Na druhou stranu i pro hladký text (například beletrie) má TEX nepopíratelné výhody, především v dokonalém algoritmu řádkového zlomu, nezávislosti na počítači a výstupním zařízení a (alespoň) možnosti adaptace na národní prostředí. Pokud jsem měl možnost hovořit s lidmi, kteří TEX zkusili používat, pak se obvykle shodují v jednom: cesta zpět nevede, TEXu se vždy propadá.

Petr Novák, Oldřich Ulrych: *TEX, článek v SWN z března 1992*

Kapitola šestá: Američan TEX v českém prostředí

Petr Olšák

Záměrně jsem dnes nevybral citát z díla Donalda Knutha. Chtěl jsem totiž připomenout článek, otištěný v Softwarových novinách minulý rok, jehož autor (O. Ulrych) se obrovskou měrou podílel a podílí na tom, že nám TEX běhá v českém prostředí. Není sám, ale kdybych se snažil dále vyjmenovávat, pak bych mohl na někoho zapomenout (a to bych nerad). Tito lidé pracují na univerzitách v Praze, Brně, Českých Budějovicích i jinde a obrovský kus práce, kterou do „počeštění“ TEXu vložili, předávají veřejnosti vesměs zadarmo. V dnešním povídání si ukážeme, jaké problémy se musí řešit při přizpůsobení TEXu našemu jazyku a našim typografickým zvyklostem.

Od verze 3 umí TEX interpretovat všechny znaky ASCII ze vstupního souboru (pokud ho tomu naučíte). Říkáme mu proto osmibitový TEX, abychom jej odlišili od starších verzí, které nepřipouštěly ve vstupním souboru symbol z horní poloviny ASCII tabulky (sedmibitový TEX). Výhodou této staré verze bylo, že přinutila uživatele psát články přenositelné na mezinárodní úrovni. Představte si, že byste poslali do Ameriky zdrojový text článku, v němž byste citovali jména svých kolegů, a pro diakritiku v těchto jménech byste například použili kódování podle Kamenických. Takový článek by za velkou louží určitě těžko zpracovávali svou konfigurací TEXu. Autor TEXu měl tedy dobré důvody, proč zpočátku nepřipouštěl „upgrade“ na osmibitovou verzi. Nakonec takovou verzi přece jen udělal, čímž umožnil snadnější šíření TEXu do neanglicky mluvících zemí, ale na druhé straně otevřel možnost pro tvorbu „lokálních“ konfigurací, které nejsou mezinárodně kompatibilní.

Pozastavme se tedy nejprve u české diakritiky. Abychom například vysázeli větu „Problémy přizpůsobení TEXu češtině“, museli bychom v sedmibitovém TEXu do zdrojového textu napsat:

```
Probl\'emy p\v{r}izp\accent23usoben\'{\i} \TeX u \v{c}e\v{s}tin\v{e}.
```

Když by někdo něco takového uviděl, okamžitě by TEX zavrhl a oprávněně by prohlásil, že to už může rovnou editovat dokument v PostScriptovém popisu stránek a z hlediska pracnosti a uživatelské neprůstřelnosti v tom není rozdíl. Proto byl pro sedmibitovou verzi vytvořen jakýsi preprocesor. Jedná se o program, který čte vstupní soubor v domluveném kódování horní poloviny ASCII a výstupem je soubor, obsahující odpovídající značky TEXu. Ten se teprve stává vstupním souborem pro TEX.

Toto řešení má několik zásadních nevýhod. Například se stane, že potřebujeme do sazby zařadit listing programu či něco podobného. V LATEXu je pro tento účel definováno prostředí `verbatim`, uvnitř něhož je text interpretován doslova tak, jak je napsaný (včetně všech značek a konců řádků). Napíšeme-li tedy v LATEXu následující text:

```
\begin{verbatim}
    /* Toto je listing mého programu v jazyce C */
    /* mohu použít jakékoli značky: \ $ # atd.  */
\end{verbatim}
```

pak na výstupu dostaneme:

```
/* Toto je listing mého programu v jazyce C */
/* mohu použít jakékoli značky: \ $ # atd.  */
```

Pokud ale použijeme preprocesor, pak budeme nemile překvapeni. Listing našeho programu bude mít poněkud exotickou podobu:

```
/* Toto je listing m\'eho programu v jazyce C */
/* mohu pou\v{z}\'{\i}t jak\'ekoli zna\v{c}ky: \ $ # atd.  */
```

Odstranit tento problém v sedmibitové verzi, aniž bychom se drbali pravou nohou za levým uchem, nelze. Je sice pravda, že použití diakritiky ve „verbatim–konstrukcích“ je velmi řídké, nicméně se to stává. Naštěstí máme osmibitovou verzi TEXu, kde tyto problémy odpadají.

Ani v osmibitové verzi ale nemůžeme zcela zapomenout na způsob psaní textů pomocí výše zmíněných značek. Tato metoda je nutná, pokud publikujeme článek do zahraničí a potřebujeme se v něm například podepsat. Pak musíme pro akcenty ve jménu použít uvedené značky, které mají mezinárodní platnost. Neškodí si proto ukázat přehled nejpoužívanějších značek.

`\'a` — á , pozor: `\'i` dává í̇ , přičemž my chceme `\'{\i}` , což dá í , `\v{e}` — ě , `\accent23u` — ů .
Další akcenty: `\^o` — ô , ``\`a`` — à , `\"u` — ü , `\=s` — s̄ , `\.m` — ṁ , `\~n` — ñ , `\u{g}` — ğ .
Ještě speciality: `\H{u}` — ű , `\t{mn}` — m͡n , `\b{h}` — ẖ , `\d{h}` — ḥ , `\c{c}` — ç .
Další znaky: `\l` — ł , `\L` — Ł , `\aa` — å , `\AA` — Å `\ae` — æ , atd...

Uvedené akcenty jsou definovány ve všech mezinárodně uznávaných formátech (plain, LaTeX, AMSTeX) pomocí TeX-primitivního příkazu `\accent`. Tento příkaz vybere příslušný znak z fontu jako akcent (číslo odpovídá kódu ve fontu), dále změří výšku následujícího písmene a akcent posadí nad toto písmeno (případně též koriguje umístění podle sklonu písma). Všimněme si, že na český kroužek nad u se při navrhování formátů nějak zapomnělo, takže jej musíme vyjádřit přímo TeX-primitivním příkazem. Dále se v mezinárodních formátech zcela zapomnělo na česká písmena ď a ť. Akcenty pro tato písmena nelze vyřešit pouze pomocí příkazu `\accent`. To by nevypadalo hezky. Proto je třeba definovat pro tato písmena nové zkratky (např. `\md` a `\mt`), které obsahují přesný požadavek na vzájemnou polohu akcentu a písmene. To je další starost navíc. Proto radši přistupme k osmibitové verzi.

Při inicializaci formátu může program číst (kromě zdrojového textu k formátu) také jakousi tabulku kódování – tabulku `.tcp`*). V této tabulce je uvedena nejen prostá konverze mezi vstupním kódováním a kódy písmen ve fontu, ale též pro každý ASCII kód tam může být definována sekvence příkazů TeXu, které se mají při načtení takového znaku provést. Znamená to tedy, že v okamžiku inicializace formátu už musíme být rozhodnuti, jaké kódování vstupních souborů budeme používat a podle toho vybereme vhodnou tabulku. Pokud (například v redakcích) přijímáme články v různých kódováních, nejvhodnější je použít pro články v jiném kódování konvertor kódování, který bývá součástí instalace TeXu.

Tabulku `.tcp` si můžeme představit jako definiční soubor akcí pro zabudovaný preprocesor, který je nyní integrovanou součástí TeXu. Pokud bychom ji ale využívali pouze tímto způsobem, pak naše problémy s „verbatim–konstrukcemi" neodpadají. Je proto třeba využít toho, že je možno sázet písmena s kódy většími než 127. Na tom je založena myšlenka postprocesoru `dvi2dvi.exe`, jehož autorem je O. Ulrych.

V tabulce `.tcp` se definuje pouze konverze ze vstupního kódování na dohodnuté kódování znaků ve fontu. Po zpracování dokumentu TeXem máme nyní v `.dvi` souboru uloženy požadavky na sazbu písmen s kódem větším než 127. Nyní záleží na tom, zda máme bitové mapy (`.pk` soubory) připraveny pro takovou sazbu – mluvíme o tzv. osmibitových fontech. Pokud takové fonty máme, pak nemusíme žádný postprocesor použít. V opačném případě se použije postprocesor, který čte soubor `.dvi` a v něm zamění veškeré požadavky na sazbu písmen s kódem nad 127 za sekvenci požadavků, podobných těm, které by do souboru `.dvi` uložil sedmibitový TeX při použití např. příkazu `\accent`. Výstupem postprocesoru je tedy znova soubor `.dvi` (tentokrát větší), který lze použít i pro sedmibitové fonty. Říká se tomu sazba s „létajícími akcenty", protože akcenty jsou nad písmena umísťována dodatečně.

Je třeba ještě podotknout, že obyčejný uživatel se nemusí o nic starat, mnohdy ani netuší, že po každém běhu TeXu je automaticky vyvolán postprocesor.

V současné době se od používání tohoto postprocesoru upouští a přechází se na plné osmibitové fonty. Na závěrečných úpravách těchto fontů, které vycházejí z úprav Computer Modern fontů provedených Petrem Novákem, právě nyní intenzivně pracuje Karel Horák z matematického ústavu ČAV (článek píši v březnu t.r.).

Kapitola první a třetí našeho seriálu byla brána z „hardcopy" předloh a sazba byla provedena za použití programu `dvi2dvi` a sedmibitových fontů. Kapitola druhá byla svícena na osvitové jednotce a zde jsem použil už osmibitové fonty v rozlišení 1270 dpi. O technickém osudu tisku dalších kapitol v současné době nemůžu nic říci.

Namítnete, že postprocesor se jeví jako vynikající řešení, neboť se ušetří místo na disku (bitové mapy osmibitových fontů jsou určitě větší). Na druhé straně argumentem pro použití osmibitových fontů je například skutečnost, že háček (tak jak existuje ve fontu Computer Modern jako samostatný akcent) byl vyvinut daleko od našich hranic. Jeho tvar bychom mohli nazvat „mezinárodní", ne vždy odpovídající tradicím české sazby. Také příkaz `\accent` jej ne vždy posadí přesně tam, kam by jeho umístění navrhl český písmař. Dále metrické údaje o vzdálenostech mezi jednotlivými dvojicemi písmen musí být pro češtinu sestaveny s individuálním přístupem k tomuto jazyku. Zvláště pro akcentovaná písmena stále není tato záležitost zcela dokonale vyřešena. Tyto věci lze zařídit pouze za použití plných osmibitových fontů. Bude proto užitečné se v některé z dalších kapitol zmínit o možnostech, jaké má návrhář fontu při použití programu METAFONT.

Můžete si myslet, že se zabývám „nanometrovými" detaily, můžete dokonce namítnout, že ve většině časopisech často najdete háčky na úplně nesmyslných místech — doslova plavou nad písmeny. To je všechno pravda. Ovšem TeX byl navržen tak, aby právě tyto problémy šly řešit a tím se odlišuje od mnohých DTP programů.

---

* Touto možností je patrně vybaven pouze emTeX (verze pro PC). Tuto možnost jsem například neviděl na instalacích pod systémem UNIX, tam ale lze tento nedostatek řešit prostým zařazením preprocesoru, nebo použitím stejného kódování vstupu, jako je kódování ve fontu.

Zabývejme se nyní poněkud viditelnějšími problémy přizpůsobení TEXu pro českou sazbu. Se dvěma nejdůležitějšími jevy jsme se setkali hned v první kapitole na návštěvě našeho přítele Ferdy. Zopakujme si je.

První věcí byla skutečnost, že ve slušných tiskovinách by se neměly vyskytovat neslabičné předložky na konci řádku. Uvedli jsme si, že symbol vlnka: '~' ve zdrojovém textu se vysází jako mezera, ale TEX má zakázáno v takové mezeře dělit řádky. Stačí tedy doplnit do zdrojového textu za neslabičné předložky tento znak. To se udělá před závěrečným zpracováním automaticky pomocí programu `vlnka.exe`. Tento program je součástí instalace emTEXu upravené pro české a slovenské prostředí, kterou šíří Československé sdružení uživatelů TEXu pod názvem CSTEX. Dokonce tam najdete program `vlnka` i ve zdrojové podobě (necelá stovka řádků v Pascalu). Samozřejmě, že CSTEX nezahrnuje jen emTEX a zmíněný program. Jsou tam řešeny vesměs všechny problémy, o kterých je řeč v této kapitole.

Dále jsme se u Ferdy seznámili se stylem `czech`. Uveďme si nyní přehledně, co se stane při zavolání tohoto stylu na začátku dokumentu. V plainTEXu jej zavoláte příkazem

```
\input czech.sty
```

a v LATEXu či AMSTEXu jej zařadíte do příkazu `\documentstyle` například takto:

```
\documentstyle[czech]{article}
```

Zavolání tohoto stylu ovlivní činnost TEXu v následujících bodech:

- Inicializuje se české dělení slov.
- Mezery za tečkami a čárkami budou menší.
- Je definován příkaz `\csprime` pro inicializování českých uvozovek.
- Příkaz `\chapter` způsobí sazbu slova „Kapitola“ a nikoli původně definovaného slova „Chapter“.
- Totéž platí pro další automaticky generovaná slova, např. Obrázek, Tabulka, Literatura apod.
- Příkaz `\today` vysází datum česky.

Věnujme se nyní jednotlivým bodům trochu podrobněji.

1. České dělení slov. TEX pracuje s tabulkou vzorů pro dělení a s tabulkou výjimek. Obojí se načítá v době inicializace formátu. Autorem české tabulky vzorů je Ladislav Lhotka. Přestože tato tabulka má velmi dobrou účinnost, občas se při dělení vyskytne chyba. Reklamace na nesprávné dělení lze posílat autorovi, který tabulku neustále aktualizuje. Způsob, jak se taková tabulka dělení definuje, je velice zajímavý a byl vyložen v článku v SWN z března minulého roku. Najdete jej samozřejmě v knize *The TEXbook*. Algoritmus na dělení slov zabudovaný do TEXu vytvořil Frang. M. Liang a obhájil na něm u Knutha dizertaci (Není to jediná dizertace, jejíž výsledek je zahrnut do TEXu a METAFONTu.)

Tabulku výjimek si může zapsat každý uživatel do svého vlastního slovníku a načítat ji v úvodu svého dokumentu. TEX je schopen načíst do paměti tabulky pro dělení slov až pro 256 jazyků. Pokud nemáme aspoň PC-386 s dostatečnou pamětí, pak je vhodné při generování formátu načíst tabulky nejvýše dvou jazyků (běžně anglický a český). Mezi jednotlivými jazyky lze přepínat i uprostřed věty pomocí TEX-primitivního příkazu `\setlanguage` . Také stojí za zmínku schopnost TEXu dělit slova „netradičně“, např. německé slovo Zucker umí v případě rozdělení psát jako Zuk-ker.

2. Mezery za tečkami a čárkami. TEX pracuje s tzv. „SF-kódy“ (space-factor codes), které je možno přidělit každému znaku. Tyto kódy způsobí deformaci mezery za určenými znaky (deformace vzhledem k „obyčejné“ mezeře, definované ve fontu). PlainTEX, LATEX i AMSTEX vycházely z tradic americké sazby. Tam dávají přednost poněkud protaženějším mezerám za tečkami a čárkami. V české sazbě je tomu právě naopak. Čeští sazeči se snažili o pokud možno stejnou vizuální plochu mezi všemi slovy a tudíž mezery za tečkami a čárkami zkracovali. Vyvoláním stylu `czech.sty` se tedy předefinují „SF-kódy“, definované v uvedených formátech a nastaví se (zatím) kompromis: mezery se ani nezkracují, ani neprodlužují. Na terminálu počítače to poznáte podle hlášení

```
\fenchspacing is set on
```

Pokud si vyzkoušíte rozdíl mezi první a druhou variantou, můžete znova namítnout, že se zabývám „nanometrovými“ problémy. Nebudu znova opakovat, proč.

3. České uvozovky. Dvojice znaků `` ` ` `` se vysází jako ligatura “ a dvojice `''` se vysází jako ” . Na „českou“ variantu uvozovek ve fontu Computer Modern pamatováno nebylo. Proto je ve stylu `czech.sty` definován příkaz `\csprime`, který předefinuje chování uvedených dvojic znaků tak, aby se sázely tytéž ligatury, ovšem v jiné výšce a v jiném pořadí. Dále i jednoduché uvozovky (symboly ` a ') se začnou sázet ‚česky‘.

4. Automaticky generovaná slova. Vyšší formáty (LATEX a AMSTEX) umožňují automatické generování některých slov. Například v LATEXu lze pro seznam literatury použít prostředí

```
\begin{thebibliography}
...
\end{thebibliography}
```

Tento seznam má stanovenou strukturu, přičemž např. odkazy a reference na literaturu v dokumentu se generují automaticky. Lze také použít seznam literatury z jisté databáze (bibTEX). Ve všech případech se seznam literatury sází jako samostatná sekce a pokud nepoužijeme styl `czech`, bude mít tato sekce název „References“ případně „Bibliography“. Použití stylu `czech` předefinuje tyto příkazy LATEXu tak, že na výstupu dostaneme název „Literatura“. Podobná věc platí i pro další automaticky generovaná slova. Například příkaz LATEXu `\listoftables` vysází seznam tabulek nadepsaný jako „Seznam tabulek“ a nikoli jako „List of Tables“. Naprosto analogická záležitost se týká příkazu `\today`, který už známe z první kapitoly.

Nevýhodou všech mezinárodně používaných formátů je skutečnost, že jej navrhli typografové v Americe, tedy vycházeli z tamnějších tradic sazby. Tyto tradice se v mnohém liší od našich a ne všechno je ve stylu `czech` zahrnuto. Zmíním se například o problému „tečkování“. V české odborné literatuře se názvy odstavců píší s tečkou za poslední číslicí, tedy tak, jak je zde uvedeno v prvním případě:

2.4. Rostliny brukvovité **2.4 Rostliny brukvovité**

přitom LATEX vám vysází o jednu tečku méně, jak je ukázáno v případě druhém. Všimněte si ale, že tečkovací problém není jednotný ani v české literatuře. Mnohdy záleží na vkusu sazeče. Dokonce se mi podařilo objevit knihu, kde jsou oba způsoby nahodile míchány. Druhý způsob (používaný LATEXem) je ale u nás přeci jen méně běžný.

LATEX také vysadí první odstavec za nadpisem bez odstavcové zarážky. Z hlediska vkusu to jedině vítám, protože tato zarážka mnohdy opticky koliduje vertikálně právě s číslem odstavce. Na druhé straně i toto pravidlo není v naší sazbě zcela obvyklé. Takto by se dalo o různých „zvycích“ LATEXu a $\mathcal{AMS}$TEXu pokračovat. Samozřejmě, že to není problém samotného TEXu. Ten se takovým drobnostem dokáže snadno přizpůsobit, problém je jen v tom, že nevznikl žádný hojně užívaný formát TEXu v dílnách českých sazečů.

Mezi prostředky pro elektronickou sazbu patří řada pomocných programů, které musely být upraveny s ohledem na češtinu, případně vytvořeny zcela znova. Například korektor překlepů šířený v CSTEXu je shodný s tím, který se používá v Mladé frontě Dnes. Tento korektor není „public domain“, nicméně členové Československého sdružení uživatelů TEXu jej mohou dostat zadarmo. Podobná situace je s editorem CSED, který je vhodný pro přípravu zdrojových textů psaných v českém a slovenském jazyce. Přímo z tohoto editoru lze volat zmíněný korektor překlepů. Dále například program na seřazování slov podle abecedy (při vytváření rejstříků) je public domain a je upraven Z. Wágnerem z anglické verze tohoto programu na česká a slovenská pravidla abecedního řazení slov podle normy ČSN 01 0181.

Uveďme ještě jeden často přehlížený problém, týkající se typografických metrických jednotek. V Americe je 10 tiskařských bodů (monotypových) rovno zhruba 9.34 bodům, používaným v sazárnách v Evropě (Didotův měrný systém). Proto by se asi nejeden odborník na typografii podivil, co to říkám za nesmysly, když tvrdím, že toto písmo je desetibodové. Ono spíš vypadá jako devítibodové. Je to tím, že je použit americký font, jehož stupně (velikosti) nejsou kompatibilní se stupni písma, používaných v Evropských sazárnách.

Tady stojíme před základní otázkou. Budeme-li vytvářet (v budoucnu) zcela nové fonty, vycházející z tradic českého písmařství, měli bychom použít i původní stupně (velikosti) písma. Víme, že TEX umožňuje návrat k tradicím ruční sazby (návrh každého stupně písma zvlášť, používání ligatur apod.). Dále TEX a METAFONT samozřejmě znají i Didotův bod (má označení `dd`), takže by neměl být problém teoreticky všude místo jednotky `pt` pracovat s jednotkou `dd`. Vytvoření písma, které vychází z Didotova měrného systému, je tedy teoreticky možné, ale znám bohužel jen velmi málo lidí, kteří by takovou věc ocenili. Rozvoj fotosazby a posléze elektronické sazby doslova převálcoval některé po staletí vytvářené tradice. Veřejnosti je například systematicky předkládán americký font TIMES v podstatě ve všech periodikách. Bibliofilové a typografové se před tímto náporem uchylují k návratu zpět k horké sazbě. Tudy ovšem cesta nevede. V tomto seriálu jsem chtěl mimo jiné ukázat, že existují elektronické systémy, které splní veškeré požadavky na kvalitní sazbu. Stačí pouze přiložit ruku k dílu a vytvořit rodiny fontů, které vycházejí z tradic českého písmařství a typografie.

V neposlední řadě nám stojí v cestě ještě jedna překážka. Mám na mysli problematický přístup českých nadšenců pro typografii a TEX k zařízení s dostatečnou rozlišovací schopností. Pro profesionální práci při návrhu fontů je potřeba zařízení s rozlišením kolem 2000 dpi. Taková zařízení jsou však na našem území pro experimentování s TEXem cenově velmi těžko dostupná. Přitom navrhovat nové tvary písma na laserových tiskárnách s 300 dpi je stejné, jako snažit se tlustým fixem napodobit perokresbu.

Všimněte si v závěru, co to znamená přizpůsobení programového systému češtině. Jedná se o pokud možno dokonalé přizpůsobení *výstupu* systému tradicím českého jazyka. Vůbec se zde nezmiňuji o takových hloupostech, jako je počeštění nabídek a helpů, což je často u jiného software zdůrazňováno na prvním místě (a mnohdy tím celé slavné „počešťování“ končí a to většinou ne vždy s nejlepším výsledkem). Takový problém se v souvislosti s programem TEX vůbec neřeší, protože, jak jsme si řekli dříve, záleží na uživateli, pod jakým prostředím (editor, menu apod.) bude pracovat. Důraz je zde kladen hlavně na výsledek činnosti systému.

Příště se zaměříme na „programování“, tj. ukážeme si nejzákladnější vlastnosti makrojazyka TEXu.