhu inicializace předefinován, ale jsou načteny pozměněné definiční soubory pro fonty (viz soubory `makefmt.bat`, `lfontscs.tex`, `fontdef.cs` a další). Výsledný efekt je stejný: napíše-li uživatel LaTeXu třeba příkaz `\huge\bf`, zavede se a použije se místo fontu CM font CS. Takový uživatel se nemusí o nic starat, protože většinou nemusí pracovat explicite s příkazem `\font`.

Výsledek po zpracování TeXem tedy obsahuje v `dvi` souboru odkazy na akcentovaná písmena ve fontech s označením CS. Ovladače nyní pracují s osmibitovými bitmapami CS fontů a nemají žádný problém. Nepoužívá se žádný postprocesor.

Co se stane, zpracováváme-li CS formátem anglický dokument? Pokud v něm je použit příkaz `\font`, pak určitě se jím zavádí CM font a nikoli CS. Ve výsledném `dvi` souboru pak máme jakýsi „mix“. Texty zpracované implicitními příkazy z formátu (`\bf`, `\rm` atd.) jsou vysázeny v CS fontech a explicite definované příkazy způsobí sazbu v CM fontech. Nyní záleží na tom, co s takovým `dvi` souborem chceme dělat. Zpracujeme-li jej na naší instalaci, pak ovladače budou číst pouze osmibitové bitmapy CS a použijí z nich jen dolní polovinu tabulky. Narazí li na požadavek sazby v CM fontu, ovladač místo toho načte odpovídající CS font, jak praví substituční tabulka (viz `subst.drv`). Takže se zpracováním pomocí Mattesových ovladačů znovu nejsou žádné problémy. Navíc nemáme bitmapu žádné abecedy v instalaci zbytečně dvakrát.

Chceme-li naše „mixované“ `dvi` poslat do ciziny, musíme jej nejprve transformovat programem `dviout`, který zkontroluje, zda v celém dokumentu není jediný odkaz na akcentované písmeno a přejmenuje v `dvi` souboru všechny CS fonty na odpovídající CM fonty a upraví kontrolní součty. Takto zpracovaný `dvi` soubor je shodný se souborem, který by mohl být teoreticky vytvořen jinde standardním formátem pracujícím pouze s CM fonty. Nikdo nepozná, že jsme pro vytvoření tohoto souboru použili náš lokální formát s předponou CS. Tím je dosažena 100% kompatibilita s CM světem.

Druhá věc, kterou může chtít uživatel udělat s „mixovaným“ `dvi` souborem je jeho transformace do PostScriptového popisu stránek pomocí programu `dvips`. Tento program v naší instalaci také pracuje pouze s CS bitmapami, ale bohužel nemá schopnost načítat substituční tabulky, jak tomu bylo u emTeX-ovladačů. Proto je do dávky zařazen před volání programu `dvips` konvertor `dviout`, který tentokrát pracuje obráceně: všechny CM názvy v `dvi` souboru přepíše CS názvy (viz soubor `others.bat`).

Je vidět, že uvedenými mechanismy jsme dosáhli tří vytýčených cílů, které jsou zmíněny na začátku tohoto odstavce.

Kódování fontů

Všimněte si, že jsem se zatím nezmínil o kódování CS fontů. Chtěl jsem, aby bylo patrné, že všechny výše zmíněné problémy jsou *nezávislé* na volbě kódování horní poloviny tabulky, takže k tomu, abychom splnili zmíněné tři cíle, jsme mohli zvolit jakékoli kódování fontů, například Pišvejcovo kódování. Na druhé straně žádné super skvělé kódování by žádný výše zmíněný problém neumožnilo řešit elegantněji.

Připomeňme si, že *vnitřní* kódování TeXu (a tedy kódování fontů) není v emTeXu závislé na *vstupním* kódování zdrojových textů, protože mezi těmito dvěma světy stojí `tpc` tabulka preprocesoru, který je zabudovaný v emTeXu. Podle této tabulky se tedy konvertuje kódování vstupního textu (Kameničtí, PC Latin2, Koi8 či jiné) do předem daného vnitřního kódování. V instalaci CSTeXu umožňujeme při inicializaci formátu vybrat ze tří `tcp` tabulek, které odpovídají třem nejběžnějším vstupním kódováním na PC strojích. Není přitom problém vyrobit `tcp` tabulku pro jiná kódování. Po zpracování preprocesorem se tedy stává *vstupní* kódování nepodstatným pro naše další úvahy, takže se jím dále nebudeme zabývat.

Původní Novákovy fonty byly v Koi8. Je zřejmé, že toto kódování není příliš perspektivní a proto je také vyloučíme z našich úvah. Navíc Láďa s Karlem zařadili do METAFONTových textů jakési „meta-kódování“, takže změna rozložení znaků je záležitostí celkem komfortního editování *jediného* souboru.

Prvním kandidátem na vhodné vnitřní kódování TeXu by se mohlo jevit kódování podle Corku. Tento standard byl smluven na schůzce představitelů uživatelů TEXu pro latinkou píšící země v Evropě. Na základě tohoto kódování vznikly tzv. DC fonty, které nabízejí sjednocení toho, co potřebují Evropské země. Najdete tam třeba å, ç, ù, Ê nebo ñ. Samozřejmě tam najdete i akcentovaná písmena, používaná u nás.

Důležitým a nezanedbatelným argumentem pro standard podle Corku je skutečnost, že se podle tohoto standardu a na základě existence DC fontů začínají šířit nejrůznější makra, která s těmito fonty pracují. Taky vzory dělení slov pro evropské jazyky se zřejmě budou šířit v kódování Cork. Dále je třeba si uvědomit, že nelze míchat různá kódování v TEXu. Tj. že bych třeba přepínal mezi DC fontem a CS fontem a přitom by se samo přepnulo i kódování. To lze řešit jenom virtuálním popisem fontu. Vnitřní kódování TEXu je jediné. Poznamenejme ale, že přepínání mezi CS a DC fonty je dosti nesmyslný požadavek, protože z hlediska sazby českých nebo slovenských textů oba fonty nabízejí stejné možnosti. DC fonty mají znatelně horší typografickou kvalitu, ale zato obsahují některé „zahraniční“ symboly, které ovšem v případě potřeby lze poskládat i z diakritiky CM fontu (viz ñ).

Bohužel standard z Corku má jednu podstatnou nevýhodu: diktuje, jak má vypadat rozložení znaků v dolní polovině tabulky. Přitom se na mnohých místech liší od CM standardu! Například tam, kde v CM fontu najdeme řecká písmena (pozice nula, jedna, dvě, ...) jsou v DC fontech umístěna různá interpunkční znaménka.

Pokud bychom tedy chtěli použít „čistý Cork“ (jako je tomu v DC fontech), museli bychom v CS fontech zasáhnout do dolní poloviny tabulky, čímž ovšem nelze splnit požadavek o nejvýše jedné bitmapě pro každou abecedu. Museli bychom totiž mít vedle sebe i originální CM fonty už třeba jen kvůli tomu, abychom nepřišli o řecká písmena. Vyznavači DC fontů si tedy plní disk nejen neužitečnými znaky, které skoro nikdy nepoužijí a které se dají poskládat pomocí diakritiky z CM fontu, ale navíc musí mít CM fonty extra zvlášť. Tento nápor na diskovou kapacitu průměrného uživatele, který požádá o instalační diskety CSTEXu naše sdružení, jsme nechtěli připustit.

Mohli jsme tedy zůstat u „polovičního Corku“, tj. použít smíšené kódování Cork-CM tak, jako tomu bylo i u předchozí verze CSTEXu. Toto řešení taky zvolili naši polští kolegové. Zde bychom se ale museli samostatně rozhodnout, kam umístit některá interpunkční znaménka, která podle Corku kolidují s CM. Taky bychom asi neobsadili všechny pozice ve fontu, ale jen ty, které obsahují znaky používané v české a slovenské sazbě. Tento přístup mohl být jistým kompromisem, který by ale nakonec stejně nic neřešil. Zřejmě by toto poloviční řešení ani neumožnilo snadno použít makra vyvinutá pro DC fonty. Vyhovovalo by pouze uživatelům, kteří si nějak vylepšili původní instalaci, odstranili postprocesor `dvi2dvi` a na tomto starém smíšeném kódování začali dále stavět (např. si sestavili virtuální popisy k méně dostupným fontům, nebo si vytvořili některá makra závislá na kódování).

Do našich úvah vstoupil další argument, který se jmenuje ISO Latin2. Je to obecná norma pro „zpracování dat, textové aplikace a výměnu informací v albánštině, češtině, angličtině, němčině, maďarštině, polštině, rumunštině, srbochorvatštině, slovenštině a slovinštině“. Norma ISO má vekou váhu zvláště mezi počítačovými odborníky, kteří se neomezili jen na hračičky typu DOS a WINDOWS. Například v UNIXovém světě je jedinou alternativou (tam nepanuje taková džungle, jako na PC). Ještě informace pro ty, co něco vědí o kódování Latin2 na PC: ISO je něco jiného. Ačkoli jde spíš o definici rozšíření ASCII kódu pro horní polovinu tabulky než o kódování fontu (tj. nehledejte tam ligatury, samostatné akcenty apod.), ukázalo se, že pokud se přizpůsobíme tomuto kódování, mají UNIXoví administrátoři ulehčenou práci s instalací češtiny v TEXu. Jde totiž o to, že standardní Knuthův TEX, který je možné instalovat pod UNIXem není emTEX. Tedy není vybaven zabudovaným preprocesorem a mechanismem `tpc` tabulek. Tím pádem musí být vstupní kódování textu shodné s vnitřním kódováním TEXu (pokud nechceme programovat filtry, které by z důvodu zpětného výstupu chyb na terminál a do `logu` musely být oboustranné). Jak už bylo řečeno, ISO Latin2 je přitom standard pro kódování českých textů na výkonných systémech a zdá se, že zde například bratři Kameničtí neprorazí. Proč tedy neudělat CS fonty v tomto kódování? UNIXoví administrátoři nebudou muset programovat ani filtry, ani nebudou muset sestavovat virtuální popisy fontů. Stačí jim, když použijí naše CS fonty.

Tak se nakonec stalo, že CS fonty jsou v horní polovině tabulky podmnožinou ISO Latin2. Jsou totiž vybrány jen některé pozice, které odpovídají písmenům z češtiny a slovenštiny. Jedná se zhruba o padesát pozic; zbylých 80 zůstalo neobsazeno. Z mého laického pohledu je toto řešení asi stejně dobré i špatné, jako použití „polovičního Corku“. Jsou lidé, kteří se přiklánějí spíš k jedné či druhé variantě, ale zatím žádný názor nepřevažuje. To zřejmě ukáže až čas. Rozhodně ale CS fonty jsou tímto rozhodnutím jednou provždy vázané k tomuto kódování a instalace $\mathcal{CS}$TEXu je na nich nyní založena. Pokud se ukáže někdy v budoucnu vhodné použít jiné kódování, bude se muset změnit název fontů. Bude-li toto zaručeno, pak může u nás vzniknout solidní přenositelnost české sazby na úrovni `dvi` souborů. Nezdá se mi nereálné, aby například některé firmy vlastnící osvitovou jednotku nabízely osvit z `dvi` souboru. METAFONTování bitmap by pak mohla řešit firma sama, protože pouze ona přesně ví, jaké rozlišení je třeba použít. Zákazník by se o to nemusel starat. Pouze by muselo být řečeno, v jakém standardu je `dvi` soubor připraven. Zda se jedná o CM, CS nebo DC fonty. Proto upozorňuji všechny rýpaly, kteří chtějí měnit METAFONTové zdroje CS fontů, že ačkoli je změna rozložení velmi snadná, nedělejte to. Rozložení fontu, jehož výpis byl ostatně přiložen ke každé instalační krabici $\mathcal{CS}$TEXu, se měnit nesmí.

Přiznáváme, že nové řešení není zdaleka ideální. Chtěli jsme jen ukázat, že v tuto chvíli zřejmě žádné řešení nebude ideální. Kdyby se pánové v Corku dohodli jen trochu rozumněji (nepožadovali změnu v rozložení dolní poloviny tabulky), vypadala by dnes situace podstatně jinak. Z dnešního hlediska se jeví, že tehdy mohlo být výhodnější, kdyby se tito pánové vůbec nedohodli. Vytvořit rozumný standard lze totiž pouze v případě, že zde nefiguruje žádný špatný standard, na němž už část TEXovské obce začala stavět.

Co dělat?

Závěrem bych chtěl upozornit na některé bolístky nového $\mathcal{CS}$TEXu a při té příležitosti nabídnout aktivním a šikovným kolegům možnost se programátorsky realizovat.

První věc se může zdát celkem jednoduchá, ale zdání klame. Jde o makro `csfonts.tex`, které redefinuje primitiv `\font` tak, že pokud napíšeme

```
\font\muj=cmcokoli
```

bude to interpretováno jako

```
\font\muj=cscokoli.
```

přičemž tato změna významu proběhne pouze tehdy, jestliže `cmcokoli` není matematický font. V tomto článku jsem se zmínil, že jsem takové makro použil při generování CS formátů, ale že jej nedoporučuji používat globálně i v dokumentu. Znamená to, že makro není dokonalé. Znám tři chyby zmíněného makra, které způsobují, že nelze ponechat redefinovaný příkaz `\font` bez možných kolizí. Za prvé, délka názvu fontu se musí skládat aspoň ze čtyř písmen, jinak makro způsobí chybu. Za druhé, primitiv `\font` nevyžaduje povinné použití znaku `=` před názvem fontu, zatímco redefinovaný `\font` ano. Za třetí, primitiv `\font` má druhotný význam v konstrukci `\the\font`, přičemž při použití makra tento význam ztrácí. Šikovný makro-programátor by možná mohl tyto problémy vyřešit. Mám ale dojem, že to je natolik komplikované, že to nestojí za to a že se budeme muset všichni zlozvyk s používáním příkazu `\font` odnaučit. Nebo snad někoho napadá nějaké snadné a elegantní řešení, které by nebylo v rozporu s požadavkem autora CM fontů?

Další věcí je problém přenosu národního textu na zařízení, které není vybaveno CS fonty. Určitě by mnohý uživatel rád využil možnosti poslat třeba český text do zahraničí v souboru `dvi` v CM standardu, tj. ve formátu zpracovatelném na jakékoli instalaci TEXu ovšem za cenu nižší typografické kvality provedení háčků a čárek. To totiž v předchozí verzi CSTEXu šlo. Tam byla tato nižší typografická kvalita háčků jedinou možnou alternativou. V nové verzi nelze zatím tohoto návratu k CM dosáhnout. Dá se sice před zpracováním TEXem použít program `cstocs`, který ztransformuje všechny výskyty akcentovaných znaků do kontrolních TEX-sekvencí, ale pak přestane fungovat dělící algoritmus TEXu pro český (případně slovenský) jazyk! Použití `cstocs` je tedy možné (a nutné) pouze v případě anglického dokumentu, kde figuruje jen několik málo českých slov (například jména autorů, citace apod.).

Co tedy udělat pro možnost zasílat plné české/slovenské texty v `dvi` souboru v CM standardu do ciziny? Zdá se mi poněkud velký luxus kvůli tomu uchovávat starou verzi CSTEXu, která navíc pracuje naprosto odlišným způsobem. Chtělo by to pro tyto účely udělat něco podobného, jako byl v původní verzi konvertor `dvi2dvi`. Olin se ovšem k tomuto programu už nechce vracet. Jeho verze byla totiž víceméně dost komplikovaná. Měla ve svém `exe` kódu pevně stanovenou množinu fontů, se kterou uměla pracovat. Tabulku akcí, co má s daným písmenem daného fontu dělat, se program `dvi2dvi` „naučil“ od TEXu v okamžiku své kompilace (v Turbo Pascalu). Matně si vzpomínám, že tato kompilace byla natolik komplikovaná (v různých fázích kompilace bylo nutno vyvolat TEX na různé soubory a pak pokračovat), že jsem kdysi snahu naučit program `dvi2dvi` pracovat s novým fontem nakonec vzdal. Nabyl jsem dojmu, že to může pochopit jen autor programu.

Ideální by bylo, kdyby existoval takový konvertor, který by obsahoval některé algoritmy TEXu přímo v sobě a nemusel se pro každý nový font znova učit, co má dělat. Jedná se většinou o algoritmy, které se spustí primitivem `\accent`. Našel by se někdo, kdo by se odvážil do toho pustit?

Další věc, která by byla obecně užitečná, je podpora nejběžnějších PostScriptových fontů v českém jazyce. Určitě by nebylo od věci, kdyby CSTUG mohl nabízet virtuální popisy těchto fontů (`VPL` – virtual property list) k veřejnému použití v TEXu. Je sice pravda, že se jedná o komerční fonty, ale ty nejběžnější z nich jsou součástí vpodstatě každého PostScriptového zařízení (včetně WINDOWS) a požadavky na jejich použití v TEXu budou určitě stále častější. Navíc tyto popisy, které se nutně opírají o vnitřní kódování TEXu, upevní pozici ISO Latin2 v našich luzích.

Nakonec jedna programátorsky možná atraktivní záležitost, ale nikterak nesouvisející s fonty. Možná už také existuje v „public domain", jen o tom nevím. Jde o to, že by se CSTEX stal atraktivnějším pro začínající uživatele, kdyby byl vybaven inteligentním hypertextovým helpem. Představme si, že máme konfigurovatelný editor, kde lze nakonfigurovat funkční klávesu tak, že se na její stisknutí vyvolá příkaz DOSu a navíc se do tohoto příkazu obkreslí slovo, na němž stojí kurzor. Například QEdit to umí a CSEd pravděpodobně taky. Jde tedy o to získat program, který nabízí po vyvolání z řádky DOSu nápovědu ke slovu, které je na řádce DOSu uvedené jako parametr. Ideální by bylo, kdyby byl program schopen najít i slova příbuzná a byl vybaven silnými hypertextovými možnostmi. Samozřejmostí je čtený text v komprimované podobě a s rychlým přístupem, protože takový LATEXbook či TEXbook zabere jistě dosti místa i ve formě nápověd. Taky možnost snadného vytváření a doplňování nápověd je nutná.

Rozhodně by se našly i další nevyřešené problémy, co by se dalo dělat. Máte-li do něčeho chuť, doporučuji nejprve kontaktovat výbor CSTUGu. Podmínkou by totiž v každém případě měla být možnost zařazení softwaru do „public domain". Určitě nebude CSTUG kupovat komerční výrobky s komerčními chybami a za komerční ceny. Taková věc je v příkrém rozporu s myšlenkou samotného TEXu.