

$\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ – historie, současný stav a budoucnost

Petr Olšák

Elektrotechnická fakulta ČVUT, Praha

Email: olsak@math.feld.cvut.cz

Abstrakt: $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ je podpora češtiny a slovenštiny v $\mathcal{T}\mathcal{E}\mathcal{X}$ u a sestává ze tří základních pilířů: $\mathcal{C}\mathcal{S}$ fonty, $\mathcal{C}\mathcal{S}$ plain a $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. V přednášce bych se pozastavil podrobněji u každého z nich, zavzpomínal na to, co se kolem vývoje jednotlivých balíčků odehrálo, upozornil na současné novinky a pokusil se odhadnout budoucí vývoj.

Počátky $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u

Záznam o počátcích počestění $\mathcal{T}\mathcal{E}\mathcal{X}$ u lze najít v článku Ladislava Lhotky [6]. Autor zde píše, že se na podzim roku 1988 sešel s Petrem Novákem a dohodli se, že udělají české osmibitové fonty (v $\mathcal{M}\mathcal{E}\mathcal{T}\mathcal{A}\mathcal{F}\mathcal{O}\mathcal{N}\mathcal{T}\mathcal{U}$) a vzory dělení slov. Nejprve to vypadalo tak, že Láďa Lhotka udělá fonty a Petr Novák vzory dělení, ale pak to dopadlo přesně naopak.

Doslechl jsem se také, že v době, kdy ještě neexistoval $\mathcal{T}\mathcal{E}\mathcal{X}$ ve verzi 3 (do roku 1989), Petr Novák vymyslel nějaké finty, při níž bylo možno dosáhnout českých akcentovaných znaků pomocí ligatur v $\mathcal{T}\mathcal{E}\mathcal{X}$ ových metrikách.

Po vzniku $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}$ u (konec roku 1990) se iniciativy nad rozšiřováním a zvelebováním české a slovenské podpory pro $\mathcal{T}\mathcal{E}\mathcal{X}$ ujal Olin Ulrych. V té době se tento projekt začal nazývat $\mathcal{C}\mathcal{S}$ $\mathcal{T}\mathcal{E}\mathcal{X}$, později $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$. Olin zapracoval Lhotkovy vzory dělení a Novákovy fonty, přidal nějaká udělátka na úrovni preprocesoru a programy manipulující s $\mathcal{d}\mathcal{v}\mathcal{i}$. První distribuce byly postaveny na $\mathcal{P}\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u, pak $\mathcal{S}\mathcal{B}\mathcal{T}\mathcal{E}\mathcal{X}$ u. V roce 1992 přešel $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ kompletně na $\mathcal{e}\mathcal{m}\mathcal{T}\mathcal{E}\mathcal{X}$ ovou distribuci pro DOS. Já jsem tehdy udělal program pro konfiguraci nabídek, který spolupracoval s DOSovými dávkami [7]. Členům sdružení jsme v letech 1993 a 1994 začali rozesílat disketové balíčky s $\mathcal{e}\mathcal{m}\mathcal{T}\mathcal{E}\mathcal{X}$ em a s českou a slovenskou podporou – $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$.

V té době většina členů používala DOS, takže $\mathcal{e}\mathcal{m}\mathcal{T}\mathcal{E}\mathcal{X}$ pro DOS byl pro ně vyhovující. V dnešní době uživatelé pracují s nejrůznějšími operačními systémy a s různými distribucemi $\mathcal{T}\mathcal{E}\mathcal{X}$ u pro tyto systémy. Z toho důvodu se v tomto textu přidržíme jen užšího významu slova $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$, definovaného takto:

$\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ je sada $\mathcal{T}\mathcal{E}\mathcal{X}$ ových maker, fontů, vzorů dělení slov a doplňujícího software pro podporu české a slovenské sazby v $\mathcal{T}\mathcal{E}\mathcal{X}$ u. Je vytvořen tak, aby mohl být použit na libovolné $\mathcal{T}\mathcal{E}\mathcal{X}$ ové distribuci na libovolném operačním systému.

V roce 1996 jsem napsal *Pár poznámek k novému $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u* [2]. Tento dokument má z dnešního pohledu poněkud zastaralý název, protože software z roku 1996 dnes sotva můžeme považovat za nový. Nicméně zde čtenář může najít podrobnou historickou poznámku mapující léta 1992–1996. Už v tomto textu z roku

1996 se snažím vymezit pojem $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ poněkud širěji, než jen $\text{em}\mathcal{T}\mathcal{E}\mathcal{X}$ ová distribuce obohacená o podporu češtiny a slovenštiny. Vymezil jsem ho ovšem až moc široko: pořád jsem za $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ považoval kompletní fungující distribuci $\mathcal{T}\mathcal{E}\mathcal{X}$ u s českou a slovenskou podporou, i když pro blíže nespecifikovaný operační systém.

Na základě vymezení pojmu $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ z roku 1996 začaly vznikat ve ftp adresáři $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u [1] podadresáře označené podle operačního systému či distribuce $\mathcal{T}\mathcal{E}\mathcal{X}$ u (např. `emtex`, `web2c`), které měly ambici nabídnout kompletní $\mathcal{T}\mathcal{E}\mathcal{X}$ ovou distribuci včetně české a slovenské podpory. Bohužel, udržovat všechny nabízené distribuce v aktuálním stavu se nedalo při jednom muži zvládnout, a proto obsahy těchto adresářů velmi rychle zastarávaly.

V roce 2002 jsem se při psaní nového *Manuálu k $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u* [3] znovu zamyslel nad obsahem pojmu $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ a pokusil jsem se jej vymezit poněkud opatrněji – stejně, jak je uvedeno zde. Značná část textu v tomto příspěvku pro konferenci S_TT 2002 je převzata právě z [3].

Nově vymezený pojem $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ se zhruba shoduje s tzv. *jádrem $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u*, což je slovní spojení použité už v roce 1996 v [2]. Jádro $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u (nově tedy $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$) nabízím správcům jednotlivých $\mathcal{T}\mathcal{E}\mathcal{X}$ ových distribucí k zařazení. Pro tyto účely jsem vytvořil nově na ftp adresáři $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u adresář `base`, který obsahuje tary $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u k zařazení do různých $\mathcal{T}\mathcal{E}\mathcal{X}$ ových distribucí. Vzdávám se tedy původní myšlenky udržovat kopie $\mathcal{T}\mathcal{E}\mathcal{X}$ ových distribucí na ftp adresáři $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u v aktuálním stavu.

$\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ i v tomto užším smyslu dělím na část povinnou (required) a nepovinnou (recommended). Povinná část musí být kompletně instalovaná v $\mathcal{T}\mathcal{E}\mathcal{X}$ ové distribuci, aby se o ní mohlo říci, že „obsahuje $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ “. Při vzájemné výměně dokumentů „psaných v $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u“ je totiž otázka, zda distribuce obsahuje $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$, důležitá. Povinná část obsahuje $\mathcal{T}\mathcal{E}\mathcal{X}$ ová makra, vzory dělení a fonty, které nejsou závislé na operačním systému ani na distribuci $\mathcal{T}\mathcal{E}\mathcal{X}$ u. Dají se tedy relativně snadno implementovat do jakékoli $\mathcal{T}\mathcal{E}\mathcal{X}$ ové distribuce. Nepovinná část obsahuje další doprovodný software, jako například `csindex` (varianta programu `makeindex`) nebo program `vlna` na doplňování vlnek za předložkami.

Povinná část $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u sestává ze tří základních pilířů: $\mathcal{C}\mathcal{S}$ fonty, $\mathcal{C}\mathcal{S}$ plain a $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. $\mathcal{C}\mathcal{S}$ fonty jsou konzervativním rozšířením Knuthových Computer Modern fontů, $\mathcal{C}\mathcal{S}$ plain je konzervativním rozšířením Knuthova formátu plain a konečně $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ je jistou modifikací běžně používaného formátu $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. Do povinné části $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{E}\mathcal{X}$ u je ještě zahrnuta podpora použití základních 35 PostScriptových fontů v češtině a slovenštině prostřednictvím virtuálních fontů.

O historii, současném stavu a budoucnosti jednotlivých pilířů si povíme podrobněji v následujících kapitolách.

2 $\mathcal{C}\mathcal{S}$ fonty

Tvary akcentů $\mathcal{C}\mathcal{S}$ fontů byly vytvořeny a implementovány v jazyce METAFONTu Petrem Novákem ve spolupráci s českými výtvarníky někdy na začátku 90. let. Autor přenechal $\mathcal{C}\mathcal{S}$ fonty $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}$ u, který s nimi může libovolně nakládat.

METAFONTový kód byl pak v letech 1992–1993 dále upraven Karlem Horákem. Karel se inspiroval z METAFONTových zdrojů pro fonty vytvořené v Polsku. Zapracoval tam možnost nastavení kódování fontu a vytvořil makra umožňující mít všechny hlavní mf soubory se stejným dvouřádkovým obsahem.

Na schůzce tvůrců CTeXu na FEL v roce 1993 bylo rozhodnuto, že Cfonty budou mít kódování podle ISO 8859-2. Později, při implementaci CTeXu do UNIXových distribucí nepodporujících změny xord/xchr vektorů se ukázalo, že to bylo velmi prozíravé rozhodnutí.

V roce 1993 jsem převzal údržbu Cfontů do svých rukou. Udělal jsem jen velmi drobné změny. Poslední 28. 9. 1996. Pak jsem vývoj Cfontů zmrazil podobným způsobem, jako Knuth přestal měnit CM fonty. Prioritním požadavkem je, aby dokument opírající se o Cfonty byl od roku 1996 formátován naprosto stejně dnes i kdykoli v budoucnu. Aby byl tento požadavek splněn, není tedy možné zasáhnout do rozměrů znaků, kernů a ligaturních tabulek.

V roce 1996 jsem do Cfontů přidal virtuální fonty podporující náhradu Computer Modern fonty a naopak.

V roce 1998 se podařilo dohodnout s autorem teTeXu Thomassem Esserem, aby zařadil do své distribuce Cfonty a celý CTeX. Od této chvíle obsahují distribuce odvozené z teTeXu implicitně kompletní CTeX.

V roce 1998 jsem také pro potřeby výstupu do formátu PDF vytvořil variantu Cfontů, tentokrát ve formátu PostScript Type1. Vyšel jsem z BaKoMa Type1 implementace CM fontů a vytvořil jsem si program `t1accent` [4], který k písmenkům přidával akcenty podle vzorových PostScriptových tahů generovaných z původních Cfontů METAPOSTem. Na mnoha místech jsem byl nucen přistoupit k mikrotypografickým kompromisům – v drobnostech se kresby některých znaků Cfontů z Type1 liší od svých originálních METAFONTových protějšků. Proto jsem distribuci Type1 Cfontů označil jako „alpha“ a do komentáře jsem dal důrazné varování, že tyto fonty je možné používat na vlastní riziko. Z toho důvodu jsem také ponechal implicitní konfiguraci programu `dvips` tak, aby program používal léty osvědčený výstup z METAFONTu, tedy bitmapy formátu `pk`. Type1 Cfonty byly původně konfigurovány výhradně pro pdfTeX.

Rozhodnutí ponechat `dvips` pracovat implicitně s bitmapami naráželo na problémy. Neustále dokola se uživatelé ptali, jak je možné, že výstup z pdfTeXu je dobrý, ale při cestě `dvips – ps2pdf` dostávají roztřesená písmenka. Byl jsem uondán velmi častým odpovídáním na tuto otázku a vysvětlováním, jak použít `dvips` s jinou než výchozí konfigurací. Ani jsem se tedy nakonec nezlobil, když v roce 2001 autor teTeXu rozhodl, že bude Cfonty pro `dvips` implicitně konfigurovat ve verzi Type1. Asi ty mikrotypografické kompromisy ani tak moc nevadí, zatímco roztřesená písmenka v PDF a neznalost použití jiné než výchozí konfigurace `dvips` způsobovala oheň na střeše.

V současné době existují volně dostupné nástroje, jako například `textrace` opírající se o `autotrace`. Tyto nástroje umožní převést METAFONTový font do Type1 „obtahováním bitmap“ skoro automaticky. Vyzkoušel jsem to na Cfontech a s výsledkem jsem nebyl vůbec spokojen: výsledné `pfb` soubory byly asi pětkrát

větší než ty moje „ručně“ vyrobené a z BaKoma fontů odvozené. Proto jsem zatím alpha verzi Type1 formátu C_Sfontů z roku 1988 neopustil.

Do budoucna bych velmi rád do C_Sfontů přidal znak euro a paragraf. Taková změna by byla zpětně kompatibilní, takže bych se jí nebránil. METAFONTové zdroje pro paragraf ověřené na všech C_Sfontech už několik let mám, ale nezveřejnil jsem je. METAFONTové zdroje znaku euro by se snadno daly převzít z jiného METAFONTového fontu. Největší potíž je ovšem v tom, že s uvedením nové verze C_Sfontů dnes nestačí zveřejnit jen METAFONTové zdroje a metriky, ale je třeba mít okamžitě s tím konzistentní Type1 varianty fontů. Do manuální práce na nové verzi Type1 varianty C_Sfontů se mi ale moc nechce. Je to nevděčná a rozsáhlá práce: pfb souborů je v balíčku 57 a každý je třeba disassemblovat, v editoru přidat nové znaky a znovu převést na pfb. Přitom s automatickými nástroji, jak jsem uvedl před chvílí, nejsem spokojen.

3 Podpora základních 35 PostScriptových fontů

Tato podpora je zahrnuta do balíčku `cspfonts.tar.gz` a má méně bohatou historii než C_Sfonty. Balíček začal vznikat v září roku 1994. Tehdy jsem zjistil, že popisy kompozitů v AFM metrice pomocí řádků CC jsou správně převáděny programem `afm2tfm` na odpovídající kompozity ve vytvářeném virtuálním fontu. Problém byl jen v tom, že originální AFM metriky od Adobe neobsahovaly popisy všech kompozitů potřebných pro český a slovenský jazyk. Z toho důvodu jsem si vytvořil program `a2ac` [5], který na základě přehledné tabulky kompozity do AFM metrik doplnil a současně doplnil kerningové informace pro nově vytvářené znaky. Za použití tohoto programu pak vznikla sada metrik a virtuálních fontů s písmenem c na začátku (např. `cptmr`).

V roce 1996 pak uveřejnil pan Wagner nové metriky generované stejným způsobem, ovšem opravil několik estetických nedostatků a navíc metriky nazval podle doporučení Karla Berryho (8z a 8t na konci). Od té doby jsou v balíčku `cspfonts.tar.gz` obsaženy metriky pana Wagnera. Česká a slovenská abeceda je v těchto metrikách kódována stejně, jako v C_Sfontech, tj. podle ISO-8859-2. Právě tomuto kódování odpovídá koncovka 8z.

Konečně v roce 1999 jsem musel po konzultaci s Karlem Berryem metriky pro rodinu Courier přejmenovat z původního *8t na nynější *8u, protože názvy s 8t na konci nám kolidovaly s názvy stejných fontů v kódování podle Corku. To je zatím poslední změna v tomto balíčku.

Do budoucna zvažuji rozšířit balíček `cspfonts.tar` o další metriky a virtuální fonty *.8z k volně šířeným Type1 PostScriptovým fontům. První na řadě může být font Charter, kterým je vytvořen tento sborník. Neuvědomil jsem si, že se tento font stal volně šířeným – mám ho totiž už léta kopužený. Takže metriky kódování *8z už mám dávno hotové, jen zatím nebyly nikde zveřejněny.

4 C_Splain

C_Splain vznikl v roce 1992 jako jednoduché a minimální rozšíření Knuthova plainu používající C_Sfonty a akceptující 8bitový vstup. Jeho vytvoření bylo motivováno zařazením do emT_EXové distribuce, která se připravovala k rozesílání členům C_STUGu.

C_Splain se opíral a stále opírá o starší makra `hyphen.lan` a `plaina4.tex`, která už měl Olin Ulrych vytvořena dříve. C_Splain v době svého vzniku načítal české vzory dělení, která vytvořil Láďa Lhotka heuristicky bez použití slovníků a programu `patgen`. Já jsem pro C_Splain vytvořil `cspplain.ini` a makro `csfonts.tex` umožňující při generování formátu číst přímo Knuthovo makro `plain.tex`, a přitom natáhnout místo CMfontů C_Sfonty.

V roce 1995 byly vzory dělení slov Ládi Lhotky vyměněny za nové české vzory dělení od Pavla Ševečka, který na to použil slovník a `patgen`. V rámci své firmy tyto vzory dělení prodává komerčním firmám pro potřeby DTP programů, jako byly Ventura, Pagemaker nebo Quark. Dnes jsou tyto vzory dělení také například ve Wordu. Aby Pavel Ševeček odlišil volně šířené vzory dělení pro C_STUG od komerčně šířených, volně šířené vzory dělení mírně modifikoval. Tvrdí se, že běžný uživatel nepozná rozdíl v kvalitě vzorů dělení komerčních a volně šířených. Ševečkovy vzory dělení slov jsou výrazně kvalitnější než původní Lhotkovy, a proto jsme u těchto vzorů dělení zůstali.

Opuštěním Lhotkových vzorů dělení došlo k poslední změně v C_Splainu, která může způsobit zpětnou nekompatibilitu v českém dokumentu: tj. dokumenty vytvořené v C_Splainu před rokem 1995 mohly dopadnout jinak než dnes, protože některá slova mohla být rozdělena jinak. Od této doby je C_Splain fixován a stabilní podobně jako Knuthův plain.

Protože jsem autorem názvu C_Splain, souborů `cspplain.ini`, `csfonts.tex` a množství dokumentace k C_Splainu a protože jej od jeho vzniku udržuji, rozhodl jsem se přísně dbát na zpětnou kompatibilitu. Změny do C_Splainu dělám jen takové, které jsou opravdu nezbytné. To se stává jednou za několik let (viz historické poznámky v `cspplain.ini`). Změny dělám tak, že pouze přidám další nejnnutnější makra, ale stávající makra a jejich význam nechávám nezměněna. Uživatelům C_Splainu ručím, že jejich dokumenty napsané v C_Splainu a opírající se o neměnné fonty (např. C_Sfonty nebo base 35 PostScriptové fonty, metriky z C_ST_EXu), budou i v budoucnu C_Splainem formátovány naprosto stejně, jako dnes.

Abych mohl takovou záruku uživatelům poskytnout, není C_Splain zveřejněn pod GNU GPL, ale jedná se o licenci velmi podobnou Knuthově. Přesné znění licence je uvedeno na konci souboru `cspplain.ini`. Zhruba řečeno, jedná se o „patent na název“. C_Splain můžete svobodně distribuovat, používat a měnit, ale pokud jej změníte, nesmíte jej dále distribuovat pod názvem C_Splain. Změny v C_Splainu může dělat jen tzv. „současný administrátor C_ST_EXu“, což jsem zatím stále já. Pokud bych v budoucnu toto břímě někomu předal, pak určitě jen takovému člověku, který má na budoucnost C_Splainu stejný názor, jako já.

C_Splain i nadále považuji za minimální rozšíření Knuthova plainu a nikdy do něj nepřidám žádné složitější makro vylepšující uživatelský komfort (jako

například `eplain` nebo `OFS`). Zastávám názor, že uživatel `plainu` a `CSplainu` chce mít všechna makra pod svou vlastní kontrolou a raději si je udělá sám, než aby spoléhal na hotová, ale méně stabilní, řešení. Uživatel `plainu/CSplainu` si vytváří vlastní stále znovu používaná makra, která mu musí fungovat ve všech i budoucích verzích `CSplainu`. Proto se snažím `CSplain` pokud možno neměnit.

Lákavá je například změna definice uvozovek v `CSplainu` jednoduše na

```
\long\def\uv#1{\clqq#1\crqq}
```

aby fungoval automatický kerning s oběma stranami uvozovek. Tuto změnu ale nikdy v `CSplainu` neudělám, protože není zpětně kompatibilní se stávajícím řešením a v některých dokumentech by uvnitř `\uv` přestaly fungovat verbatim konstrukce. Raději budu psát do omrzení do dokumentace, že taková jednoduchá definice je asi lepší, než ta z `CSplainu`, a že si ji každý může zařadit do svých `maker`.

Změnu v `CSplainu` z <Feb. 2000> považuji za asi největší, kterou jsem byl ochoten udělat. Tehdy jsem zařadil do `CSplainu` alternativní kódování `Cork`, které samozřejmě není a nikdy nebude v `CSplainu` implicitní. Implicitní nadále zůstává kódování podle `ISO-8859-2`, tedy podle `CSfontů`. K přidání podpory `Corku` mě motivovala skutečnost, že se kolem mě pohybovalo mnoho uživatelů `plainu`, kteří rádi používají fonty v tomto alternativním kódování. Svým krokem jsem jim umožnil používat `CSplain`, takže si nemusejí vytvářet své vlastní formáty.

Jsem si vědom problému s touto změnou spojeném: do `emTeXu` (binárky `tex.exe` i `tex386.exe`) se ani po změně paměťových parametrů pomocí přepínačů nevejde pět vzorů dělení slov (angličtina, čeština v `ISO-8859-2` a v `Corku` a slovenština v `ISO-8859-2` a v `Corku`). V této distribuci je možno formát vygenerovat jen binárkou `htex386.exe`. Proto zůstává podpora kódování podle `Corku` v `CSplainu` jako nepovinná, tj. nelze očekávat, že ji všichni uživatelé `CSTeXu` budou mít. Také se dá tato podpora vypnout pomocí `\let\Cork=\relax` před zavedením souboru `csplain.ini`.

Možná už tušíte, proč jsem čekal až do roku 2000 s přidáním takového rozšíření. Usoudil jsem, že na prahu nového tisíciletí už snad `emTeXovou` distribuci moc lidí nepoužívá.

Pokud jde o budoucnost `CSplainu`, pak nepředpokládám žádné výrazné změny a vždy budu nekompromisně dodržovat zpětnou kompatibilitu. Jediná věc, která by mohla kompatibilitu ohrozit, je zavedení kvalitnějších slovenských vzorů dělení slov. Věřím, že slovenští kolegové raději přejdou na kvalitnější vzory dělení slov, než aby lpěli na absolutní zpětné kompatibilitě s ne příliš kvalitními vzory dělení. Otázka pouze zůstává, kdo nové slovenské vzory dělení udělá...

5 $\text{CS}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

$\text{CS}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ vytvořil zhruba v roce 1992 Jiří Zlatuška. Od něj pochází myšlenka načtení vzorů dělení stejného jazyka v různých kódováních a předefinování $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ového makra `\DeclareFontEncoding`. Na své soukromé implementaci `TeXu` tehdy provozoval mimo jiné fonty kódované v `KOI-8`, takže přepínání vnitřního

kódování L^AT_EXu si vlastně udělal pro svoje potřeby. Jiří Zlatuška je pravděpodobně také autorem maker `\splithyphens` a `\standardhyphens`, která zapínají a vypínají inteligenci znaku `-`. Veškerá makra napsal dobře dokumentovaná pro použití v systému `docstrip`. Svou práci zveřejnil pod licencí GPL mimo jiné podle jeho slov proto, že pokud to bude někoho zajímat, tak to může dále udržovat a zvelebovat podle svých vlastních představ. On sám se kvůli své zaneprázdněnosti v jiné oblasti tímto problémem později zřejmě nezabýval.

Po schůzce tvůrců CS₂TEXu v roce 1993 převzal starost o CS₂L^AT_EX podle dohody Zdeněk Wagner, který vytvořil definice kódování LL2. Vytvořil také pro CS₂L^AT_EX definiční soubory `fd` jednak pro CS₂fonty a jednak pro PostScriptové fonty (balíček `cspsfonts.tar.gz`). V té době byl ještě CS₂L^AT_EXu implementován pro verzi L^AT_EXu 2.09. V em₂TEXové distribuci CS₂TEXu je stále tato implementace obsažena (`cs1t209.zip` v adresáři `emtex`). Součástí dnešního CS₂TEXu (`cs2latex.tar.gz` v adresáři `base`) už podpora této staré verze L^AT_EXu není.

Ačkoli možnost načíst vzory dělení jednoho jazyka ve více kódováních dal CS₂L^AT_EXu už Jiří Zlatuška, byla tato možnost až do roku 1999 implicitně v souboru `hyphen.cfg` vypnuta a CS₂L^AT_EX bez editace tohoto souboru pracoval jen v kódování CS₂fontů. Důvod už jsem zmínil v kapitole o CS₂plainu: v em₂TEXové distribuci se pět vzorů dělení do paměti binárek `tex.exe` a `tex386.exe` prostě nevešlo.

Soubor `czech.sty` má asi podstatně delší historii než CS₂L^AT_EX. Pochází z dílny Olina Ulricha, který se zřejmě inspiroval podobným stylovým souborem pro německý jazyk. Olin rovněž vytvořil makra `\csprimeson` a `\csprimesoff`. Zdeněk Wagner pak převzal Olinův stylový soubor a upravil jej pro provoz v CS₂L^AT_EXu. Slovenskou část včetně vzorů dělení slov vytvořila Janka Chlebíková. Soubor `slovak.sty` je přesnou kopií souboru `czech.sty` s výjimkou slovensky specifických částí.

V duchu licence GPL převzal zhruba v roce 1997 údržbu CS₂L^AT_EXu Jaroslav Šnajdr. Udělal několik úprav stylových souborů `czech.sty` a `slovak.sty` včetně přechodu na novou definici `uvozovek`, uvnitř jejichž argumentu nefungují verbatim konstrukce. Tím kuriózně způsobil, že CS₂L^AT_EXem od této doby nejde bez chyb formátovat český překlad úvodu do L^AT_EXu, který je pod názvem balíčku `csuvodlat.tar.gz` součástí dokumentace CS₂TEXu. Je to názorná ukázka toho, co může způsobit změna kódu, která nerespektuje zpětnou kompatibilitu. Pan Šnajdr rovněž napsal `html` dokumentaci k CS₂L^AT_EXu, která popisuje instalaci CS₂L^AT_EXu ze zdrojových souborů použitím `docstripu`. Použijete-li ale balíček `cs2latex.tar.gz`, pak nemusíte `docstrip` aplikovat, protože vedle zdrojových souborů jsou tam už přítomny i všechny soubory, které vznikají po aplikaci `docstripu`.

Já osobně jsem o L^AT_EX a tím pádem CS₂L^AT_EX jevil od začátku malý zájem, protože osobně používám CS₂plain. V roce 1999 jsem nicméně přidal pár řádek maker do souboru `czyhyphen.tex` tak, aby byl použitelný v babelizovaném L^AT_EXu. Do té doby totiž tato větev L^AT_EXu používala Lhotkovy vzory dělení, zatímco v CS₂L^AT_EXu jsme už čtyři roky měli daleko kvalitnější Ševečkovy vzory dělení. Tyto novější vzory dělení jsou napsány za použití TEXových sekvencí, což je sice

nezávislé na kódování češtiny, ale balíček Babel to implicitně nedokáže strávit a očekává vzory dělení v kódování T1. Upravený soubor jsem nazval Babelovsky: `czhyph.tex`, zatímco v \LaTeX u zůstává původní soubor `czhyphen.tex`.

Na výborové schůzi v roce 1999 jsem dostal za úkol prověřit možnost spojení babelizovaného \LaTeX u s \LaTeX em. Neustálé dotazy začínajících uživatelů, kteří si pletou tyto dva \LaTeX y, nás utvrzují v tom, že by se pro sloučení mělo něco udělat.

Analyzoval jsem proto makra Babelu a udělal návrh na možné zapracování funkcionality \LaTeX u do Babelu. Domnívám se, že \LaTeX klidně může přestat existovat, ale babelizovaný \LaTeX musí bezpodmínečně převzít všechny vlastnosti \LaTeX u tak, aby dokumenty dříve zpracovávané \LaTeX em byly naprosto stejné a bez jediné úpravy zpracované novým babelizovaným \LaTeX em. Kvůli tomuto požadavku musí babelizovaný \LaTeX umět načítat vzory dělení stejného jazyka ve více kódováních, jako to nyní dělá \LaTeX . Dospěl jsem k závěru, že čistým řešením tohoto problému je jediné zásah do jádra \LaTeX u samotného, aby dokázal při změně kódování fontů přepnout automaticky i vzory dělení. Zlatuška kvůli tomu předefinoval makro jádra \LaTeX u `\DeclareFontEncoding`. Tato záplata, či jinak řečeno odmítnutí původního kódu tohoto makra, je na úrovni Babelu podle mého názoru velmi nečisté řešení. Skutečnost, že přepínání vzorů dělení při přepnutí kódování fontů \LaTeX ové jádro neřeší, považuji za chybu \LaTeX u. V roce 1999 jsem tedy požádal \LaTeX -team, aby zapracoval změnu v duchu Zlatuškovy návrhu do \LaTeX ového jádra. Můj návrh nebyl \LaTeX -teamem akceptován. Za těchto okolností nejsem schopen zapracovat funkcionalitu \LaTeX u do Babelu, protože to prostě nejde. Uživatelé \LaTeX u se budou muset nadále potýkat s tím, že jejich oblíbený formát trpí určitou schizofremií.

Společně se sloučením \LaTeX u s Babelem jsem připravoval zásadní revizi stylů `czech.sty` a `slovak.sty` – v podstatě jsem měl v úmyslu jejich totální přepsání. Tyto stylové soubory obsahují množství reliktních z dob minulých, plno zcela nepoužívaných větví ve složitém větvení pomocí `\if` a stávají se totálně nepřehlednými. Protože ale ke sloučení \LaTeX u s Babelem nakonec nedošlo, upustil jsem zatím od plánu pracovat na těch stylových souborech. Není ale vyloučeno, že k tomu dojde v budoucnosti. V takovém případě počítám s tím, že makra `\splithyphens` a `\standardhyphens` přesunu z formátu do stylového souboru, kam přirozeně patří. Dokumenty, které tato makra používají, a přitom nemají v záhlaví `\usepackage{czech}` ani `\usepackage{slovak}`, pak nebudou fungovat. Předpokládám, že takových dokumentů není mnoho, protože \LaTeX a stylové soubory jsou většinou používány současně.

Protože pan Šnajdr se přestal \LaTeX em zabývat, byl jsem nucen v roce 2002 zanést do stylových souborů jednu opravu podle požadavku pana Kubena osobně. Neznamená to ale, že bych se ujal iniciativy nad \LaTeX em. Jak jsem už vysvětlil, jsem ochoten převzít iniciativu jen tehdy, když bude \LaTeX ové jádro umět přepínat mezi různě kódovanými vzory dělení stejného jazyka. Přitom členové \LaTeX -teamu jsou toho názoru, že to možná bude zapracováno až do \LaTeX u 3.

6 Báječní muži na počítačích strojích

Vzpomínám-li na přelom 80. a 90. let minulého století, pak z hlediska rychlosti rozvoje informačních technologií mluvím vlastně o počítačovém pravěku. Právě v té době začal vznikat C_ST_EX. Tak jako „báječní muži na létajících strojích“ položili základy dnešní letecké dopravy, stejně bychom o lidech, kteří se v té době motali kolem počítačů mohli říkat „báječní muži na počítačích strojích“. Zvláště jsme pak vděční těm, po kterých kromě nadšení a okouzlení nad novou progresivní technikou zůstalo taky něco užitečného. Podíváme-li se na historii C_ST_EXu, můžeme směle prohlásit, že tam najdeme takových báječných mužů celou řadu (ve svém článku jsem zmínil i jednu ženu). Je možné, že jsem na někoho zapomněl, protože sám jsem se dostal k T_EXu až relativně pozdě: v roce 1992.

Reference

1. Výchozí adresář C_ST_EXu je na <ftp://math.feld.cvut.cz/pub/cstex> a kopie se nalézá na <ftp://ftp.cstug.cz/pub/local/cstex>. V dalších odkazech bude toto místo označováno jako `cstex:/`.
2. Petr Olšák. *Pár poznámek k novému C_ST_EXu*. Volně šířená dokumentace k C_ST_EXu, viz `cstex:/parpozn.tex` nebo `cstex:/parpozn.pdf`. 1996.
3. Petr Olšák. *Manuál k C_ST_EXu*. Dokumentace k C_ST_EXu, `cstex:/cstexman.tex` nebo `cstex:/cstexman.pdf`. 2002.
4. <ftp://math.feld.cvut.cz/pub/olsak/tlaccent/>.
5. <ftp://math.feld.cvut.cz/pub/olsak/a2ac/>.
6. Ladislav Lhotka. *České dělení pro T_EX*. Zpravodaj Československého sdružení uživatelů T_EXu, 4/1991, pp 8–9.
7. Petr Olšák. *Program MNU: Konfigurovatelné menu pro spouštění aplikací pod DOSem*. Zpravodaj Československého sdružení uživatelů T_EXu, 3/1992, pp 141–148.