

OBSAH

Mirka Misáková: Písmo s variantní šířkou: nová naděje pro naše úzké sloupce?	65
Pavel Janík ml.: Fonty ve formátu TrueType v pdf \TeX u	82
Antonín Strejc: Recenze: CE Písma, loga, symboly II	88
Michal Kvasnička: Fontanasia	96
Jaroslav Fojtík: Konverze dokumentů z WordPerfectu do \LaTeX u – $\WP_2\LaTeX$	97
Zdeněk Wagner: \LaTeX ová kuchařka/4	105
Štefan Porubský: Kam za TUGboat-mi?	127

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

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

Zdeněk Wagner
Vinohradská 114
130 00 Praha 3

Disketu formátujte nejlépe pro DOS, formát Macintosh 1.44 MB je též přijatelný. Nezapomeňte přiložit všechny soubory, které dokument načítá (s výjimkou standardních součástí \TeX u), zejména v případě, kdy vás nelze kontaktovat e-mailem.

ISSN 1211-6661

Písmo s variantní šířkou: nová naděje pro naše úzké sloupce?

MIRKA MISÁKOVÁ

Abstrakt

Algoritmus řádkového zlomu optimum-fit (implementovaný v $\text{T}_{\text{E}}\text{X}_{\text{u}}$) je obecně považován za velmi kvalitní. Přesto však existuje rozsáhlá podmnožina zlomových problémů, kde jsou jeho výsledky neuspokojivé. Jde zejména o sazbu do úzkých sloupců při požadavku zarovnaných okrajů, kdy bývá výsledku dosaženo typicky pouze použitím dodatečného rozvolnění bílého místa (`\emergencystretch`), což vede k esteticky nepřijatelným poruchám v pravidelné šedosti odstavce.

Jedním z možných pokusů o řešení tohoto problému je návrat k staletí nepoužívané technice, vycházející z principů uplatněných při sazbě Gutenbergovy 42 řádkové Bible: rozšíření znakové sady o šířkové varianty liter. Výběr vhodného takto modifikovaného řezu pro jednotlivé řádky pak pomůže minimalizovat děravost sazby.

Úvahy o implementaci této techniky v digitální sazbě zřetelně vedou k použití systému METAFONT (který umožní při modifikaci písma zachovat konstantní šířku tahu) a k využití stávající podoby optimum-fit v $\text{T}_{\text{E}}\text{X}_{\text{u}}$. Skutečně systémová podpora této techniky by vyžadovala implementaci zcela nového sázecího systému (přinejmenším jeho zlomové části); i s menšími náklady však můžeme zkoumat samotnou smysluplnost metody a provádět estetické experimenty.

Tato prezentace ukazuje výsledky jedné z těchto možných jednodušších implementací, která za spolupráce $\text{T}_{\text{E}}\text{X}_{\text{u}}$, METAFONTu a perlového postprocessingu nad DVI dovoluje uživateli novou techniku prozkoumat a případně použít ve svém dokumentu.

Demonstrovány jsou ukázky řešení zlomu odstavců $\text{T}_{\text{E}}\text{X}_{\text{em}}$ a změny v šedosti po přelámaní s variantními šířkami písma. Jsou diskutovány hranice a omezení nové metody.

O průměrném dokumentu

Napíše-li jednoho krásného dne $\text{T}_{\text{E}}\text{X}$ ista svoje `\bye` a vydá se na procházku do skutečného světa, za brány strážené složenými závorkami, patrně jako obvykle zjistí, že kvalita průměrného dokumentu vytvářeného počítačovými systémy konce dvacátého století je stále neuspokojivá. Při bližší analýze většího

statistického vzorku možná shledá, že situace je přece jen lepší než třeba před pěti lety – prvotní nadšení z pouhé existence DTP konečně opadá a návrháři systémů na tvorbu dokumentů i uživatelé samotní začínají projevovat určitou osobní disciplínu při osvojování si výtvarných 500 let vyvíjející se nauky, zvané typografie. Drtivou většinu drobných sazečských problémů, vyvstávajících před sazečem hladké sazby z věčného zápasu tří paradigmat (jednotnosti, informace a struktury) černého řemesla, dovede již řešit každý program, který si osobuje právo být nazýván sazečským systémem. Patří k dobrému tónu zvládat dělení slov, ligatury, kerning, nezlomitelné mezery, vícero pomlček.

O to paradoxněji působí absence pokroku na úrovni, kterou čtenář vnímá podstatně rychleji a razantněji: v boji proti děravé sazbě. Čím to, že toto téma, tolik důležité pro tiskaře ze starých dobrých dob, je nyní tak neuspokojivě opomíjeno, ve vzácné shodě téměř všemi systémy? Snad je to proto, že odstranění děravé sazby nelze dosáhnout aplikací nějakého jednoduchého algoritmického postupu. S trochou nadsázky můžeme říci, že co se týče jednolitě šedi stránky, zůstaly digitální sazárny na úrovni psacích strojů. Beznadějně pak působí skutečnost, že dokonce i \TeX je na tom díky zlomovému algoritmu optimum fit pouze relativně lépe, ale ani on se nevyhne všem problémům.

Co je to totiž *děravá sazba*? Můžeme například říci, že sazba hladkého textu, jejíž mezerovité mezery nejsou v rozmezí 66%–150% šířky ideální mezery, jak ji navrhl tvůrce fontu. I u dokumentů sazovaných \TeX em se velmi často setkáme i s několikanásobným překročením udané horní meze šířky. Jak se pokusit meliorovat obdobné výsledky udává např. Philip Taylor v [6], ale všeobecný konsenzus uznává, že například zarovnané úzké sloupce s pravidelnou úrovní šedi textu jsou často neřešitelnou úlohou. Se snižující se `\hspace` totiž problémy se zlomem samozřejmě citelně narůstají . . .

Dnešní úzké sloupce

Nač ovšem vůbec potřebujeme zarovnané úzké sloupce? Nebylo by nejjednodušším řešením saziet cca 70 znaků na řádek, pro oko čtenáře (podle [4]) nejpříjemnějších? Nebylo. Sazba je vždy kompromisem mezi ergonomií čtení a celkovým designem, který může na zlom obtížné partie požadovat (obtékání obrázků, sazba denního tisku). Daleko typičtějším postupem bývá mezispisový rozpal, řešení jak otřesné tak hojně používané. Je s podivem, že se metoda tak striktně odsuzovaná¹, vůbec do počítačové sazby dostala. Dnes navíc její použití uvádí do rozpaků zejména čtenáře některé z národních typografických škol, které letter-spacing používaly jako zvýrazňování `p r o s t r k á v á n í m` při nedostatku vhodné kursivy. Pouze delší úvaha na téma „co autor mohl myslet zvýrazněním právě

¹“A man who would letterspace lowercase would steal a sheep.” Traduje se, že tuto poznámku F. Goudy utrousil při pohledu na čestný diplom, předávaný mu za vynikající design.

tohoto slova“ obvykle vede ke zjištění, že je na řádku samo a je tedy rozpáleno čistě z důvodu zarovnání sazby. Proces přenosu informace se tak neobyčejně zpomaluje a komplikuje.

Daleko méně odsouzeníhodným řešením je použít praporek, tj. řádky nezarovnávat. Důvody, které brání zvolit tento styl pro veškeré úzké sloupce, jsou poctivě řečeno, iracionální. Ale typografie, jakožto disciplína sloužící veskrze iracionálním bytostem, na ně musí brát ohled. Lidé prostě *chtějí* zarovnané sloupce. Tak jako architekti (často s typografy srovnávaní) by zákazníkům jen stěžii vnutili nezarovnané stěny, tak se na knihu upravenou na praporek díváme neúvěřivě; tolerujeme ho snad pouze v místě, kde by pokus o zarovnané řádky způsobil ještě horší porušení jednotné šedi, než praporek sám.²

Chtělo by to tedy nějaký další způsob, použitelný v boji proti děravé sazbě. Jednou z možností je právě přístup diskutovaný ve zbývající části textu, tj. sázet z písma, které obsahuje variantní řezy širších a užších liter (při zachování základních prvků jejich kresby). Tím je dán lámacímu algoritmu další stupeň volnosti při hledání optimálního zlomu; algoritmus není omezen na pouhé rozšiřování mezer. Některé situace kritické pro běžnou sazbu se v takovém systému stávají snáze řešitelnými (například řádky, na nichž je minimum mezer – o to více je tam písmen, která je možno jemně variovat a dosáhnout slušného výsledku. Písmen je sice více než mezer, není ovšem možné je modifikovat třeba o polovinu jejich šířky, jak je to běžné u bílého místa. Pro zjištění, nakolik je tedy tato idea nosná a použitelná, byly využity právě experimenty, které daly základ jádru tohoto článku.

Diskuse smysluplnosti

Širší emka. Užší očka. Není to náhodou danajský dar, který ve snaze sazbu zpravidelnit naopak naruší zřakovou pohodu, poněvadž mozek čtenáře bude zmaten nezvyklými nepravidelnostmi ve tvarech znaků? To je závažná otázka. Povrchní specifikace problému navíc může vést k představě výsledku, který zkušený typograf odsoudí – zdá se, že jde o *míchání fontů* v jeho nejhorší podobě, vždyť na stránce se budou vyskytovat patrně desítky různých řezů, v nejhorším případě bude mít každý řádek svůj specifický font. Zavrhováno je přitom již použití více než zhruba 3–4 fontů v celém dokumentu. Zde ovšem není záměrem učinit dokument co nejpestřejším (jak tomu patrně je u sazečů hýřících rozmanitými rodinami a řezy), ale naopak utlumit ho do dokonalejší jednotvárnosti. Modifikace znaků by neměly překročit mez, za níž už jsou rozpoznatelné bez bližšího

² Jiným příkladem toho, jak čtenáři hatí elegantní algoritmická řešení, je dělení slov v esperantu. Tvůrce tohoto jazyka povolil dělit v libovolném místě libovolného slova; uživatelé jazyka však přesto vymýšlejí různá umělá omezení, která vedou k vzorům pro dělení, která mají velikost srovnatelnou (nebo větší) s jinými jazyky ...

zkoumání. Míru této hranice by bylo nutno odvodit empiricky; bude se asi lišit jak pro jednotlivé čtenáře, tak jednotlivá písma. První odhad počítá s modifikacemi maximálně okolo 5 % původní šířky znaku. Dalším požadavkem by měl být jednotný vzhled v rámci celého řádku, který je největší jednotkou, na niž se vnímání čtenáře skutečně soustředí.

Zda se přesto objeví u podobného dokumentu časté regrese, neurčitý pocit nesprávnosti či prostě namáhavé čtení, to předem nelze říci.³ Zapotřebí je zřejmě řady praktických testů, jejichž výsledky navíc nelze zobecnit na odlišné písmové rodiny.

Vhodné je také připomenout, že jde primárně o práci s minuskami; v textu titulku, na němž má oko delší dobu spočinout a mimovolně zkoumat zobrazené kontury, je zřejmě nutné uplatnit ještě jiné principy, než u sazby hladkého odstavce, kde je cílem nerušit, předat informaci a zmizet.

Historické reminiscence

Je-li typograf na pochybách, je vždy vhodné obrátit se do minulosti, ke zkušenostem prověřeným generacemi. Při studiu historických souvislostí vidíme, že určité modifikace této metody používal ne jeden sazeč, který v obtížné situaci potřeboval dokonale vyrovnanou sazbu. Oldřich Hlavsa [2] uvádí příklad variabilních znaků, které se objevily ve vzorníku písem z roku 1920.

Za zmínku stojí také fakt, že v tradiční horké sazbě byla běžná (přibližně lineární) kontrakce šířek znaků do cca 1 %. Dosahovalo se jí pevným dotažením šroubů na sazečské loďce při využití plasticity písmové litiny.

Co na to Jan z Dobré Hory?

Pokud bychom měli snahu považovat výše zmíněné precedenty za ojedinělé výstřelky, můžeme zamířit ještě hlouběji ke kořenům, do dílny Johanna Gutenberga. Přesné záznamy o jeho technikách *umění množování knih* nejsou známy, ale přesto víme, že obdivovaně stejnoměrně šedi své 42řádkové bible dosáhl za použití desítek slítků, častých zkratek, umístění interpunkce do středu mezislovního místa a zejména neobyčejně rozsáhlé znakové sady. Patrně právě výběr liter vždy z několika šířkových variant mu umožnil sázet ony dokonale zarovnané řádky, kterými nadchl Evropu a kterými se tak zásadně odlišoval od manuskriptů. Dá se předpokládat, že jeho záměrem nebylo nic menšího, než dosáhnout *jednotně širokého* bílého místa v celém dokumentu. Obrovské množství práce, které problému věnoval, znovu potvrzuje, jak významnou úlohu děravé sazbě staří sazeči přisuzovali.

³Cesty lidského vidění jsou podivné. Příkladem budiž dlouhý spor o bezpatkové písmo, jež by mělo být čitelnější, neboť bez zamlžování serify přivádí oko přímo k podstatnému tvaru písmene. Přitom je ke čtení méně vhodné, neboť mu chybí „uzavírací lišta“, v níž jsou písmena pomocí patek sevřena a která plynule vede oko po řádku.

BOSSEMAGNETE
LICHTMASCHINEN DER FIRMA EISEMANN
PALLAS-VERGÄSER/ESSENER LÄGERMETALL
HEINRICH MEISNER
KONSTRUKTIONSMATERIAL/BASEL-SUD 11

všichni vy na Slezské, všichni vy dím,
hlubokých páni vy dolů;
přijde den, z dolů jde plamen a dým,
přijde den, sůčtujem spolu!

الفئة
الفئة

BOSSEMAGNETE
LICHTMASCHINEN DER FIRMA EISEMANN
PALLAS-VERGÄSER/ESSENER LÄGERMETALL
HEINRICH MEISNER
KONSTRUKTIONSMATERIAL/BASEL-SUD 11

< e e e a a t t k k >

Obrázek 1: S pomocí variantních písem bylo možné bez zvláštního úsilí vytvořit zarovnanou, ale přesto těsně sázenou reklamu. (*vlevo*) Teprve podrobnější pohled na Preissigovo řešení sazby básnické sbírky odhalí modifikované varianty některých liter, kterými dosahuje řádků s vyrovnaným světlem a malebnějšího vzhledu. [3] (*uprostřed*) Zajímavý je také pohled na některá písma nelatinková, která obdobný nápad – měnit šířku užitých znaků – aplikují zcela běžně. (*vpravo*)

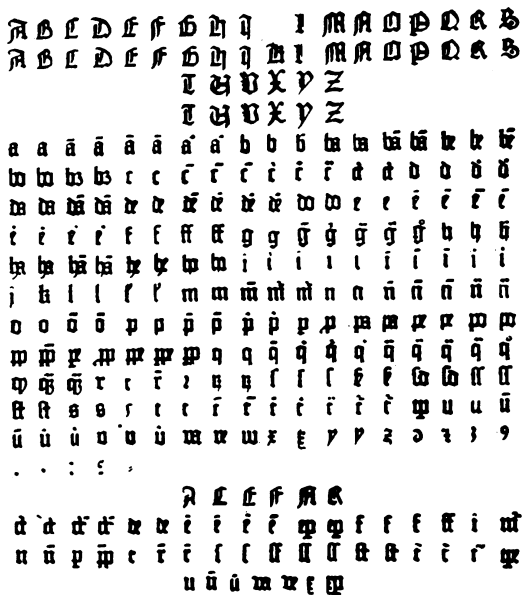
Implementace

Kdy přesně v průběhu historie ztratili sazeči potřebu tvořit stránky s tak dokonalou pravidelností šedi mi není známo. Ze všeho nejpravděpodobněji tato tradice nepřežila již přechod ze švabachově texturových písem ke kulatým latinkám dnešního typu. S opuštěním výrazných vertikálních rysů písma zmizela snaha vytvořit z tiskového zrcadla pravidelnou mříž a postupem staletí převládla jediná, nejpohodlnější metoda, jak zarovnat okraje řádku: vložím širších mezislovních mezer. Techniky jiné, jak jsme ukázali, se sice táhly celými dějinami sazby, nikdy však už nebyly nasazeny plošně. Domnívám se, že to nebylo způsobeno estetickými odsudky, ale nepřekonatelnými technologickými obtížemi. Teprve elektronická sazba přináší jednoduché způsoby, jak s podobnými mikrotypografickými efekty začít experimentovat a případně je začlenit do našich dokumentů.⁴

Fonty

Při úvahách, jak myšlenku variování šířek liter uvést do života, se jako první zásadní problém objeví fonty. V přípravě variantních řezů lze využít v zásadě

⁴Skutečně intenzivní a transparentní použití variantně širokých liter by samozřejmě znamenalo zcela novou generaci lámacího modulu sázečícího algoritmu. Takový úkol byl zcela nad možností diplomky, v jejímž rámci byly tyto teze diskutovány. Pustila se do něj nicméně firma URW; její hz-system je však typicky komerční produkt: informační vakuum je dokonalé, testovací ani jiná verze není, pokud vím, na trhu. Přesto byla tušená existence hz-systemu pro autorku inspirací a primární nadějí, že zvažovaný přístup *má smysl* zkoumat. Soustředila se ovšem na spolupráci s TeXem, na vybudování pomůcek, které komukoliv umožní testovat použitelnost a limity metody; jednou ji někdo třeba implementuje skutečně systémově.



Et ingressus angelus ad eam dixit. Ave gratia plena: dominus tecum: benedicta tu in mulieribus. Que cum audisset: turbata est in sermone eius: et cogitabat qualis esset ista saluatio. Et ait angelus ei. Ne timeas maria: inuenisti enim gratiam apud deum. Ecce concipies in utero et paries filium: et uocabis nomen eius ihesum. Hic erit magnus: et filius altissimi uocabitur. Et dabit illi dominus deus sedem dauid patris eius: et regnabit in domo iacob in eternum: et regni eius non erit finis. Dixit autem maria ad angelum. Quomodo fiet istud: quoniam uirum non cognosco? Et respondens angelus dixit ei. Spiritus sanctus superueniet in te: et uirtus altissimi obumbrabit tibi. Ideoque et quod natus erit te sanctus: uocabit filium dei. Et ecce elizabeth cognata tua: et ipsa concepit filium in senectute sua. Et hic mensis est sextus illi qui uocatur sterilis.

Obrázek 2: Znaková sada použitá v Gutenbergově bibli čítala stovky položek. (vlevo) Ze 42řádkové Bible. (vpravo)

asi dvou přístupů: (a) předem poněkud rozšířit znakovou sadu o varianty určité skupiny znaků, nebo (b) řez vhodné šířky generovat až za běhu, podle požadavků sázecího systému. První řešení, patrně využitě v hz-systému, má určité nevýhody (omezená variabilita v důsledku pevně dané množiny znaků), ale i výhody: znakovou sadu může/měl by připravit školený grafik a zabránit případným excesům, které mohou vzniknout při automatickém generování, a disková režie je menší. Druhé řešení vyžaduje velmi dobrou spolupráci sázecího systému s programem na tvorbu fontů. Taktéž množství fontů použitých v dokumentu bude patrně enormní⁵. Požadavek měnit kresbu znaků při zachování základních charakteristik písma (zejména šířky tahu) diktuje použití METAFONTu.⁶

Řádkový zlom

„Za použití *optimum fit* algoritmu k děravé sazbě prakticky nedochází.“ To je názor vycházející ze zkušeností prvních let používání T_PXu. Skutečnost je však

⁵Zmírnit tuto nevýhodu by bylo možné přechodem na nový typ správy fontů: něco na způsob fontserverů, které generují vždy pouze potřebné znaky a nikoliv celé řezy.

⁶Ačkoliv do budoucna je možno spatřovat určitou naději v Multiple Master systému, který snad do postscriptového světa zavede tolik potřebou metaness.

jiná. Lidé jsou příliš líní suplovat práci algoritmu pro dělení slov či přeformulovat svoje sdělení. Koncept *optimum fit* a celého paradigmatu box-glye-penalty je nicméně velmi silný; jeho (patrně nikoliv příliš obtížným) rozšířením o objekt *pružný box*, který by v sobě slučoval některé vlastnosti boxu a glye, by bylo možné lámání přepracovat tak, aby bral v úvahu, že i materiál boxu vytvářeného řádku má určitou roztažitelnost. Dnešní badness řádků je počítána pomocí vztahu $b = |r|^3 \times 100$. V případě možnosti natahovat jak mezery tak znaky bychom mohli při určování poměru roztaženosti r vzít v úvahu něco jako

$$\alpha \times \text{deformovanost mezer} + \beta \times \text{deformovanost znaků}.$$

Vyvážením α a β by uživatel mohl vyjádřit, zda preferuje řádky spíše děravé nebo řádky s deformovaným fontem. Vhodným nastavením těchto parametrů by také mohl dosáhnout zpětné kompatibility s $\text{T}_{\text{E}}\text{X}$ em.

Co s tím v $\text{T}_{\text{E}}\text{X}$ u

Dosti už teoretizování. Pojdme se podívat, co je možné udělat v dnešním $\text{T}_{\text{E}}\text{X}$ u, abychom konečně viděli, jak taková novátorská sazba *vypadá*; jak funguje a působí na čtenáře. Po zvážení různých přístupů (prototypový systém jako změnový soubor k $\text{T}_{\text{E}}\text{X}$ u, testovací kostra sázecího systému nezávislá na $\text{T}_{\text{E}}\text{X}$ u a další) zvítězila myšlenka postprocessingu nad DVI a kooperace Perlu, $\text{T}_{\text{E}}\text{X}$ u a META-FONTu.

Metoda

Při přípravě systému bylo zapotřebí řešit tyto okruhy problémů:

Řádkový zlom s přihlédnutím k pružnosti boxů. Pružné boxy implementované řešením simuluje za pomoci pružného datového typu v $\text{T}_{\text{E}}\text{X}$ u již existujícího: *glue*. Optimum fit v $\text{T}_{\text{E}}\text{X}$ u bere v úvahu obsah registru `\rightskip` (obsahujícího glue, které má být umístěno na pravý okraj řádku). Zlomíme-li odstavec s nastavením například `\rightskip=0pt plus 0.052\hsize minus 0.047\hsize`, (`\hsize` udává šířku zrcadla), dosáhneme stejného efektu, jako bychom všem objektům na řádku umožnili roztažení/stlačení o 5 % procent. Tyto zalámané řádky dále (za pomoci vhodného makra) opatříme značkami udávajícími pozici začátku a konce řádku. K tomu dobře poslouží primitivum `\special`, které umožňuje zapsat do DVI výstupu značky důležité pro další zpracování.

Rozpal – správné usazení materiálu na zalomeném řádku. V této fázi je opuštěn $\text{T}_{\text{E}}\text{X}$ a další práce probíhá nad výstupním souborem DVI, který je analyzován perlovským skriptem. Je ponechán beze změny až do míst, kde se nacházejí `\special` značky označující místa začátků a konců řádků. Prostor mezi párem těchto značek udává plochu, kam je zapotřebí rozmístit objekty řádku a minimalizovat odchylku od požadované světlosti. Skript zjistí, jakou šířku mají

vyskytující se znaky; o posunech heuristicky rozhodne, zda se jedná o mezeru či kerning (kerning nemodifikuje, mezery chce přesázet). Spočte, o kolik je zapotřebí variovat font, a tímto novým fontem řádek vysadí. Není-li potřebná metrika k dispozici, počká, až ji další skript vygeneruje.

Příprava variantních fontů sestává z automatizované výroby zdrojových textů v METAFONTu. Za základ slouží rodina DC fontů. Perlový skript `generujfont` dostává jako parametr font, z jehož názvu plyne, z jakého řezu je odvozen a o kolik procent se liší (například `dcr8+3w.mf` je osmibodový font rozšířený o tři procenta). Odpovídajícím způsobem je modifikován zdrojový text původního řezu (konkrétně tedy hodnota jeho parametru pro šířku `u#`) a pomocí METAFONTu vyrobeny metriky a bitmapy nového fontu. Implicitní snahou bývá připravit deset šířkových variant (lišících se odchylkou v intervalu $\langle -5, -4, \dots, 4, 5 \rangle$ procent od původního řezu), konkrétní sazba je poté provedena fontem nejvíce se blížícím vypočtenému požadavku. Mít pro každý řádek dokumentu skutečně vlastní font by nebylo z výpočetního hlediska patrně zvládnutelné. Ve speciálních případech je možné na požádání vygenerovat nezaokrouhlenou variantu.⁷

Ekvivalence systémového a implementovaného řešení

Implementované řešení je v mnoha ohledech skutečně pouhou simulací systematického přístupu. K nejzávažnějším zjednodušením patří:

- Fakt, že x % šířky řádku není rovno součtu x % šířek pružných boxů na něm se vyskytujících. Rovnost platí pouze za předpokladu, že můžeme variovat šířku *všech* zúčastněných objektů. Původní záměr zněl dosáhnout dokumentu se všemi mezerami přesně stejnými, proto můžeme hodnotit negativně skutečnost, že při zlomu se počítá i s modifikací mezer. Na druhé straně můžeme na metodě, která deformuje písmena, ale mezery naprosto nikoliv, shledávat cosi umělého. Problematičtější je fakt, že na řádku mohou být další oblasti, které je zakázáno modifikovat. Uživateli je zajisté nutno dát možnost, aby sdělil, že určitou pracně vybudovanou sazečskou konstrukci nechce pozměnit ani o mikron anebo že v určitém místě explicitně mění referenční bod sazby.⁸ Aby bylo možno zlepšit sazbu alespoň ve zbytku odstavce, je uživateli doporučeno uzavřít nebezpečné úseky sazby opět do dvojice `\special`ových značek, které rozpalovacímu algoritmu ozřejmí, že označený materiál má vysadit beze změny. Nepříjemností zůstává ovšem možná chyba při zalámání: předpoklad možnosti modifikovat materiál řádku o x % selhává, je-li přítomna nemodifikovatelná oblast široká podstatný zlomek řádku.

⁷Například pro oblíbenou úlohu „vysadit titulkové slovo přesně na danou šířku“.

⁸To stávající implementaci rozpalu nad DVI zmate; k testovacím účelům zcela stačí texty, jejichž horizontální a vertikální souřadnice přirozeně rostou – jiné deviace nejsou ošetřovány.

- Pouze jediný možný přístup, jak pohlížet na badness tvořeného řádku. V tomto řešení je pružnost boxů simulována přidáváním glue do proměnné `\rightskip`. Jedinou možností je tedy počítat badness běžným $|r|^3 \times 100$, bez rozlišení škodlivosti bílého místa a deformovaných znaků. Stejně tak při hledání nejvýhodnějšího celého odstavce nejsme schopni přibrat do výpočtu nějakou obdobu `\adjdemerits`, která by penalizovala sousedství řádků s rozšířenými a zúženými literami.
- Neanalyzování obsahu hboxů. Vyskytne-li se na řádku hbox, je v DVI souboru obvykle reprezentován další úroveň zanoření zásobníku. Protože přítomnost této konstrukce zhusta věští nějaké netypické jevy v sazbě (odlišnost skutečné a deklarované šířky objektů, explicitní posuny sazby tam a zase zpět, složitější uživatelem konstruované objekty), zanechává implementované řešení tato místa sazby beze změny.

Estetické experimenty

Jednotlivé ukázky jsou opatřeny komentářem a numerickými charakteristikami, přesto je doporučeno provést nejdříve estetická pozorování vlastní, nezatížené autorčinnými předsudky. Empirická zjištění ukazují, že rozdíly ve vnímání mikrotypografických efektů se u různých jedinců diametrálně liší; pro nějaká nezpochybnitelná hodnocení by asi bylo zapotřebí uspořádat rozsáhlou sérii psychologických a ergonomických srovnávacích testů.⁹

Při pokusu ohodnotit výsledky práce alespoň nějakým počitatelným způsobem byly zvoleny následující metriky:

Badness, tj. ohodnocení roztaženosti glue na řádku, jak ho spočte T_EX. Je uváděna u některých příkladů ukazujících, jak by daný text zalomil stávající T_EX. Současný pohled na hodnotu *badness* a vzhled řádku (zejména rozpal mezer) může dát dobré vodítko pro uchopení pojmu např. ‚sazby s `\tolerance1000`‘. Pro definování stavu nabízených ukázek však *badness* nepostačí, neboť drtivá většina demonstrováných úzkých sloupců prochází do třetího průchodu zlomového algoritmu (kdy je k roztažitelnosti řádku přidána hodnota `\emergencystretch`). V tomto průchodu (bohužel?) T_EX přidane glue zohledňuje při výpočtu ideálního odstavce, badness jeho řádků a celkových *demerits*, ale při konečném vysazení a zápisu o překladu jsou hodnoty *badness* přehodnoceny, jako by k přidání *emergencyglue* nedošlo. Údaje o *badness* takového odstavce mají tedy podstatně informativnější (ve smyslu méně přesný) charakter. Proto je uváděna ještě metrika další,

⁹Veškerá pozorování laskavých čtenářů o migrénách vzniklých nekonečným drážděním zrakových nervů, které se znovu a znovu bezúspěšně snaží uchopit abecedu tak, jak ji znají (tj. se stálou šífkou), jsou s díky vítána.

Procentová odchylka od ideálně široké mezery. Při zjištění těchto hodnot se vychází z rozměru první mezislovní mezery na řádku (to je ovšem určité zjednodušení, neboť při výskytu více řezů písma na jednom řádku nemusejí mít všechny mezislovní mezery stejnou šířku, ale jako orientační ukazatel kvality řádku je to dostačující). Záporná hodnota ovšem znamená stlačení, např. *Overfull boxy* mají mezery široké -33% . Zaokrouhlováno je na celá procenta. K odstavcům přelámaným s přihlédnutím k rozšířené sadě variant šířek písma je připojena

Procentová odchylka šířky použitého písma. Kladná hodnota znamená, že bylo zvoleno písmo o daný počet procent širší než původní, záporná hodnota signalizuje zúžení. Vyhledáním sousedících velmi rozdílných řezů (např. $+5\%$ a -4%) lze posoudit kritická místa tohoto druhu sazby, kde se oko čtenáře potýká s nejméně výraznými rozdíly v kresbě liter.

Ukázky zlomu jsou sázeny s běžnými nastaveními formátu plain (zejména `\pretolerance100, \tolerance200, \hfuzz0.1pt, \adjdemerits10000`).

Příklad první

Úvodní ze série ukázek napovídá, že při zlomu skutečně úzkých sloupců, například s (`\hsize4cm`), mívá \TeX značné problémy. Povolená hranice tolerance 200 je sice poměrně přísná, na druhé straně se nejedná o žádný matematický text s řadou nerozložitelných formulí, ani o vysoce odborný text, jehož případně pro češtinu netypické termíny by mohly mást algoritmus pro dělení slov. Zlom odstavce je však natolik obtížný, že ani po třetím průchodu nejsou odstraněny všechny *overfull boxy*. Hodnoty roztaženosti mezer v řešení za pomoci variantní šířky fontu (druhý sloupec vpravo dole) naznačují, že ani sazba s pětiprocentním praporkem nezabránila použití `\emergencystretch`. Použité fonty nicméně umožnily srazit roztažení mezer v zásadě o řád. Při srovnání šestých řádků (vlevo a vpravo dole) vidíme, že stejný materiál vysazený s mezerami dosti roztaženými (34 %) je nahrazen variantou s dokonce poněkud zúženými (-1%). Toto paradoxní řešení bylo zvoleno proto, že výběr nejpříhodnější z 11 možných šířkových variant zanechal na řádku méně místa, než by bylo zapotřebí pro mezery ideální. Zvýšením počtu variant písma by bylo možné snížit míru těchto „zbytečně“ neideálních mezer.

Šestý a sedmý řádek taktéž demonstrují sousedství o 9 % rozdílných písem, čili jednoho z hraničních míst sazby. Soustředěný pohled na svisle sousedící „m“ ukazuje, že rozdíly v kresbě jsou dobře patrné.

Srovnání posledních třetin odstavce se zdá býti zřetelným vítězstvím nového systému. Zde autorce ani vrozená skepse nezabránila pochválit pravidelnou šedost i kompaktnější závěr s rozumnější délkou východového řádku.

5	18%	Norská runová jména
1	11%	jsou pozdější, z doby, kdy
8	-14%	bylo ve Skandinávii použí-
10000	-33%	váno už pouze 16 run, takže
10000	-33%	kompletní seznam jmen run
20	29%	této oblasti nemáme. Ná-
2	-9%	zvy, které runám daly jiné
10000	-33%	germánské národy, neznáme
10000	-33%	vůbec (ačkoliv některá pís-
9	-14%	mena gótské abecedy mají
10000	-33%	k jménům run jistý vztah).
7	20%	Ze 16 <i>přeživších</i> norských
15	26%	run jich většina odpovídá
10000	-33%	jejich anglosaským protějškům;
28	32%	a tuto podmnožinu pova-
87	47%	žujeme za runy nejstarší,
8	-14%	pocházející z dávných ger-
-	8%	mánských dob.

Norská runová jména jsou pozdější, z doby, kdy bylo ve Skandinávii používáno už pouze 16 run, takže kompletní seznam jmen run této oblasti nemáme. Názvy, které runám daly jiné germánské národy, neznáme vůbec (ačkoliv některá písmena gótské abecedy mají k jménům run jistý vztah). Ze 16 *přeživších* norských run jich většina odpovídá jejich anglosaským protějškům; a tuto podmnožinu považujeme za runy nejstarší, pocházející z dávných germánských dob.

5	18%	Norská runová jména	+1	5%	Norská runová jména
1	11%	jsou pozdější, z doby, kdy	+1	4%	jsou pozdější, z doby, kdy
154	57%	bylo ve Skandinávii pou-	+5	8%	bylo ve Skandinávii pou-
329	74%	žíváno už pouze 16 run,	+5	40%	žíváno už pouze 16 run,
2005	135%	takže kompletní seznam	+5	62%	takže kompletní seznam
32	34%	jmen run této oblasti ne-	+5	-1%	jmen run této oblasti ne-
10000	-33%	máme. Názvy, které runám	-4	-3%	máme. Názvy, které runám
768	98%	daly jiné germánské ná-	+5	52%	daly jiné germánské ná-
5	18%	rody, neznáme vůbec (ač-	+2	-1%	rody, neznáme vůbec (ač-
35	-23%	koliv některá písmena gót-	-2	-1%	koliv některá písmena gót-
169	98%	ské abecedy mají k jmé-	+5	25%	ské abecedy mají k jmé-
72	44%	nům run jistý vztah). Ze	+5	9%	nům run jistý vztah). Ze
10000	238%	16 <i>přeživších</i> norských	-2	-2%	16 <i>přeživších</i> norských
4391	176%	run jich většina odpo-	-4	-9%	jich většina odpovídá jejich
3029	155%	vídá jejich anglosaským	+3	13%	anglosaským protějškům;
536	87%	protějškům; a tuto pod-	+3	3%	a tuto podmnožinu pova-
2884	153%	množinu považujeme za	+5	-1%	žujeme za runy nejstarší,
0	0%	runy nejstarší, pocházející	-1	-3%	pocházející z dávných ger-
2591	147%	z dávných germánských	0	0%	mánských dob.
-	-	dob.			

Vlevo nahoře: jak zalomil plain. Vlevo dole: přidaná `\emergencystretch1em`. Vpravo nahoře: sazba na praporek (ideální mezery, `\rightskip` plus minus 5%, `\emergencystretch1em`). Vpravo dole: úprava zlomu na praporek za pomoci deformovaných fontů.

10000	-33%	Ani při návodu nemůžeme od-			Ani při návodu nemůžeme
10000	-33%	dělovat to, co je správné, od toho,			oddělovat to, co je správné,
12	-16%	co je pouze zdánlivě správné, po-			od toho, co je pouze zdánlivě
10000	-33%	něvadž právě to není sporným stra-			správné, poněvadž právě to není
14	-17%	nám nikdy předem známo. Proto			sporným stranám nikdy pře-
29	33%	zde uvádím úskoky bez ohledu			dem známo. Proto zde uvádím
10000	-33%	na objektivní pravdu či nepravdu,			úskoky bez ohledu na objektivní
9	-14%	neboť to člověk sám nemůže bez-			pravdu či nepravdu, neboť to člo-
15	26%	pečně vědět. Teprve sporem má			věk sám nemůže bezpečně vědět.
143	56%	být pravda zjištěna. A pak při			Teprve sporem má být pravda
86	47%	každé debatě nebo argumentaci			zjištěna. A pak při každé debatě
10000	-33%	vůbec se musíme shodnout na ně-			nebo argumentaci vůbec se mu-
175	60%	čem, odkud – jakožto od prin-			síme shodnout na něčem, odkud
0	1%	cipu – hodláme otázku, o kterou			– jakožto od principu – hodláme
111	51%	jde, zkoumat: <i>Contra negantem</i>			otázku, o kterou jde, zkoumat:
190	64%	<i>principia non est disputandum.</i>			<i>Contra negantem principia non</i>
0	1%	(Nechť se nediskutuje s tím, kdo			<i>est disputandum.</i> (Nechť se nedis-
10000	-33%	popírá platnost základních pojmů			kutuje s tím, kdo popírá platnost
-	0%	a vět.)			základních pojmů a vět.)
273	69%	Ani při návodu nemůžeme	+5	17%	Ani při návodu nemůžeme
80	46%	oddělovat to, co je správné, od	+5	124%	oddělovat to, co je správné,
10000	-33%	toho, co je pouze zdánlivě správné,	+5	56%	od toho, co je pouze zdánlivě
0	-4%	poněvadž právě to není sporným	+1	-3%	správné, poněvadž právě to není
552	88%	stranám nikdy předem známo.	+5	114%	sporným stranám nikdy pře-
219	65%	Proto zde uvádím úskoky bez	+5	19%	dem známo. Proto zde uvádím
145	56%	ohledu na objektivní pravdu či	-3	-12%	úskoky bez ohledu na objektivní
175	60%	nepravdu, neboť to člověk sám	-3	-4%	pravdu či nepravdu, neboť to člo-
72	44%	dem nemůže bezpečně vědět. Teprve	-1	2%	věk sám nemůže bezpečně vědět.
14	26%	sporem má být pravda zjištěna.	+5	27%	Teprve sporem má být pravda
725	96%	A pak při každé debatě nebo	+1	0%	zjištěna. A pak při každé debatě
1248	116%	argumentaci vůbec se musíme	+3	0%	nebo argumentaci vůbec se mu-
1342	118%	shodnout na něčem, odkud –	+1	1%	síme shodnout na něčem, odkud
218	64%	jakožto od principu – hodláme	+0	2%	– jakožto od principu – hodláme
179	60%	otázku, o kterou jde, zkoumat:	+5	15%	otázku, o kterou jde, zkoumat:
133	57%	<i>Contra negantem principia non</i>	+5	-11%	<i>Contra negantem principia non</i>
124	56%	<i>est disputandum.</i> (Nechť se ne-	-3	-13%	<i>est disputandum.</i> (Nechť se nedis-
0	-3%	diskutuje s tím, kdo popírá plat-	-2	-4%	kutuje s tím, kdo popírá platnost
-	0%	nost základních pojmů a vět.)	0	0%	základních pojmů a vět.)

Příklad druhý

Vlevo nahoře: plain. Vlevo dole: přidaná `\emergencystretch`1em. Vpravo nahoře: sazba na praporek. Vpravo dole: deformované fonty. \TeX ovské řešení přinaší 6 řádků s `badness` 10 000. Varianta s praporkem ukazuje, že úvodní řádky odstavce je možné zalomit pouze velmi krátké (za pomoci `\emergencystretch`). A skutečně, i po přesazení zůstávají na druhém řádku skutečně velmi široké (124%) mezery. Útěchou může být snad jen pohled vlevo, kde \TeX samotný nedovedl odstavec rozpálit vůbec.

25	-20%	Eristická dialektika je umění
9	-15%	diskutovat, a sice tak diskutovat,
10000	-33%	aby člověk vždy získal pravdu, tedy
0	6%	<i>per fas et nefas</i> . Lze totiž mít ve
87	47%	věci samé pravdu objektivně, a
10000	-33%	přece se člověk v očích posluchačů,
143	56%	ba leckdy i ve svých vlastních,
10000	-33%	ocitne v nepravu – tehdy, vyvrátí-
147	56%	li odpůrce můj důkaz a platí-li
26	31%	toto vyvrácení již také jako vy-
19	-19%	vrácení tvrzení samého, jež přece
0	-4%	lze dokazovat ještě jinak; v tako-
1	-7%	vém případě je ovšem poměr pro
1	10%	odpůrce opačný: získá vrch, jak-
5	-12%	koli je objektivně v nepravu. Jak
-	0%	je to možné?

25	-20%	Eristická dialektika je umění
9	-15%	diskutovat, a sice tak diskutovat,
259	68%	aby člověk vždy získal pravdu,
66	43%	tedy <i>per fas et nefas</i> . Lze totiž
21	29%	mít ve věci samé pravdu objek-
37	36%	tivně, a přece se člověk v očích
0	3%	posluchačů, ba leckdy i ve svých
1199	114%	vlastních, ocitne v nepravu –
2150	139%	tehdy, vyvrátí-li odpůrce můj
341	75%	důkaz a platí-li toto vyvrácení
338	75%	již také jako vyvrácení tvrzení
29	33%	samého, jež přece lze dokazovat
364	77%	ještě jinak; v takovém případě
1960	134%	je ovšem poměr pro odpůrce
1478	122%	opačný: získá vrch, jakkoli je
17	27%	objektivně v nepravu. Jak je to
-	-	možné?

Eristická dialektika je umění diskutovat, a sice tak diskutovat, aby člověk vždy získal pravdu, tedy *per fas et nefas*. Lze totiž mít ve věci samé pravdu objektivně, a přece se člověk v očích posluchačů, ba leckdy i ve svých vlastních, ocitne v nepravu – tehdy, vyvrátí-li odpůrce můj důkaz a platí-li toto vyvrácení již také jako vyvrácení tvrzení samého, jež přece lze dokazovat ještě jinak; v takovém případě je ovšem poměr pro odpůrce opačný: získá vrch, jakkoli je objektivně v nepravu. Jak je to možné?

-2	2%	Eristická dialektika je umění
-1	-4%	diskutovat, a sice tak diskutovat,
+5	23%	aby člověk vždy získal pravdu,
+5	17%	tedy <i>per fas et nefas</i> . Lze totiž
+4	1%	mít ve věci samé pravdu objek-
+5	7%	tivně, a přece se člověk v očích
+0	3%	posluchačů, ba leckdy i ve svých
+5	72%	vlastních, ocitne v nepravu –
+5	79%	tehdy, vyvrátí-li odpůrce můj
+5	31%	důkaz a platí-li toto vyvrácení
+5	30%	již také jako vyvrácení tvrzení
+3	5%	samého, jež přece lze dokazovat
+5	32%	ještě jinak; v takovém případě
+5	93%	je ovšem poměr pro odpůrce
+5	80%	opačný: získá vrch, jakkoli je
+3	6%	objektivně v nepravu. Jak je to
0	0	možné?

Příklad třetí

Vlevo nahoře: plain. Vlevo dole: přidaná `\emergencystretch`1em. Vpravo nahoře: sazba na praporek. Vpravo dole: deformované fonty. Tato ukázka demonstruje patrně typický způsob užití systému: `TEX` se za pomoci `\emergencystretch` podařilo zalomit, ale možných odstavců je tak málo, že ani volnost přidaná zavedením praporku nijak nezměnila vybraný zlom. Variantní šířkou písma tedy pouze ‚dorovnávané‘ mezery, respektive snažíme se poněkud zmírnit velmi řídké řádky. Vzhledem k hornímu omezení deformace písma (5%) zůstanou šířky mezer ovšem stále ‚nepřijatelné‘ (pro srovnání viz ideální mezery v sazbě na praporek). Kladem tohoto řešení je naopak fakt, že drtivá většina řádků má obdobný typ deformace – dosti rozšířené písmo. Nenastávají tedy tak značné problémy s kompatibilitou jako jinde.

10000	-33%	Tvůj příklad Llanfairpwllgwyn-				Tvůj příklad Llanfairpwll-
10000	-	gyllogerychwyndrobwlllantysil-				gwyngyllgogerychwyndrobwll-
157	58%	iogogogoch (čili Llanfairu P.G.,				llantysiliogogogoch (čili Llanfairu
200	63%	jak se prý běžně zkracuje toto				P.G., jak se prý běžně zkracuje
1	-7%	město ve Walesu) je přece jenom				toto město ve Walesu) je přece
84	47%	okrajový. Němčina taky nestojí				jenom okrajový. Němčina taky
10	23%	a nepadá s tím, že se v ní ,pro-				nestojí a nepadá s tím, že se v ní
85	47%	středí pro vývoj aplikací‘ řekne				,prostředí pro vývoj aplikací‘
10000	-	,Anwendungsentwicklungsumgebung‘.				řekne ,Anwendungsentwicklungs-
						umgebung‘.
1342	118%	Tvůj příklad Llanfairpwll-	+5	38%		Tvůj příklad Llanfairpwll-
10000	-	gwyngyllgogerychwyndrobwll-	+5.52	0%		gwyngyllgogerychwyndrobwll-
7030	206%	llantysiliogogogoch (čili Llan-	-2	-5%		llantysiliogogogoch (čili Llan-
1831	131%	fairu P.G., jak se prý běžně	+5	1%		P.G., jak se prý běžně zkracuje
40	37%	zkracuje toto město ve Walesu)	+5	9%		toto město ve Walesu) je přece
159	58%	je přece jenom okrajový. Něm-	+5	13%		jenom okrajový. Němčina taky
3	-9%	čina taky nestojí a nepadá s tím,	-2	0%		nestojí a nepadá s tím, že se v ní
132	54%	že se v ní ,prostředí pro vývoj	+5	102%		,prostředí pro vývoj aplikací‘
1	12%	aplikací‘ řekne ,Anwendungsent-	-2	17%		řekne ,Anwendungsentwicklungs-
-	-	wicklungsumgebung‘.	0	0%		umgebung‘.

Příklad čtvrtý

Vlevo nahoře: jak zalomil plain. Vlevo dole: přidaná `\emergencystretch1em`. Vpravo nahoře: sazba na praporek (ideální mezery, `\rightskip` plus minus 5%, `\emergencystretch1em`). Vpravo dole: úprava zlomu na praporek za pomoci deformovaných fontů. Setká-li se \TeX se skutečně neřešitelným problémem v podobě velmi dlouhých slov (při začátku odstavce stačí i slova kratší než $2 \times \text{\hspace{1em}}$), nepomůže mu samozřejmě k zarovnanému okraji ani značná hodnota `\emergencystretch`. Přidané glue při zlomu uváže a vysadí ho na pravý okraj textu (viz druhý řádek vlevo dole). Dokonce i slova teoreticky snesitelně dlouhá způsobí často extrémní problémy – viz 206procentní mezery na třetím řádku. Přelámanému řešení lze leccos vytknout, ale ze srovnání vychází velmi dobře. Obtížný druhý řádek je vyřešen přípravou fontu s přesně požadovanou šířkou. Tady dokonce vyšla větší než je zvolená pětiprocentní hranice – při bližším zkoumání dotyčného řádku a okolí však překvapivě plyne, že sousedství fontů lišících se o 7,52 % nepůsobí v tomto případě nijak zvlášť nepříjemně.

8	21%	Ani při návodu nemůžeme oddělovat to,
24	31%	co je správné, od toho, co je pouze zdánlivě
10000	-33%	správné, poněvadž právě to není sporným stranám nikdy předem známo. Proto zde uvádím
0	7%	úskoky bez ohledu na objektivní pravdu či
61	42%	nepravdu, neboť to člověk sám nemůže bezpečně vědět. Teprve sporem má být pravda zjištěna. A pak při každé debatě nebo argumentaci vůbec se musíme shodnout na něčem, odkud – jakožto od principu – hodláme otázku, o kterou jde, zkoumat: <i>Contra negantem principia non est disputandum</i> . (Nechť se nediskutuje s tím, kdo popírá platnost základních pojmů a vět.)
8	-14%	
10000	-33%	
9	-15%	
42	41%	
7	20%	
-	0%	

8	21%	Ani při návodu nemůžeme oddělovat to,
334	74%	co je správné, od toho, co je pouze zdánlivě správné, poněvadž právě to není sporným stranám nikdy předem známo. Proto zde uvádím úskoky bez ohledu na objektivní pravdu či nepravdu, neboť to člověk sám nemůže bezpečně vědět. Teprve sporem má být pravda zjištěna. A pak při každé debatě nebo argumentaci vůbec se musíme shodnout na něčem, odkud – jakožto od principu – hodláme otázku, o kterou jde, zkoumat: <i>Contra negantem principia non est disputandum</i> . (Nechť se nediskutuje s tím, kdo popírá platnost základních pojmů a vět.)
275	70%	
621	91%	
480	84%	
533	87%	
1248	116%	
1635	127%	
1616	126%	
1005	108%	
58	41%	
1	9%	
73	48%	
5	19%	
-	0%	

+5	84%	Ani při návodu nemůžeme oddělovat to, co je správné, od toho, co je pouze zdánlivě správné, poněvadž právě to není sporným stranám nikdy předem známo. Proto zde uvádím úskoky bez ohledu na objektivní pravdu či nepravdu, neboť to člověk sám nemůže bezpečně vědět. Teprve sporem má být pravda zjištěna. A pak při každé debatě nebo argumentaci vůbec se musíme shodnout na něčem, odkud – jakožto od principu – hodláme otázku, o kterou jde, zkoumat: <i>Contra negantem principia non est disputandum</i> . (Nechť se nediskutuje s tím, kdo popírá platnost základních pojmů a vět.)
+3	0%	
+5	29%	
+5	42%	
+5	44%	
+5	47%	
+5	67%	
+1	2%	
+5	22%	
+0	-2%	
+5	14%	
+5	96%	
+5	27%	
+5	96%	
0	0%	



Příklad pátý

Vlevo nahoře: jak zalomil plain. Vpravo nahoře: větší tolerance a `\looseness1`. Vlevo dole: úprava zlomu na praporek pomocí deformovaných fontů. Vpravo dole: meziřící.

Variantní šířky fontů lze využít kromě zlepšení děravé sazby úzkých sloupců i v řadě dalších typografických záměrů. Tato ukázka demonstruje snahu vysadit odstavec rozumné šířky a délky o řádek delší (budiž to zapotřebí kvůli nějakému vyššímu výtvarnému záměru). \TeX takové řešení najde, ovšem za cenu zvýšení tolerance z 200 na 1635. Výsledkem je navíc úžasné ‚meziřící‘.

Naše řešení tyto nepřijemné dopady zmírňuje. Obdobným způsobem lze variantní šířky fontů použít při zlepšování odstavce, který musí být vysazen s určitou hodnotou `\parfillskip`. V takových případech je děravá sazba častým jevem i u odstavců s poměrně širokými řádky.

Symposium o tolerantnosti

V hloubi šedesátých let, kdy se na české půdě začala do úvah a rozhovorů vracet některá zakázaná nebo zapomenutá té-

Symposium o tolerantnosti

V hloubi šedesátých let, kdy se na české půdě začala do úvah a rozhovorů vracet některá zakázaná nebo zapomenutá té-

Symposium o tolerantnosti

V hloubi šedesátých let, kdy se na české půdě začala do úvah a rozhovorů vracet některá zakázaná nebo zapomenutá té-

Symposium o tolerantnosti

V hloubi šedesátých let, kdy se na české půdě začala do úvah a rozhovorů vracet některá zakázaná nebo zapomenutá té-

Symposium o tolerantnosti

V hloubi šedesátých let, kdy se na české půdě začala do úvah a rozhovorů vracet některá zakázaná nebo zapomenutá té-

Příklad šestý

První čtyři ukázky: $\text{T}_{\text{E}}\text{X}$. Pátá: nadpis užším řezem.

Sazba nadpisu daného znění a dané velikosti na danou šířku přináší občas problémy. Zúžit font o tři procenta je řešením zdaleka nejelegantnějším.

Na závěr série ukázek ještě dodejme některé obecné soudy. Problémová jsou místa, kde se stýkají řádky se značnými rozdíly v typu deformace řezu písma. Takto úzké a krátké odstavce ovšem není možno zalomit tolika způsoby, aby z nich bylo možno vybrat řešení s podstatně kompatibilnějšími sousedními řádky – zvýšením hodnoty `\adjdemerits` se pouze zvyšují celkové demerits odstavců, ale k pronikavému zlepšení nedochází. Podstatně lepší situace je v tomto ohledu u zlepšování vzhledu odstavců násilně roztažených (kladná `\looseness`, menší `\parfillskip`), kde metoda typicky pouze ‚stahuje mezery‘ a použito je téměř všude roztažené písmo.

K estetickému hodnocení ukázek snad jen jedinou poznámku: laici podrobení testům většinou ‚nic nevidí‘ (chápu, že tento výsledek je možno interpretovat dvojznačně). Naproti tomu u osob zběhlejších v pozorování mikrotypografických jevů žádost o explicitní označení míst vnímaných při čtení nepřijemně (bez znalosti číselných charakteristik) vede k podpoření dojmu, že každý čtenář reaguje zcela rozdílně.

Závěrem

Především mi dovoluňte se omluvit za patrně trochu příliš motivačních poznámek v úvodní části tohoto textu. Tento článek je poslední tečkou za částí obhájené diplomky a byl psán se záměrem vzbudit ve zdatných T_EX-programátorech dojem, že variantní řezy písma jsou zajímavý nástroj, který by se hodilo mít k dispozici. Kdokoliv by měl zájem podniknout experimenty vlastní, ať už pro inspiraci při řešení obtížného dokumentu nebo při hledání námětu na programovací projekt, může využít skriptů a maker komentovaných a vystavených na <http://www.fi.muni.cz/~imladris/vlw>.

Veškeré úpravy, zlepšení či dokonce komplexní přepracování nadhozené ideje jistě uvítá přinejmenším ta část T_EXistů, kteří (stejně jako já) mají rádi netradiční hrátky s typografií.

Odkazy

- [1] Martin Davies: *The Gutenberg Bible*. The British Library Board, 1996.
- [2] Oldřich Hlavsa: *Typographia 1–3*, 1976–1986,
- [3] Oldřich Menhart: *Tvorba typografického písma*. SPN, 1957.
- [4] Philip Taylor: *Electronic Typesetting and T_EX: Book Design for T_EX Users*. Sborník zvaných přednášek SOFSEM '93, ÚVT MU Brno, 1993.
- [5] Arthur Schopenhauer: *Eristická dialektika čili Umění dostat v každé debatě za pravdu*. Zvláštní vydání, 1994.
- [6] Philip Taylor: *Pragmatický přístup k odstavcům*. Zpravodaj Československého sdružení uživatelů T_EXu, **4** (3), 110–116 (1994).
- [7] URW Software Hamburg: *hz-program: Micro-typography for advanced typesetting*, 1993.
- [8] Adolf Wild: *La typographie de la Bible de Gutenberg* Cahiers Gutenberg, Septembre, 1995.

Tento článek je mírně rozšířenou verzí příspěvku na konferenci TUG'98. Některé sázené ukázky pocházejí z knihy [5].

Mirka Misáková
FI MU, Botanická 68a, Brno
imladris@fi.muni.cz

V tomto článku se seznámíme se základními informacemi o fontech ve formátu TrueType a jejich použití v programu pdf \TeX , který umožňuje produkovat alternativní výstup programu \TeX ve formátu PDF. Seznámíme se se základním postupem nutným k použití těchto fontů i pro české texty a ukážeme si, kde najít kvalitní fonty v tomto formátu.

Formát PDF (*Portable Document Format*) podle specifikace, kterou naleznete na adrese <http://www.adobe.com/supportservice/devrelations/PDFS/TN/PDFSPEC.PDF>, umožňuje mimo jiné použití fontů ve formátu Type 1 (i instanci fontu Multiple Master Type 1), Type 3, Type 0 a TrueType. Dokument ve formátu PDF může také obsahovat pouze určitou (většinou používanou) část fontu – takový font nazýváme *redukovaný* (anglicky *font subset*). A právě fonty ve formátu TrueType se budeme zabývat v tomto článku.

Formát TrueType původně navrhla společnost Apple Computer (<http://www.apple.com>) za účelem odstranění nutnosti platit vysoké licenční poplatky majitelům jiných technologií návrhu písem a také jako řešení některých principiálních omezení písem ve formátu Type 1. Prvním operačním systémem, který plně podporoval fonty TrueType, byl System 7 pro počítače Macintosh v květnu roku 1990. Společnost Microsoft, která měla pro vývoj nového formátu písma podobné důvody jako společnost Apple Computer (obávala se, že svěří nejdůležitější část svého operačního systému do cizích rukou a nebude na ní mít vliv), implementovala podporu TrueType fontů ve svém operačním systému Windows 3.1 v dubnu roku 1991. Od té doby se formát TrueType stal nejpoužívanějším formátem pro popis obrysových (*outline* fontů). Formát TrueType je hojně používán v operačních systémech společnosti Microsoft i nyní. Např. operační systém Microsoft Windows NT Server verze 4.0 obsahuje 39 předinstalovaných fontů v tomto formátu (např. Arial, Times, Courier, Verdana apod.) a navíc spousta (a mnohdy i kvalitních) TrueType fontů je k dispozici na síti Internet a mnoho uznávaných tvůrců písma již nabízí svá díla i v podobě TrueType fontů (např. Bitstream – <http://www.bitstream.com>, ITC – <http://www.itcfonts.com> apod.). Proto by bylo škoda, kdyby neexistovala možnost využití těchto fontů v sázecím systému \TeX .

Specifikace formátu TrueType verze 1.0 je k dispozici na serveru společnosti Microsoft <http://www.microsoft.com> pod názvem *TrueType 1.0 Font Files* (bohužel ve formátu čitelném pouze programem Microsoft Word, a proto při-

vlastek Open, který je ve specifikaci hodně používán, ztrácí svůj smysl). Specifikace je poměrně rozsáhlá (384 stran), ale přehledně a srozumitelně vysvětluje principy a specifika fontů ve formátu TrueType.

Ale nyní již k vlastnímu použití TrueType fontů v našich dokumentech. Pdf \TeX , stejně jako \TeX , potřebuje ke každému fontu metrické informace, které jsou uloženy v souboru *název_fontu.tfm*. Pokud tedy chceme použít např. font s názvem *arial*:

```
\font\arial=arial\arial Text napsaný fontem Arial
```

musíme pdf \TeX u nějakým způsobem vytvořit soubor *arial.tfm*, který tyto metrické informace obsahuje. Existuje několik způsobů, jak toho dosáhnout (ten nejpřímější, tedy spuštění oblíbeného editoru a editaci souboru *arial.tfm* popisovat nebudu, je poněkud obtížný...). Tento problém ale vyřešil sám autor pdf \TeX u tím, že vytvořil program *ttf2afm*, který nám náš úkol velmi zjednoduší. Pomocí tohoto programu vytvoříme soubor metrik ve formátu AFM (*Adobe Font Metrics*), který pomocí programu *afm2tfm* převedeme do finální podoby vhodné pro \TeX .

Ale to bychom předbíhali, začněme postupně. Rozhodneme se, že v našem dokumentu chceme používat font Arial, ale bohužel jej nemáme k dispozici ani ve zdrojové podobě (METAFONT) ani v jiných podporovaných formátech (PK, Type 1). Proto použijeme font Arial ve formátu TrueType. Tento font najdete v každé instalaci operačního systému Microsoft Windows v souboru s názvem *arial.ttf*.

Nyní tedy máme k dispozici ten správný font, ale chybí nám k němu základní metrické informace, bez kterých \TeX nemůže provést správně sazbu. Musíme vyrobit soubor s těmito informacemi, aby jej \TeX mohl použít. Jak již bylo řečeno, právě za tímto účelem byl vytvořen program *ttf2afm*. Jeho použití je následující:

```
ttf2afm název_fontu [ encoding_vector ]
```

Použití je tedy zcela zřejmé. Pokud neuvedeme volitelný parametr (*encoding_vector*), jsou všechny znaky mapovány na kód /.notdef a v AFM souboru je jejich pořadové číslo (značka C nebo CX v souboru AFM) rovno -1. V našem případě tedy provedeme příkaz:

```
ttf2afm arial.ttf 8r.enc >arial.afm
```

kterým získáme AFM soubor *arial.afm*, v němž jsou znaky kódovány podle kódovacího vektoru TeXBase1Encoding, který je definován v souboru *8r.enc* distribuovaném v balíku *dvips*. Tento soubor musíme mít v adresáři s naším fontem nebo musíme zadat jeho plnou cestu.

Samozřejmě můžeme použít i jiné kódovací vektory (např. *8a.enc* – Adobe Standard Encoding), ale připravíme se tím o možnost používání některých znaků.

Ze souboru `arial.afm` můžeme nyní vyčíst poměrně zajímavé informace. Např.: jak se font jmenuje (jak `FamilyName`, tak i `FontName`), jaké znaky obsahuje, kdo je jeho autorem (`Notice`), metrické informace jednotlivých znaků (`bounding box`) a kerningové páry. Vytvořený AFM soubor může vypadat např. takto:

```
Comment Converted at Sun Jul 5 23:50:29 1998 by ttf2afm ...
StartFontMetrics 2.0
FontName ArialMT
FullName Arial
Notice Typeface (c) The Monotype Corporation ...
ItalicAngle 0
IsFixedPitch false
FontBBox -709 -324 1071 1037
UnderlinePosition -217
UnderlineThickness 150
CapHeight 1466
XHeight 1062
Ascender 1854
Descender -434

StartCharMetrics 227
C 1 ; WX 333 ; N dotaccent ; B 111 583 220 692 ;
...
EndCharMetrics

StartKernData
StartKernPairs 104
KPX space A -55
...
EndKernPairs
EndKernData
EndFontMetrics
```

Nyní musíme vytvořit soubor s metrickými informacemi pro \TeX , tedy soubor s příponou `.tfm`. Ale tento úkol nám opět ulehčí jeden program, tentokrát je to `afm2tfm` z balíku `dvips`.

```
afm2tfm arial.afm -T 8r.enc
```

Tímto příkazem vyrobíme soubor `arial.tf`, který je již vhodný pro použití \TeX em. Po skončení tohoto příkazu se vypíše řádek, který nám bude sloužit jako vodičko pro zápis nového fontu do fontové mapy (podobně jako u `dvips`

do souboru `psfonts.map`). V našem případě na výstupu obdržíme následující řádek:

```
arial ArialMT " TeXBase1Encoding ReEncodeFont " <8r.enc
```

První údaj nám říká, že metrické informace jsou v souboru `arial.tfm`. Druhý údaj (`ArialMT`) je postscriptové jméno fontu. Poslední údaj nám říká, v jakém kódování jsou znaky ve fontu a v souboru metrických informací kódovány. Tyto údaje jsou velmi důležité, protože podle nich budeme konstruovat nový záznam do tabulky fontů.

PdfTeX má oproti klasickému TeXu jeden výrazný rozdíl. Pokud ve zdrojovém textu použijeme nějaký font, tak TeX umístí do výsledného DVI souboru pouze odkaz na daný font a je záležitostí DVI ovladače, jak si s tím poradí. PdfTeX to má poněkud složitější, protože formát PDF (už jen z názvu) by měl být přenositelný, a právě proto musí (přesněji – měl by) obsahovat definici všech použitých fontů (výjimkou je 14 vestavěných fontů v každém prohlížeči tohoto formátu). Proto musí pdfTeX používat nějaký mechanismus „hledání fontů“. Autor pdfTeXu adaptoval mechanismus podobný DVI ovladačům a používá zvláštní mapovací soubor k převodu jmen fontů na jejich fyzické umístění a kódování. Soubor se typicky jmenuje `pdftex.map` nebo `tetex.map` a záznam pro náš případ fontu Arial bude vypadat následovně:

```
arial ArialMT <arial.ttf <8r.enc
```

Význam všech čtyř výrazů je zřejmý, jde po řadě o jméno souboru obsahujícího metrické informace, postscriptové jméno fontu, fyzické umístění fontu na disku a kódování použité při výrobě souboru TFM.

Pokud jsme vše udělali podle návodu, můžeme např. po přeložení následujícího zdrojového textu pdfTeXem

```
\nopagenumbers
\font\arial=arial at 3cm
\arial
Font Arial
\bye
```

obdržet například následující výstup:

Font Arial

Umíme tedy použít font ve formátu TrueType v pdfTeXu. Jistě vás ale napadne, jak to bude s háčky a čárkami. Ano, to je správná otázka. Samozřejmě,

že existuje způsob, jak vyrobit i česká písmenka z fontů ve formátu TrueType. Podobně jako při počešťování fontů ve formátu Type 1 použijeme balík `a2ac` a skript `mkfnt`, který nám počeštění fontů ve formátu TrueType velmi usnadní. Onen magický příkaz zní:

```
mkfnt arial arial 8z
```

Samozřejmě musíme mít nejprve správně nainstalován balík `a2ac`. Tento příkaz nejprve spustí program `afm2tfm`, který vytvoří soubor `rarial.tfm` s metrickými informacemi o jednotlivých znacích fontu a také vytvoří soubor `arial8z.vpl`, ze kterého potom v druhém kroku vytvoří virtuální font `arial8z.vf`. Nyní již zbývá pouze doplnit záznam o souboru `rarial.tfm` do fontové mapy. Doplňme tedy řádek

```
rarial ArialMT <arial.ttf <8r.enc
```

do souboru `tetex.map`.

A nyní je již vše připraveno pro použití fontu Arial i v českých textech. Např.:

Počeštěný Arial. ěščřžýáíé

Následující obrázek ukazuje několik dalších fontů ve formátu TrueType z operačního systému Microsoft Windows NT.

Font ArialMT - ěščřžýáíé

Font CourierNewPSMT - ěščřžýáíé

Font TimesNewRomanPSMT - ěščřžýáíé

Font Verdana - ěščřžýáíé

Není ale všechno zlato, co se třpytí. Samozřejmě při použití TrueType fontů v pdf \TeX u můžeme narazit na některé věci, které při použití \TeX u pokládáme za samozřejmé – např. ligatury písmen `f` a `i` (na obrázku vlevo je sazba fontem Arial a vpravo Type 1 fontem):

fi fi

Další problémy mohou vzniknout například při sazbě dlouhé pomlčky (em-dash) nebo při sazbě makra pro název T_EXu.

1 -- 2 1 – 2
T_EX T_EX

Ale i přesto jsou TrueType fonty použitelné a mohou nám pomoci v případě „fontové nouze“.

Na závěr jsem si ponechal několik odkazů na zajímavá místa plná relevantních informací k tematice fontů ve formátu TrueType:

Microsoft Typography: <http://www.microsoft.com/typography/>

TrueType Font File specification: http://www.microsoft.com/typography/tt/tt_open/msdn/ttospec.zip

TrueType Typography: <http://www.truetype.demon.co.uk/>

Kvalitní fonty ve formátu TrueType můžete zakoupit, případně zdarma získat na následujících serverech:

Free TrueType Fonts: <http://www.microsoft.com/typography/fontpack/default.htm>

Monotype Typography: <http://www.monotype.com/>

The Font Bureau, Inc.: <http://www.fontbureau.com>

Linotype Library: <http://www.linotypelibrary.com>

Pavel Janík ml.
Pavel.Janik@inet.cz

Na jaře roku 1997 uvedla firma DTP Studio na trh zajímavý CD-ROM *CE Písma, loga, symboly* (dále PLS). Po půl roce pak vydala i rozšířenou verzi PLS II, se kterou jsem měl možnost se blíže seznámit, díky tomu, že firma laskavě zapůjčila předsedovi ČSTUGu Petru Sojkovi jedno CD k vyzkoušení. V následujícím textu si tedy dovoluji předložit výsledek svého testování, jehož hlavním smyslem bylo od samého počátku především ověření použitelnosti nabízených PostScriptových písem v \TeX u.

Co všechno na CD-ROMu naleznete

Jak už název napovídá, hlavním obsahem cédéčka PLS II je datová část, členěná do tří podskupin:

- *CE písma* – 257 počestěných¹ fontů ve formátech Adobe Type 1 a TrueType. Součástí CD je kompletní tisknutelný katalog všech písem.
- *Loga* – přes 500 katalogizovaných logotypů významných českých i zahraničních firem. Loga jsou ve vektorovém formátu AI (Adobe Illustrator).
- *Symboly* – zhruba 80 rozmanitých symbolových a ornamentálních sad ve formátu TrueType, včetně azbuky, symbolů pro sazbu jízdních řádů apod.

Na stříbrném disku dále naleznete testovací 30denní verzi programu Corel Draw 7 (DTP Studio v sobě nezapře autorizovaného distributora produktů firmy Corel Corporation).

Po cédéčku se můžete pohodlně procházet použitím prohlížeče HTML souborů. Projevem běžné slušnosti je zařazení „free“ prohlížeče (zde MS Internet Explorer 3.02 CZ) pro ty, kdo jej ještě nemají.

Přestože si myslím, že všechny uvedené části mohou být pro mnohé uživatele zajímavé, soustředím se ve zbylém textu již pouze na CE písma ve formátu Type 1.

CE písma ve formátu Type 1

PLS II obsahuje celkem 257 řezů CE písem. Ke každému řezu je dostupný odpovídající PostScriptový popis fontu *.pfb plus metrika *.pfm. Jako základ pro

¹Termínem „počestění“ v tomto textu míním přidání akcentovaných znaků české, slovenské a některých dalších abeced. Přesnější výraz „postředovychodoevpřstění“ jsem zamítl, aby si laskavý čtenář nezlámal jazyk.

**A B C D E F G
H I J K L M N O P Q R
S T U V W X Y Z
Á Ä Č Ď É Ě Í Ĺ Ľ Ń Ó Ô Ö
Ř Ŕ Š Ť Ú Ů Ü Ý Ž**

**a b c d e f g h i j k l m n o
p q r s t u v w x y z
á ä č ď é ě í Ĺ Ľ Ń ó ô ö ř ŕ
š ť ú ů ü ý ž**

1 2 3 4 5 6 7 8 9 0

počeštění posloužily fonty z volné distribuce firmy HiStroke. Rozšíření o akcentované znaky CE abeced provedl ing. Daniel Jenne z firmy DTP Studio (mj. spolupracoval i na počešťování písem z rodiny Computer Modern). Pro úplnost uvedu, že všechny fonty jsou kódovány dle normy CP 1250.

Bohužel, na disku nejsou metriky ve formátu AFM (Adobe Font Metrics) [2], ale pouze, jak jsem již uvedl, v redukovaném tvaru PFM (Printer Font Metrics), který je sice postačující pro implementaci fontů ve Windows, ale pro konverzi metrik do \TeX ovského formátu TFM není vhodný. PFM totiž obsahuje pouze podmnožinu informací z AFM, takže konverze PFM na AFM není triviální záležitostí. Jelikož jsem ale chtěl vyzkoušet použití písem v \TeX u, potřeboval jsem samozřejmě vyrobit metriky *.tfm. Zkusil jsem kontaktovat autora a požádal ho, zda by mi mohl poskytnout oněch 257 metrik ve formátu AFM, vhodném ke snadné konverzi. Rád bych zde vyzdvihl, že pan ing. Jenne projevil velké pochopení a prakticky obratem mi zaslal elektronickou poštou kompletní sadu AFM metrik, takže se mi po nějaké době podařilo všechny fonty zdárně implementovat.

Potřebné informace k přípravě fontů pro sazbu \TeX em je možné nalézt v [4], [5], [7] či [10].

Zjištění vady na kráse

Za nejzávažnější nedostatek považuji totální absenci kerningových dat v metrických souborech. Ihned po zjištění této nectnosti jsem adresoval dotaz autorovi, který mi sdělil, že originální fonty sice kerning obsahovaly, ale byl prý tak příšerný, že sazba fontem bez kerningu vypadala lépe než s kerningem. Proto prý nevyhovující kerning raději ze všech fontů odstranil. Ujistil mne ale, že na doplnění vyhovujících kerningových párů se usilovně pracuje a příští upgrade je zcela jistě bude obsahovat. Osobně sice nepovažuji za profesionální přístup, když se pouštějí na trh ještě nedodělané produkty, ale zřejmě zde sehrály svou roli komerční důvody (CD-ROM si jistě koupí řada lidí, kteří ani nevědí, co to kerning je) a dále fakt, že dodělat opravdu dobrý kerning pro 257 fontů musí nepochybně zabrat spoustu času.

Dalším nedostatkem, ale přece jen už snadněji stravitelným, je absence ligatur v souborech *.pfb.

Poslední závadou, která se ale dá snadno obejít, je chybné pojmenování akcentovaných znaků \acute{t} , \acute{d} , \acute{l} a \acute{L} . Uvedené znaky mají symbolické názvy /tquoteright, /dquoteright, /lquoteright a /Lquoteright místo obvyklých a dnes už i normalizovaných názvů s označením akcentu jako caron, tedy např. /tcaron atd. Pokud tedy máte připravenou kódovací tabulku, kterou chcete předložit jako parametrický soubor konvertoru `afm2tfm`, musíte si v ní správné názvy na chvíli opravit na nesprávné a konverze se zdaří, jinak uvedené čtyři

***A B C D E F G
H I J K L M N O P Q R
S T U V W X Y Z
Á Ä Č Ď É Ě Í Ĺ Ľ Ń Ó Ô Ö
Ř Ť Š ť Ú Ú Ü Ý Ž***

***a b c d e f g h i j k l m n o
p q r s t u v w x y z
á ä č ď é ě í ĺ ľ ů ó ô ö ř ř
š ť ú ů ü ý ž***

1 2 3 4 5 6 7 8 9 0

znaky nebudou ve výsledku dostupné. Všechny ostatní symbolické názvy znaků jsou běžné a nečinily problémy. I tuto drobnou závadu přislíbil autor opravit.

Co je možné pochválit

Z pohledu uživatele $\text{T}_{\text{E}}\text{X}$ u se mi jako největší pozitivum jeví fakt, že fonty obsahují „skutečné“ akcentované znaky. Jsou vytvořeny opravdu precizně a pokusná sazba ukáže, že typografická kvalita textu je vyšší, než jakou můžeme docílit metodou kompozitních znaků.²

Rozdíly se projeví u problematických písmen \acute{t} , \acute{d} (ve slovenštině také u písmena ł) uprostřed slov. V následujících ukázkách je vždy horní z dvojice shodných textů vysázen pomocí fontu s kompozitními znaky (New Century Schoolbook Bold) z balíku $\mathcal{C}_{\text{S}}\text{T}_{\text{E}}\text{X}$, dolní pak pomocí obdobného CE fontu z testované kolekce PLS II.

nešťastný
nešťastný

jako ďas
jako ďas

dršťková polévka
dršťková polévka

Ludĕku!
Ludĕku!

Tak ať!
Tak ať!

Koľko?
Koľko?

Vzbuďte se!
Vzbuďte se!

Jelikož nemusí nahrazovat háček apostrofem, vyhrávají v tomto srovnání CE písma z PLS II. U fontů s kompozitními znaky je akcent vidět často blíž k následujícímu znaku, ke kterému nepatří, a někdy s ním dokonce splývá. Háček je oproti apostrofu subtilnější, a proto je při návrhu písma možné podstatně lépe jej polohovat.

Další výhodou spatřuji v bohatosti nabídky písem, ve které lze nalézt pěkná písma knižní, novinová, nadpisová i různá kaligrafická ozdobná písma pro akcidenční sazbu. Mezi 257 řezy je 32 rodin, u kterých jsou dostupné všechny čtyři běžné varianty (písmo normální, polotučné, kurzíva a tučná kurzíva). Mnoho uživatelů, a zejména těch, kteří používají $\text{T}_{\text{E}}\text{X}$ k profesionální typografické práci, může v PLS II získat již docela slušný repertoár písem, umožňující vždy pro daný druh tiskoviny zvolit vhodný font. Uvažme, že v instalacích $\mathcal{C}_{\text{S}}\text{T}_{\text{E}}\text{X}$ u je nabízena pouze statická antikva Computer Modern, esteticky vhodná hlavně pro odborné texty, méně pro beletrii, ještě méně pro noviny a už vůbec pro akcidenční sazbu, doplněná o základní sadu Adobe fontů, počestěných ne zcela dokonalou metodou kompozitních znaků. Při stávající nabídce je tedy výběr vhodného a současně kvalitního národního fontu mnohdy nemožný.

²Tuto metodu používá například známý konvertor **a2ac** [6] Petra Olšáka.

A B C D E F G
H I J K L M N O P Q R
S T U V W X Y Z
À Á Â Ã Ä Å Æ Ç È É
Ê Ë Ì Í Î Ï Ñ Ò Ó Ô Õ
Ö × Ø Ù Ú Û Ü Ý Þ

a b c d e f g h i j k l m n o
p q r s t u v w x y z
ā ä å ç è é ê ë ì í î ï ñ ò ó ô õ ö ø ù
š ť ů ů ů ŷ ž

1 2 3 4 5 6 7 8 9 0

Za kolik?

CD-ROM PLS II je možné u firmy jednotlivě objednat za 750 Kč. Na webovských stránkách [1] je však možné se dočíst, že firma poskytuje výrazné množstevní slevy a například při odběru 50 kusů stojí jedno cédečko už jen 253 Kč, jinými slovy, jeden řez písma pak vyjde na méně než 1 korunu!

Příznám se, že právě tento fakt mě přivedl k úvaze o možném hromadném objednání CD-ROMu pro členy sdružení ζ TUG a případné další zájemce a stal se jedním z hlavních popudů pro sepsání této recenze.

Souhrnné hodnocení a perspektivy

Vzhledem k tomu, že na trhu doposud zcela chyběla písma ve formátu Type 1 v dobře udělané národní úpravě, je třeba projekt PLS hodnotit bezesporu velmi kladně. CD-ROM si díky svému zajímavému obsahu a relativně nízké ceně patrně získá mezi počítačovou veřejností mnoho příznivců.

Mým úkolem však bylo zhodnotit především jeho jednu část z hlediska možného využití v $\text{T}_{\text{E}}\text{X}$ u. Z této pozice musím, hlavně kvůli absenci kerningových párů, dílo hodnotit jako nedodělané, a tudíž ještě nezralé pro doporučení k nákupu.

Dobrou nadějí je však příslib autora, že kritizované nedostatky budou odstraněny v příští edici CD-ROMu PLS III. Doporučuji tedy zatím s nákupem počkat a navrhuji výboru ζ TUGu, aby případnou objednávkovou akci zorganizoval až na následující verzi tohoto CD-ROMu.

Ze stejných důvodů nehodlám zveřejňovat TFM metriky, které jsem vyrobil k vyzkoušení sazby v $\text{T}_{\text{E}}\text{X}$ u. Počítám ale s tím, že bych stejným způsobem otestoval příští verzi PLS III (pokud bude dodělán kerning) a teprve tyto soubory bych potom zveřejnil.

Odkazy

- [1] *Písma, loga, symboly II*. WWW stránka firmy DTP Studio, s.r.o. <http://www.dtpstudio.cz>.
- [2] *Adobe Font Metrics File Format Specification*. (PDF soubor), Adobe Technical Support, 1995, <http://www.adobe.com>.
- [3] *PostScript Language Reference Manual*. Adobe Systems Inc., 2nd edition, Addison-Wesley, 1992.
- [4] Tomas Rokicki: *DVIPS: A $\text{T}_{\text{E}}\text{X}$ Driver*. Manuál dodávaný v elektronické formě s balíkem `dvips`.
- [5] Karl Berry: *Font naming conventions*. In: [4].

A B C D E F G
H I J K L M N O P Q R
S T U V W X Y Z
Á Ä Ć Ď Ě Ě Ĵ Ľ Ľ Ń Ó Ó Ö
Ř Ř Š Ť Ú ù ü ý Ž

a b c d e f g h i j k l m n o
p q r s t u v w x y z
á ä ě ě í ĺ ĺ ň ó ó ö í ř
š ť ú ù ü ý ž

1 2 3 4 5 6 7 8 9 0

BALLANTINES CE BOLD ITALIC

- [6] Petr Olšák: *Program a2ac – manipulace s fontem na úrovni PostScriptu*. Zpravodaj Československého sdružení uživatelů T_EXu, **6** (1), 23 (1996).
- [7] Petr Olšák: *Typografický systém T_EX. ČSTUG 1995*.
- [8] Donald E. Knuth: *Virtual Fonts: More Fun for Grand Wizards*. TUGboat, Vol. 11 (1990), No. 1.
- [9] Petr Sojka: *Virtuální fonty, accents a přátelé*. Zpravodaj Československého sdružení uživatelů T_EXu, **4** (2), 56 (1994).
- [10] Pavel Herout: *PostScriptové fonty pro ty, co o nich moc nevědí*. Zpravodaj Československého sdružení uživatelů T_EXu, **6** (1), 43 (1996).
- [11] Oldřich Hlavsa: *A Book of Type and Design*. SNTL Praha & Tudor Publishing Co. New York 1960.
- [12] Bohuslav Blažej: *Grafická úprava tiskovin*. SPN Praha 1990.

Fontanasia

MICHAL KVASNIČKA

Rád bych vás informoval, že jsem vytvořil WWW stránku Fontanasia. Na tuto stránku bych chtěl umístit všechna volně šiřitelná počestěná písma, která lze využít pro sazbu T_EXem. „Počestěnými písmy“ v tomto kontextu myslím písma v CS-kódování.

V současné době je stránka rozdělená do pěti částí:

1. postscriptová písma počestěná přímo v .pfa nebo .pfb souboru
2. postscriptová písma počestěná na úrovni .afm metriky
3. postscriptová písma počestěná pomocí virtuálních fontů (virtuální fonty jsou použity všude – rozdělení znamená, jaké informace jsou k fontu k dispozici)
4. Metafontová písma počestěná přímo ve zdrojovém kódu
5. Metafontová písma počestěná pomocí virtuálních fontů

V současné době stránka obsahuje asi 15 rodin písem. V kategorii (1) zatím nejsou žádná písma, v kategorii (2) URW Nimbus Roman (tuší někdo, jak se liší od Times Romanu?) a Adobe Utopia. V kategorii (3) jsou zatím zahrnuta ta postscriptová písma, která jsou integrována do postscriptových tiskáren. V kategorii (4) je Malvern a páně Olšákův fascinující Slabikář. V kategorii (5) pak CM Pica (Memphis), Fraktura a Švabach, kompletní Pandora a Punk. Ukázky písem nepřikládám, protože jsou vystaveny na Internetu.

Na závěr několik proseb: Pokud jste počestili nějaké pěkné písmo nebo ho máte stažené ze sítě, prosím, napište mi, pokusím se ho přidat do databáze. (Po-

kud máte pěkně počestěné písmo pouze v Corku, nevadí, mám awk skript, který převede virtuální fonty do CS-kódování.) Pokud najdete na stránkách nějaké chyby, dejte mi, prosím, vědět. Pokud dokážete zveřejněná písma klasifikovat podle ON 88 1101, tj. podle třídění Jana Solpery, prosím, pošlete mi tyto klasifikace.

Adresa Fontanasie je <http://www.econ.muni.cz/~qasar/fontanasia/>.

Přeji příjemné T_EXování.

Michal Kvasnička
qasar@econ.muni.cz

Konverze dokumentů z WordPerfectu do L^AT_EXu – WP₂L^AT_EX

JAROSLAV FOJTÍK

Tento článek se zabývá konverzí textových dokumentů z textového procesoru Word Perfect v 5.x do typografického systému L^AT_EX. Autor v něm popisuje vlastní vylepšenou verzi původního programu, která umí konvertovat akcentované znaky (tedy upravená pro češtinu) a navíc přidává konverzi dalších speciálních rysů, např. tabulek a matematických formulí.

1. Úvod

WP51 byl před několika lety nejúspěšnější textový editor. Byl znám hlavně v USA, ale i v Čechách byl hojně používán. Například všechny knížky vydavatelství GRADA v něm byly vytvářeny. Takto jsem se s ním seznámil. Poté jsem byl donucen začít psát v L^AT_EXu. Proto jsem přemýšlel, jestli by bylo možno původní texty ve WP nějakým způsobem převést do L^AT_EXu. Nalezl jsem konvertor na internetu a pokusil se provést převod textu. Protože výsledky konverze nebyly pro mne uspokojivé, tak jsem se jej rozhodl upravit. Navíc jsem si přál poznat vnitřní strukturu souborů WP.

2. Historie programu

Znalci archivu CTAN mohou být překvapeni. Na co další nový konvertor, když už jsou v archivu dva dostupné? Proč nevyužít rovnou některého z nich? Pokusím se tedy odpovědět.

První konverzní program, jehož autorem je R. C. Houtepen, je poměrně starý a umí konvertovat velmi málo rysů WP. Hlavním a největším problémem původního konverzního programu při konverzi dokumentů v češtině/slovenštině (a jiných jazycích používajících písmena s akcenty) je, že tyto znaky prostě ignoruje. Autora nelze kontaktovat, a proto od tohoto programu bylo odvozeno několik dalších mutací. Za základ popisovaného programu byl vzat právě tento kód. Avšak byl téměř celý přepsán.

Druhý dostupný program vznikl převedením původního programu WP2LATEX z PASCALU do jazyka C++ a přidáním možnosti konvertovat rovnice. Na úpravách se podíleli autoři Glenn Geers, Michael Covington, Claudio Porfiri a Dirk Lellinger. Ani tento program však neřeší problém češtiny korektně. Navíc má jednu odpornou vlastnost, že při konverzi složitějších rovnic padá. Autoři to nazývají Force Lazy Bracketing (Vnucování Líného Závorkování). Po komunikaci s jedním¹ z autorů jsem se dozvěděl, že s tímto produktem již nechtějí mít nic společného.

Samozřejmě, že mohou existovat konverzní programy z WP do L^AT_EXu od jiných autorů. Za zmínku stojí program `wp2x`, který je velmi obecný a umožňuje konverzi z WP do HTML, L^AT_EX a jiných systémů. Je sice velmi obecný, ale jeho konverzní možnosti se omezují jen na nejjednodušší příkazy pro formátování dokumentů. Je volně k dispozici na `sunsite.unc.edu` včetně zdrojového kódu v jazyce C.

Dalším programem podobného zaměření je `texperfect` of Johna Forkoshe. Vzhledem k dosažené kvalitě konverze se k němu raději vyjadřovat nebudu.

Když jsem se dozvěděl o vzniku tolika derivátů původního programu, rozhodl jsem se vytáhnout ze šuplíku svoji upravenou verzi, dodělat do ní všechny nové rysy, které jsem v různých mutacích našel, poslat ji do archívu CTAN a nabídnout ji v tomto časopise potenciálním zájemcům.

Ať se na mne autoři ostatních konvertorů WP nezlobí, ale zde popisovaný program je s nimi nesrovnatelný. Pokouší se o řádově vyšší způsob konverze se kterým jsem se zatím nikde nesešel. Navíc jako jediný je schopen (s jistými omezeními) zpracovat dokumenty Word Perfectu verze 4.x a verze 6.x (Tato vlastnost je dostupná od verze WP₂L^AT_EX 2.43). Domnívám se, že by bylo lepší spojit úsilí a dodělat zbylé zatím nekonvertované vlastnosti WP (např. čínské znaky, speciální symboly, čtverečky) než vyvíjet další vlastní jednoduchý konvertor.

¹Nepsaným konsensem je, že se o program vždy stará ten, kdo provedl modifikaci jako poslední.

3. WP₂L^AT_EX v akci

Popisovaný program je napsán v jazyce Pascal a je nabízen i se zdrojovými texty. Dále již je plně funkční verze v C++. Z důvodu jistých omezení Pascalu již tato verze nebude dále rozvíjena. V balíku `wp2latex.zip` je konvertor přeložen pro použití v operačních systémech MSDOS, Linux, OS/2 Warp 3.0 a Windows 3.11.

Syntaxe programu:

```
WP2LATEX [WP-filename.WP [TeX-filename.TEX]] [/cp895 | /cp852]
  [/texchars] [/charset1 | /charsetCZ] [/optimizesection]
  [/?] [/silent]}
```

Základní spuštění programu pro první experimenty je velmi jednoduché. Stačí pouze odeslat z příkazové řádky jméno programu bez argumentů. Ten si sám požádá o zadání vstupního souboru a následně nabídne soubor výstupní. Stačí odklepnout klávesu ENTER a po krátké době je již vytvořen výstupní soubor pro L^AT_EX. Pro první pokus můžete použít například soubor `test.wp`. Výstupní soubor konvertoru již může být rovnou překládán L^AT_EXem. Pro překlad bude vyžadován ještě soubor `wp2latex.sty`, který je součástí základního balíku.

Neočekávejte lepší vzhled dokumentu vzniklého pouze konverzí z WP do L^AT_EXu! Nejspíše budete muset provést ruční editaci dokumentu. Důvodem této skutečnosti je, že L^AT_EX obsahuje hlavně příkazy pro nastavování struktury dokumentu a slovní procesor WP obsahuje spíše příkazy pro změnu vzhledu textu. Někdy bude též vhodné editovat původní WP dokument a odstranit z něj zbytečné řídicí znaky. To jsou ty znaky, které se v tištěné podobě dokumentu neobjeví, ale jsou stále v souboru (např [tab] na konci řádky). Nebudou-li výsledky konverze uspokojivé na první pokus, je možné zkusit spustit konverzní program s jinými parametry.

Za zmínku stojí, že vlastní konverze je prováděna ve dvou průchodech. V prvním průchodu je prováděna detekce prostředí a je konvertována většina rysů WP. Ve druhém průchodu je prováděna optimalizace textu, doplňován úvod a zakončení sekcí a dokumentu.

Pro vaši představu nyní uvedu seznam rysů, které jsou programem WP₂L^AT_EX akceptovány. Podotýkám, že ne všechny vlastnosti vyjmenovaných řídicích kódů jsou konvertovány na 100 %.

- Posunutí (doleva, doprava, nahoru, dolů) [`AdvDn`] [`AdvUp`] [`AdvLft`] [`AdvRtg`]
- Křížové odkazy
- Rovnice
- Rozšířené znaky (cizí jazyky/akcenty, matematické symboly, řecká abeceda - ne čárové kresby)
- Zarovnání textu napravo [`Fls Rgt`]

- Poznámky pod čarou [Footnote]
- Poznámky na konci textu [Endnote]
- Umístění textu :
 - vlevo [Just Left]
 - napravo [Just Right]
 - centrované [Just Center]
 - plné [Just Full]
- Vynucený konec stránky (Hard Page) [HPg]
- Vynucený konec řádky (Hard Return) [HRt]
- Záhlaví a Zápatis [Header] [Footer]
- Odsazení :
 - vlevo [->Indent]
 - vlevo a vpravo [->Indent<-]
- Obrysové písmo (Outlining)
- Přepis (Overstrike) [Ovrstk]
- Umístění čísel stránek
- Tabelátory : [Tab]
 - nastavení
 - levé tabelátory
 - centrované tabelátory
 - pravé tabelátory
- Tabulky
- Následující tvary a velikosti písem :
 - Zvláště velké (Extra large) [Ext Large]
 - Velmi velké (Very large) [Vry Large]
 - Velké [Large]
 - Malé [Small]
 - Drobné písmo [Fine]
 - Horní index [Supscript]
 - Dolní index [Subscript]
 - Kurzíva [Italic]
 - Kapitálky (Small caps) [Sm Cap]
 - Tučné [Bold]
 - Podtržené (Underlined) [Und]
 - Dvojitě podtržené (Double-underlined) [Db1 Und]

4. Rozbor dílčích aspektů konverze

4.1. Konverze češtiny

Tomuto tématu se budu nyní věnovat o něco podrobněji. Problémy, které zde bylo potřeba řešit jsou celkem dva. Je používáno u nás velké množství způsobů

kódování češtiny. Tento problém vlastně $\text{T}_{\text{E}}\text{X}$ již řeší sám o sobě pomocí svého způsobu zápisu akcentovaných znaků. Proto program $\text{WP}_2\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ provádí implicitně konverzi právě do sedmibitového $\text{T}_{\text{E}}\text{X}$ u. Na požádání umí navíc výstupní kód vytvořit v osmibitovém kódování. Znaký s akcenty, které se v tomto kódování nevyskytují, jsou samozřejmě ponechány v notaci $\text{T}_{\text{E}}\text{X}$ u. Pro přepnutí do tohoto kódování lze použít následujících přepínačů.

`/cp895` Zvolí výstupní kódování Kamenických.

`/cp852` Zvolí výstupní kódování ISO8 (Latin 2).

Dalším problémem, který vzniká při převodu akcentovaných znaků, je několik interních reprezentací češtiny. Pro řešení tohoto problému jsem navrhl přepínače `/charset1` a `/charsetCZ`. Předpokládám, že nejčastější je znaková sada `charset 1`. Pokud budete mít jinou znakovou sadu², zkuste mi poslat soubor `charactr.wp`.

Pro přehled o konverzních schopnostech programu doporučuji překlad souboru `charactr.wp`, který je dodán spolu s programem `wp2latex.exe`. Pak můžete rovnou na obrazovce vidět, které znaky jsou překládány, a z přeloženého souboru `charactr.tex` je možno vypátrat jak.

4.2. Matematické formule

Zápis matematických formulí ve WP je velmi podobný zápisu formulí v $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u. Proto, jak jsem se již zmínil, předchozí verze umožňovala provádět konverzi formulí. Použitá metoda se snažila detekovat celý výraz a podle toho vytvořit zápis v $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u. Podle mě byla použitá technika příliš složitá. Dokonce tak složitá, že ji ani sám autor nedokázal dovést do bezchybného stavu.

Proto ve své verzi konvertoru používám jinou (vlastní) metodu založenou na kontextové gramatice. Zjednodušeně by se dalo říci, že všechny symboly Word Perfectu představují neterminální symboly. Příkazy $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u jsou považovány za terminální symboly. V programu je definováno několik desítek přepisovacích pravidel, které jsou aplikovány na množinu symbolů do té doby, dokud se v množině vyskytují neterminální symboly.

Domnívám se, že není potřeba zatěžovat čtenáře dalšími informacemi tohoto rázu. Poznávám, že se mi zatím nepodařilo ve WP napsat takový výraz³, který by byl překonvertován chybně. To však neznamená, že se takový výraz nepodaří najít čtenáři tohoto článku.

4.3. Tabulky

Podobně, jako $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ umí i Word Perfect vytvářet tabulky. Tabulka ve WP vypadá asi následujícím způsobem:

²Lokalizátoři a tvůrci Word Perfectu, styďte se za takový zmatek!

³Možnosti zápisu formulí ve WP jsou o dost chudší než $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u.

```
[Tbl Def:I;3,5.31cm,5.31cm,5.31cm]
[Row] [Cell] Left Centered [Cell] Center [Cell] Right Centered
[Row] [Cell] 111 [Cell] 222 [Cell] 333 Centered
[Tbl Off]
```

A její ekvivalent po konverzi je:

Left centered	Center	Right centered
111	222	333

Při konverzi tabulek nastává pouze jediný problém. WP má mnohem obecnější možnosti zápisu tabulek. Umožňuje některé čáry vynechávat, měnit pozici textu v jednotlivých buňkách, některé buňky stínovat atd. Všechny tyto rysy zůstávají nekonvertovány.

5. Speciality

`/optimizesection`

Program se snaží odhadovat z příkazů WP příkazy pro formátování textu. Detekce kapitol a oddílů je odvozována ze symbolů typu `[Mark: ToC ToCLevel]`. Pro definici oddílů ve WP je však potřeba udělat mnohem více. Většinou je navíc změněn typ písma, jeho velikost atd. Právě tento 'nepořádek' je po zapnutí přepínače `/optimizesection` vyhadzován.

`/texchars`

Normálně jsou znaky typu `$`, `{` atd. konvertovány na `\$`, `\{`. Zapnutí uvedeného přepínače tuto volbu potlačí. Tím je umožněno používat WP jako textový editor pro \LaTeX .

Editace dokumentů \LaTeX u ve WP, které mohou obsahovat azbuku

Při použití parametru `/texchars` lze navíc přidávat i řídicí příkazy \TeX u a \LaTeX u do zdrojového dokumentu WP a tyto příkazy nebudou při konverzi poškozeny. Tímto by se dalo psát rusky mnohem komfortněji než v T602 a používáním programu AZBTOTEX. Tato možnost vychází ze schopnosti konvertovat azbuku do \LaTeX u. Při vhodném nastavení WP lze zobrazit současně jak azbuku tak latinku na obrazovce. Existuje způsob, jak zobrazit na textové obrazovce současně 512 různých znaků a WP jej podporuje.

Konverze dokumentů z WORDu do \LaTeX u

Byl vyzkoušen způsob konverze dokumentů z WORDu⁴ do \LaTeX u. Ve WORDu je potřeba zapnout výstupní filtr pro WP 5.1 a uložit dokument v tomto formátu.

⁴Tento odstavec jsem do textu zařadil zejména z důvodu, že jsem byl na tuto věc již několikrát tázán.

Další postup při konverzi je již totožný s konverzí dokumentů napsaných ve WP. Tento postup byl skutečně ověřen avšak nečekejte od takto konvertovaných dokumentů zázraky. Při konverzi z WORDu do WP dochází ke ztrátě některých informací, které již nelze automaticky obnovit.

Konvertor musel být speciálně upraven pro tento způsob konverze. Word totiž na svém výstupu vytváří množství formátovacích příkazů (smetí), které nemají v dokumentu co dělat a zhoršují jeho přenositelnost.

Konverze poškozených dokumentů

Další zajímavou vlastností je možnost konvertovat poškozené dokumenty, které již nelze načíst ani do WP. Ten se při jejich načítání většinou zhroutí⁵. Pro konverzi poškozených dokumentů je potřeba zapnout přepínač `-safe-mode`. Zvýšená kontrola nepřipustí poškozené objekty ke konverzi. Zejména od této volby si neslibujte zázraky, ale někdy stojí za pokus zachránit co se dá.

Konverze dokumentů zapsaných ve WP 6 a WP 4

Tato možnost je zatím pouze experimentální. Množství konvertovaných rysů je dosud malé. Pokud je mi známo, ještě se o to nikdo nepokusil (mám na mysli hlavně WP 6).

6. Diskuse o kvalitě konverze

Když jsem se rozhodl upravit konvertor, v manuálu jsem se dočetl podivné věci. Prý je po konverzi potřeba vytvořit několik filtrů pro program SED a navíc několik programů v C. Pak je prý potřeba vytvořit si vlastní styl a již bude konvertovaný dokument vypadat skvěle.

Pokusil jsem se program upravit, aby negeneroval na svém výstupu nekorrektní příkazy \LaTeX u. Pokud je mi známo, žádný další autor konvertoru z WP do \LaTeX u se o tuto věc ani nepokusil. Podařilo se mi provést přes 40 dílčích zlepšení, která se promítají do kvality dokumentu po konverzi. Ale dokonalosti se mi zatím dosáhnout nepodařilo.

Některé důvody jsou:

1. Prováděná konverze je pouze dílčí a některé rysy stále zůstávají ignorovány.
2. V dokumentech Word Perfektu se nacházejí nechtěné zapomenuté objekty (jako např. `[Tab]`, `[Tab Set]`, `[indent]` namísto `[Tab]` atd.). Proto doporučuji používat `[Tab]` co možná nejméně.
3. Ve Word Perfektu je povoleno dělat nečisté věci, které jsou v \LaTeX u zakázány, například `\section{Some sect\tableofcontents ion}`. Jak jsem

⁵Toto jsem si prakticky ověřil po uložení své diplomové práce na vadnou disketu.

se již zmínil, těžiště všech optimalizací leží právě v odstraňování a nápravě těchto zjevů.

7. Implementace programu

Současná verze `wp2latex-2.48` je zapsána v C++. Je přeložitelná překladači Borland C++ 3.11, Borland C++ OS/2, GCC DJGPP, GCC Linux a HP UX (posledním překladačem s určitými problémy).

Celý program podporuje začlenění knihovny `gettext`, která umožňuje překládat všechna výstupní hlášení za chodu programu. V současné době mám k dispozici pouze hlášení v českém jazyce (což je pro čtenáře českého časopisu jistě dostatečné). Způsob jakým knihovna `gettext` překládá za chodu texty nepatří do tohoto článku. V případě zájmu by bylo možno o této problematice napsat jiný článek.

Pro operační systémy typu UNIX je povolen zápis argumentů začínající jak lomítkem, tak i pomlčkou.

Vzhledem k rozšíření Pascalu v jiných OS než je DOS a vzájemné nekompatibilitě překladačů Pascalu jsem od verze v Pascalu upustil. Hlavním důvodem byla nemožnost pracovat efektivně s řetězci delšími než 256 znaků.

8. Závěr

Programem $\text{WP}_2\text{L}^{\text{A}}\text{T}^{\text{E}}\text{X}$ dávám uživatelům nástroj pro konverzi dokumentů. Nabízený program je schopen konvertovat i dokumenty o rozsahu několika megabajtů. Tímto vám dávám možnost převést stávající dokumenty z textového procesoru WP do $\text{L}^{\text{A}}\text{T}^{\text{E}}\text{X}$ u a tím je vymanit z komerčního prostředí plného pěkných obrázků a vzájemných nekompatibilit. Poznamenávám, že velké množství programů má výstupní filtr pro WP5.x, a takto by bylo možno konvertovat některé dokumenty do $\text{L}^{\text{A}}\text{T}^{\text{E}}\text{X}$ u dvoustupňově.

Přeji všem uživatelům $\text{WP}_2\text{L}^{\text{A}}\text{T}^{\text{E}}\text{X}$ u, aby jim program dobře sloužil. Vývoj uvedeného programu nepovažuji za ukončený, a proto vítám jakékoli další připomínky včetně spolupráce na jeho dalším vývoji. V případě nejasností nebo s připomínkami se můžete obrátit na moji Emailovou adresu `fojtik@vision.felk.cvut.cz` nebo `fojtik@cmp.felk.cvut.cz`. Nejnovější verze programu je k dispozici na mé *www* stránce `http://cmp.felk.cvut.cz/~fojtik/`. Popisovaný program je též možno získat z archívu CTAN (CTAN:support/wp2latex).

Jaroslav Fojtík
fojtik@vision.felk.cvut.cz

V tomto článku se popisuje postup, jak modifikovat vzhled názvů kapitol, podkapitol, apod. Důraz je kladen zvláště na pochopení mechanismu, jak LaTeX pracuje s kapitolami. Život je velmi rozmanitý a nelze počítat s tím, že již existuje balík, jenž dokáže zcela vyplnit jakékoliv požadavky nakladatele. Nejsou tedy popisována existující makra, ale je zde poskytnut návod, pomocí něhož lze dosáhnout libovolného vzhledu, ať už programováním příslušných maker od začátku nebo modifikací vhodného balíku.

Jak je v tomto seriálu obvyklé, budeme pokračovat v číslování kapitol. Poslední kapitola v předchozí části má číslo 16.

17. Značkování kapitol

Čitelnost dokumentu vyžaduje, aby začátky kapitol byly nějakým způsobem graficky zvýrazněny. Kapitola může začínat na nové stránce nebo je alespoň vertikálně odsazena od předchozího textu. Název je obvykle sázen jinou velikostí písma, může být v kurzívě nebo tučný, dokonce lze použít jiný typ písma. Pod nadpisem také bývá vertikální mezera. To je pouze krátký přehled hlavních rysů.

Je zřejmé, že všechny uvedené požadavky snadno vyplníme vhodnými makry. Pro přechod na novou stránku použijeme `\clearpage` nebo `\cleardoublepage`. Vertikální mezeru udělá `\vspace`, dokonce toto makro vypustí mezeru, pokud by název začínal na nové stránce, aniž bychom to vynucovali. Změna písma také nečiní LaTeXu žádný problém.

Začínající uživatel může dospět k názoru, že vlastně speciální značky pro kapitoly nepotřebuje. Vždyť lze napsat prostě:

```
\vspace{2cm}
{\large\bfseries\makebox[2em][l]{17.}Značkování kapitol}\par
\vspace{5mm}
\noindent ... začátek prvního odstavce kapitoly
```

Takový přístup ale má řadu nepříjemností. Začněme tím, že nebudou fungovat makra pro křížové odkazy `\label` a `\ref`. Ta totiž potřebují informaci, kterou vytvářejí právě speciální značky pro sazbu názvů kapitol¹. Mohli bychom je sice dopsat ručně, ale tím bychom si přidělali dost práce.

¹Podobně jako Donald E. Knuth se autor tohoto článku také dopouští drobných zjednodušujících lží. Celá pravda bude zjevena v jednom z dalších pokračování.

Problematické je také udržení jednotného vzhledu. Zatvrzelí odpůrci strukturního značkování budou trvat na tom, že se nespletou a výše uvedenou sekvencí vždy napíšou stejně. Situace se ale zhorší v případě, kdy nakladatel navržené formátování názvů nepřijme a bude trvat na změně. Nalezení všech názvů a jejich jednotné přepsání není vůbec příjemná práce. Pokud vyhodíme kapitolu nebo naopak dovnitř dokumentu kapitolu přidáme, je nutno vše ručně přechíslovat. Také jsme se ještě nezmínili o tom, že obvykle potřebujeme vytvořit obsah.

Velké nakladatelské domy i někteří jednotlivci si udržují databázi s odkazy do mnoha různých dokumentů. Při překročení určitého rozsahu již nelze takovou databázi udržovat ručně a je nutné, aby dokument prohlédl „indexovací robot“ a příslušné záznamy do databáze automaticky zapsal. Takový robot se ovšem ve špatně strukturovaném textu nevyzná.

Je zřejmé, že používání strukturovaných značek pro názvy kapitol ušetří spoustu práce a nabízí mnoho dalších možností, které běžní \LaTeX isté nevyužívají. Mohou však být nezbytné pro vydavatele, kteří dostávají od autorů zdrojové \LaTeX ové texty.

O výhodách strukturovaného značkování svědčí i praxe z redakce Zpravodaje. Některé příspěvky jsou psány v plain \TeX u. Například Petr Olšák ve svých článcích používá:

```
\def\nadpis #1\par
{... definice vzhledu názvu...}
```

V redakci je doplněn obvyklý obal `\begin{clanek} ... \end{clanek}` a definice nadpisu je nahrazena příkazem:

```
\def\nadpis #1\par
{\section{#1}}
```

Tím lze dosáhnout jednotného vzhledu článku s minimálními zásahy do textu. Pokud má příspěvek libovolné strukturované značkování, pak tedy vůbec nevádí, že není napsán s použitím stylu pro Zpravodaj.

18. \LaTeX a SGML, aneb struktura dokumentu

Tato část bude možná někomu připadat nudná a snad se bude čtenářům i zdát, že sem vůbec nepatří. Bude se totiž poměrně dlouho psát o záležitostech, které se \LaTeX u netýkají. Přesto se nakonec souvislosti objasní, a navíc bude předveden způsob, jak by každý autor měl nad svým dokumentem uvažovat.

Řeč bude o SGML. Tato zkratka znamená Standard Generalized Markup Language. Znalci mi jistě odpustí zjednodušení, jichž se v následujícím textu budu dopouštět. Cílem dalších odstavců je pouze ukázání základních rysů SGML a jejich přirovnání k \LaTeX u. Jemný úvod do SGML, vhodný pro začátečníky, lze najít na <http://sable.ox.ac.uk/ota/teip3sg/>, odkazy na další užitečné zdroje jsou na <http://www.icpf.cas.cz/wagner/links.html>.

SGML je jazyk sloužící k definici struktury dokumentu. Každý typ dokumentu obsahuje určité elementy (název, jména autorů, kapitoly, obrázky, tabulky, atp.) a pomocí SGML určíme, které z těchto elementů jsou povinné a kde se různé elementy smí vyskytovat. S dokumentem pak provádíme v zásadě dvě operace: validaci a transformaci. Při validaci ověřujeme, že dokument je syntakticky správný, tedy že všechny povinné elementy jsou přítomny na svých místech a že předepsané vztahy mezi jednotlivými elementy jsou dodrženy. Při transformaci převádíme dokument na jinou formu.

Předvedeme si to zjednodušeně na příkladu tohoto Zpravodaje. Budeme se na něj dívat jako na knihu, přičemž jednotlivé články pro nás budou představovat kapitoly. Číslo, které právě čtete, by v SGML mohlo vypadat například takto:

```
<zpravodaj>
  <chapter title="Písmo s variantní šířkou:
    nová naděje pro naše úzké sloupce?"
    author="Mirka Misáková">
    Text článku...
  </chapter>
  <chapter title="Fonty ve formátu TrueType v pdfTeXu"
    author="Pavel Janík ml.">
    Text článku...
  </chapter>
  <chapter title="Recenze: CE Písma, loga, symboly II"
    author="Antonín Strejc">
    Text článku...
  </chapter>
  <chapter title="Fontanasia" author="Michal Kvasnička">
    Text článku...
  </chapter>
  <chapter title="Konverze dokumentů z WordPerfectu
    do LaTeXu -- WP2LaTeX"
    author="Jaroslav Fojtík">
    Text článku...
  </chapter>
  <chapter title="LaTeXová kuchařka/4" author="Zdeněk Wagner">
    Text článku...
  </chapter>
  <chapter title="Kam za TUGboat-mi?" author="Štefan Porubský">
    Text článku...
  </chapter>
</zpravodaj>
```

Uvedený příklad ukazuje, že názvy jednotlivých elementů se uzavírají do

úhlových závorek < a >. Některé elementy mohou mít různé atributy, např. <chapter> v našem případě má název a jméno autora. Snadno najdeme korespondenci mezi <cosi> ... </cosi> a L^AT_EXovým prostředím \begin{cosi} ... \end{cosi}. Rozdíl jejich pojetí bude předveden později.

Podíváme-li se hlouběji do textu, mohli bychom najít jemnější strukturu. Tento článek by mohl v SGML vypadat např. takto:

```
<chapter title="LaTeXová kuchařka/4" author="Zdeněk Wagner">
  V tomto článku se popisuje postup, jak modifikovat...
  <section title="Značkování kapitol">
    Čitelnost dokumentu vyžaduje, aby...
  </section>
  <section title="LaTeX a SGML, aneb struktura dokumentu">
    Tato část bude možná...
  </section>
  ...
</chapter>
```

Při validaci nás bude především zajímat, zda má každá kapitola název. Také jsme viděli, že Zpravodaj smí obsahovat pouze kapitoly (články). Zkontrolujeme tedy, zda se nevyskytuje text mimo kapitolu. Můžeme ale k definici dokumentu přistupovat i jiným způsobem. Protože se nic nesmí vyskytovat mimo kapitoly, lze v definici oznámit, že koncovou značku </chapter> lze vynechat. Programy, které zpracovávají SGML dokumenty, pak před <chapter> chybějící ukončující značku doplní. To je významný rozdíl od L^AT_EXu; zde \end{prostředí} nikdy vynechat nesmíme.

Dosud jsme nemluvili o tom, jak je určen vzhled dokumentu. SGML totiž tuto záležitost neřeší². Převod do vizuální formy je pouze jednou z mnoha transformací, kterou lze s dokumentem provést. Zpravodaj musíme převést na podklady pro tisk a do elektronické formy (PDF). K oběma účelům bychom použili L^AT_EX. Kromě toho lze nějakým indexovacím robotem udržovat databázi, která by umožňovala vyhledávání článků, autorů, případně i klíčových slov ve všech číslech Zpravodaje současně.

Uvedený případ Zpravodaje je velmi jednoduchý a použití SGML se zde zřejmě nevyplatí. Představte si ovšem, že chcete dokument pěkně vytisknout, převést jej do PDF, do Online Book ve formátu pro OS/2, Man Page pro UNIX a prezentovat jej na WWW³. Právě k takovým účelům lze využít SGML. Před transformací si navíc ověříme, zda je náš dokument napsán syntakticky správně.

Existuje řada nástrojů pro převod SGML do L^AT_EXu i opačně, ale o těch zde mluvit nebudeme. Jejich popis by totiž vystačil na několik samostatných

²Úmyslně zde zamčlíme DSSSL (Document Style Semantics and Specification Language), neboť to překračuje rámec tohoto článku.

³Jazyk HTML je aplikací SGML.

článků. Důkazem tohoto tvrzení jsou příspěvky [1–4] přednesené letos v březnu na $\text{T}_{\text{E}}\text{X}$ NorthEast Conference a publikované v TUGboatu.

Velmi dlouho jsme mluvili obecně, zdánlivě odtažitě, takže je už nejvyšší čas, abychom se dostali k jádru problému. Ukázali jsme si, že SGML definuje strukturu dokumentu a umožňuje jeho validaci. Autor tedy dostane od programu vynadáno, když zapomene název kapitoly a jiné podstatné elementy. $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ takové schopnosti nemá. Vynadá nám, když místo `\chapter` napíšeme `\capter` nebo když vynecháme `\end{itemize}`. Nebude se ale bouřit, když mu předložíme knihu bez kapitol. Zděsíme se až v případě, kdy napíšeme značku `\section`. První sekce v knize bez kapitol totiž bude mít číslo 0.1!

Vévodkyně z Alenčiných dobrodružství v říši divů [5] řekla: „... and the moral of that is – ‘Be what you would seem to be’ – or, if you’d like it put more simply – ‘Never imagine yourself not to be otherwise than what it might appear to others that what you were or might have been was not otherwise than what you had been would have appeared to them to be otherwise.’“

A jaké poučení z toho plyne pro nás? Každý autor by si měl předem promyslet strukturu dokumentu, který hodlá vytvářet, aby jeho výtvar nedopadl jako zmíněná poznámka Vévodkyně.

V dalším textu se už budeme věnovat výhradně $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u, i když se k SGML budeme občas vracet.

19. Typy $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ových dokumentů

Než začneme s výkladem, připomeneme, že v $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ové kuchařce/3 [7] byla uveřejněna Tabulka 1 na straně 147. V textu je zmínka, že se k ní ještě budeme vracet. Právě zde ji budeme potřebovat. V době, kdy je tento článek psán, je již zmíněné číslo dostupné v elektronické podobě na WWW serveru sdružení <http://www.cstug.cz/>. Vezměte si ji k ruce, abyste se v následujících odstavcích snáze orientovali.

Setkal jsem se s názorem, že třída `REPORT.CLS` je šablonou pro všechny zprávy. To ovšem není pravda. Autor tohoto výroku zjevně neprovedl řádně analýzu svého dokumentu a nezamyslel se nad schopnostmi $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u.

Tvoříte-li dokument v SGML, musíte definovat typ dokumentu. Někdy lze vhodný DTD⁴ najít, jindy musíme DTD upravit k obrazu svému, a pokud máme smůlu, nenajdeme nikde nic použitelného a vše si musíme udělat sami.

Tvoříme-li dokument v $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u, potřebujeme „document class“ a případně „package“ (dříve se jim říkalo „style file“). Často si vystačíme se soubory, které najdeme na CTAN. Někdy je musíme modifikovat nebo si přidáváme vlastní makra a jen zřídka píšeme nové třídy z ničeho.

⁴DTD = Document Type Definition.

Standardní L^AT_EX obsahuje třídy pro nejběžnější typy dokumentů. Leslie Lamport analyzoval obvyklé požadavky na knihy, zprávy, články a dopisy. Podle nich vytvořil třídy BOOK, REPORT, ARTICLE a LETTER. Třída LETTER nemá žádné značky pro kapitoly, neboť v dopisech se takové elementy nevyskytují. Jak zjistíme z Tabulky 1 z minulé části, třídy BOOK a REPORT mají na rozdíl od ARTICLE značku `\chapter`. Kniha se od zprávy liší tím, že se tiskne oboustranně a má jiná živá záhlaví. Řadu rysů lze ovšem změnit. I článek lze tisknout oboustranně, což je například vidět na tomto Zpravodaji. Hlavní soubor totiž začíná

```
\documentclass[twoside]{article}
```

Píšeme-li výzkumnou zprávu, musíme se zamyslet, jaké strukturní elementy budeme využívat. Jestliže v textu budeme mít `\section` a `\subsection`, pak musíme použít třídu ARTICLE. Pokud ale z nějakého důvodu trváme na použití třídy REPORT, musíme kapitoly a podkapitoly značkovat pomocí `\chapter` a `\section`.

Co má ale chudák autor udělat, když napíše celou zprávu, v níž použije `\section` a `\subsection`, a pak je mu vnuceno použití třídy REPORT? Přepisování všech názvů je pracné, ale není třeba zoufat. Stačí následující definice:

```
\let\subsubsection=\subsection
\let\subsection=\section
\let\section=\chapter
```

Příkazy lze rozšířit i na další značky jako `\paragraph`, ale pozor: pořadí příkazů je důležité. Je možný též opačný postup, když v dokumentu máme `\chapter` a jsme nuceni použít ARTICLE. Změní se ovšem nejen místa značek v přiřazeních, ale i pořadí příkazů.

20. Číslování kapitol

Když si napíšete příspěvek s použitím stylu pro Zpravodaj, zjistíte, že `\section` vytvoří nadpis bez čísla. Z běžných příruček víte, že nadpis bez čísla se sází makrem s hvězdičkou, tedy pomocí `\section*`. Zdálo by se tedy, že makro `\section` je zcela přepsáno a očíslování názvů nelze dosáhnout jednoduchou úpravou. Opak je pravdou. Číslování bylo odstraněno jednoduchým L^AT_EXovým trikem.

V minulé části jsme se zabývali tvorbou obsahu. Zmínili jsme se o čítači `tocdepth`, jehož hodnota rozhoduje o tom, které úrovně názvů se v obsahu objeví. Nyní si řekneme, že hodnota čítače `secnumdepth` určuje, které nadpisy budou číslovány. Funguje to zcela analogicky. L^AT_EXová kuchařka vyžaduje číslování až do úrovně `\subsection` včetně. Z Tabulky 1 z minulé části tedy plyne, že musíme použít příkaz:


```
\setcounter{secnumdepth}{2}
```

Chceme-li potlačit číslování kapitol v celém dokumentu, měli bychom používat výhradně čítač `secnumdepth`. Nevýhodou hvězdičkových maker není delší jméno. Kromě toho si můžeme nadefinovat

```
\newcommand\s{\section*}
```

a pak dokonce píšeme mnohem méně znaků. Potíž je však v tom, že hvězdičková makra nezapisují názvy do obsahu a neposkytnou informaci pro živé záhlaví.

20.1. Čítače pro číslování kapitol

Číslo pro každou úroveň kapitoly je udržováno ve stejnojmenném čítači. Lze s ním pracovat pomocí běžných L^AT_EXových příkazů. Můžeme například měnit jejich hodnotu. Tak je to prováděno v tomto seriálu. Na začátku tohoto článku je uveden příkaz:

```
\setcounter{section}{16}
```

Ten zajistí, že první kapitola tohoto dílu bude mít číslo 17. Příkaz `\section` totiž nejprve zvětší hodnotu čítače `section` o jedničku a teprve výslednou hodnotu vytiskne. Přitom též vynuluje čítač `\subsection`. Podíváte-li se do rozvoje maker pro sazbu příslušného nadpisu, zjistíte, že inkrementaci čítače `section` zařídí makro `\refstepcounter`. Nulování `subsection` tam ale nenajdete. To se odehrává jinde. Čítač `subsection` byl totiž deklarován příkazem:

```
\newcounter{subsection}[section]
```

To znamená, že čítač `subsection` bude automaticky vynulován při každé změně hodnoty čítače `section`.

Podobně jsou deklarovány i ostatní čítače. Jedinou výjimkou je čítač `part`, který je nezávislý na ostatních čítačích. Pokud chcete, aby změna čítače `part` způsobila vynulování nejvyššího dalšího čítače, musíte v třídě `ARTICLE` použít povel:

```
\@addtoreset{section}{part}
```

Třídy `BOOK` a `REPORT` vyžadují

```
\@addtoreset{chapter}{part}
```

Nezapomeňte, že uvedený příkaz můžete použít jen v místě, kde má znak `,` kategorie 11 – písmeno (viz např. L^AT_EXová kuchařka/1 [6]).

Někdy se může hodit opačný případ: makro `\@removefromreset`, které by odstranilo jméno čítače ze seznamu těch, které se mají automaticky nulovat. Takové makro vytvořil Johannes Braams [9].

Hodnotu čítače můžete zjistit L^AT_EXovým makrem `\value`. Pokud ale chcete hodnotu získat ve svém složitém makru zejména při změně pořadí expanze pomocí `\expandafter`, je výhodnější použít přímo skutečné jméno čítače. To dostanete předsunutím `\c@` před L^AT_EXové jméno. `\value{section}` tedy poskytne stejný výsledek jako `\c@section`. Trocha opatrnosti je nezbytná u přiřazování. Povel `\setcounter{x}{y}` provede *globální* přiřazení `\global\c@x=y`.

Ke každému čítači existuje též makro pro vytištění jeho hodnoty. Jméno tohoto makra získáte předsazením `\the` před jméno čítače. Nadpis v `\section` se tedy tiskne voláním makra `\thesubsection`.

20.2. Špatný způsob změny formátu číslování

V knize *The L^AT_EX Companion* [8] je uveden příklad, jak špatně a správně modifikovat formát číslování. Zde jej uvedeme jen v mírně změněné podobě.

Zarámování čísla jsme dosáhli definicí:

```
\renewcommand\thesubsection
      {\fbox{\thesection.\arabic{subsection}.}}
```

Všimněte si, že jsme přidali i koncovou tečku, která je v českých dokumentech obvyklá, ale standardní L^AT_EXové třídy ji vynechávají. Vše vypadá dobře až do chvíle, kdy pomocí `\ref` chceme odkázat, že špatný způsob je demonstrován v podkapitole 20.2. Pokud podobný přístup využijete pro kapitoly v třídě BOOK, dočkáte se u čísel rovnic, tabulek a obrázků nemilého překvapení.

20.3 Správný způsob

Podkapitola 20.3 uvádí správný způsob. Je nutno modifikovat jiné makro:

```
\renewcommand\@secntformat[1]{\fbox
      {\csname the#1\endcsname}\enspace}
```

Makru je při volání dodán jeden parametr: jméno příslušného čítače. V našem případě je to `subsection`. Konstrukce s `\csname` (viz [10] nebo [11]) vytvoří při expanzi `\thesubsection`. Makro `\enspace` představuje mezeru o velikosti poloviny čtverčku.

Popsané makro je také správným způsobem, jak přidat za čísla kapitol koncovou tečku. Je to provedeno ve stylu pro Zpravodaj. Příslušná definice zní:

```
\renewcommand\@secntformat[1]{\@nameuse{the#1}.\hspace{.5em}}
```

Místo konstrukce s `\csname` je v ní použito L^AT_EXové makro `\@nameuse`. Koncová tečka je tímto přidána jednotně za názvy všech úrovní.

Podívejte se ještě jednu pozorně na číslo u názvu předcházející podkapitoly. Použili jsme v ní změněnou definici `\thesubsection` a současně platilo výše uvedené makro `\@secntformat`. Proto je tečka i za rámečkem.

21. Změna vzhledu nadpisů

Standardní L^AT_EXové třídy neposkytují příliš mnoho možností a z typografického hlediska nejsou moc dobré. Estetičtějším dojmem působí nadpisy vytvářené třídami z balíku NTGCLASS. Pro jednoduché články a zprávy poskytují tyto třídy postačující řešení, ale pro sazbu krásné literatury jsou zcela nevyhovující.

Knižní tvorba vyžaduje mnohem rozmanitější návrh a nelze počítat s tím, že bychom hotová makra někde našli. Dobrý úvod k problematice změny vzhledu nadpisů najdeme v již zmiňované knize *The L^AT_EX Companion* [8]. Kniha je napsána autory současné implementace L^AT_EXu, takže ji lze považovat za přesný zdroj informací. Bohužel však návod zůstává na povrchu, u vysvětlení základních vnitřních maker L^AT_EXu, ale to v mnoha situacích nestačí. V následujících podkapitolách uvedeme popis základních maker, ale pustíme se hlouběji do nitra L^AT_EXového kernelu, tak jak jsem jej byl kdysi nucen prostudovat. Vysvětlování bude ilustrováno na příkladech skutečných knih.

21.1. Makro `\@startsection`

Toto makro se používá vnitřně pro definice téměř všech příkazů pro názvy kapitol. Výjimkou jsou pouze názvy nejvyšších úrovní, kde je požadován přechod na novou stránku a složitější formátování. Definice makra pro danou úroveň nadpisu pak nemá žádné parametry a při volání `\@startsection` musí dodat šest parametrů. Případnou hvězdičku a nepovinný parametr v hranatých závorkách `\@startsection` ošetří. Syntakticky vypadá volání takto:

```
\@startsection{name}{level}{indent}{beforekip}{afterskip}{styl}
```

Význam jednotlivých parametrů je následující:

name je jméno příkazu, který definujeme, ale bez zpětného lomítka, tedy např. `section`. Z minulého dílu asi tušíte, že jej makro musí znát, aby poslalo správnou informaci pro vytváření obsahu.

level je úroveň nadpisu. Tu musíme zadat nejen kvůli správnému zápisu do obsahu, ale i pro rozhodnutí, zda se má příslušný nadpis číslovat.

indent je odsazení nadpisu od levého okraje. Záporná hodnota způsobí vyčnívání nadpisů vlevo mimo sazební obrazec.

beforekip je vertikální mezera mezi nadpisem a předcházejícím textem. Je to rozměr typu *glue*, takže smí obsahovat `plus` a `minus`. Pokud je zadaná hodnota záporná, bude první odstavec vysazen bez odstavcové zářádky. Skutečná vertikální mezera je vždy určena absolutní hodnotou. Protože před sazbou nadpisu L^AT_EX zahájí odstavec, přidá se k zadané hodnotě ještě `\parskip`.

afterskip je vertikální mezera pod nadpisem. Pokud je ale zadaná hodnota záporná, pak začíná text na stejném řádku a za nadpisem se vynechá místo určené záporně vzatou hodnotou (tedy kladné). Zadává se opět rozměr typu *glue*. Před první odstavcem T_EX přidá `\parskip`. Z toho vyplývá, že při použití makra `\@startsection` nelze pod nadpisem vynechat menší prostor než `\parskip`.

style definuje styl nadpisu. Lze použít libovolné příkazy, které mění styl písma a sazby. Nadpis je sazen uvnitř skupiny, takže zadané příkazy nemohou

ovlivnit vzhled následujícího textu. Můžeme použít například `\bfseries`, `\Large`, `\centering` a mnoho jiných.

Jako jednoduchý příklad uvedeme definice ze stylu pro tento Zpravodaj. Zde používáme jen `\section` a `\subsection`.

```
\renewcommand\section
  {\@startsection{section}{0}{0pt}%
  {-22dd minus -6pt}{11dd minus 3pt}{\raggedright\large\bfseries}}
\renewcommand\subsection
  {\@startsection{subsection}{1}{0pt}%
  {-22dd minus -6pt}{1pt minus 1pt}{\raggedright\bfseries}}
```

Všimněte si, že v rozměrech typu *glue* v uvedených definicích není část plus. \TeX totiž každou roztažitelnost považuje principiálně za nekonečnou. Za jistých okolností se \TeX může rozhodnout, že na stránce není dostatečně vhodné místo pro zlom, oznámí `Underful \vbox` a roztažitelné mezery rozpálí. Tím v sazbě mohou vzniknout obrovské díry. Naproti tomu je stažení vždy konečné. V nejhorším případě pak \TeX nechá větší mezeru dole, což nebývá tak ošklivé, jako nadměrné mezery uprostřed stránky. Také *afterskip* u `\subsection` je deklarován jako `1pt minus 1pt`, což je velmi malá kladná hodnota a na rozhození řádkového rejstříku se téměř neprojeví. Ještě menší kladnou hodnotou je `1sp minus 1sp`, ale vizuálně v tom nebude velký rozdíl.

21.2. Makro `\@afterheading`

Nadpisem kapitoly nesmí končit stránka, ale musí být pod ním nejméně dva řádky prvního odstavce. Pokud by na stránce nebylo dost místa, musíme vysadit nadpis na novou stránku. To vše lze snadno vyřešit vhodnými penaltami. \LaTeX má vše potřebné definováno v makru `\@afterheading`, jež je posledním povelům všech příkazů pro sazbu nadpisů kapitol. Makro ignoruje veškerý vertikální materiál. Před sazbou prvního odstavce zkontroluje logický přepínač `\if@afterindent`. Podle něj se rozhodne, zda má první odstavec mít odstavcovou zarážku. Na konci odstavce obnoví původní hodnoty penalt.

Stejným způsobem funguje vnitřně i `\@startsection`. Podle znaménka hodnoty *beforeskip* se nastaví `\@afterindenttrue` nebo `\afterindentfalse` a zavolá se `\@afterheading`.

21.3. Makra `\secdef` a `\@dblarg`

Nadpisová makra mají tři varianty: základní, s nepovinným parametrem v hranatých závorkách a s hvězdičkou. Makro `\@startsection` tyto případy řeší uvnitř, nemusíme se tudíž o nic starat. Pokud však vytváříme složitější nadpisové makro, musíme si hvězdičku i hranaté závorky ošetřit sami. Autoři \LaTeX u nám pomohli tím, že v kernelu dali k dispozici makro `\secdef`.

Makro `\secdef` má dva parametry. Těmi jsou příkazy, které se mají vykonat podle toho, zda následuje či chybí hvězdička. Například makro `\part` lze definovat způsobem:

```
\renewcommand\part{ ... \secdef \normalpart \starredpart}
\newcommand\normalpart[2][?]{ ... }
\newcommand\starredpart[1]{ ... }
```

Definice makra `\starredpart` odpovídá běžným L^AT_EXovým zvyklostem. Hvězdičku spolklo při své expanzi `\secdef`, takže se o ni nemusíme starat. Složitější situace nastává u `\normalpart`. Všimněte si, že toto makro má nepovinný parametr. Víme, že nepovinný parametr slouží k zadání kratší varianty názvu, která se zapisuje do obsahu a objevuje se v živém záhlaví. Standardní hodnotou tohoto nepovinného parametru je otazník. Přesto můžeme nepovinný parametr vynechat a v obsahu budeme mít správný název. Je to způsobeno tím, že se `\normalpart` zavolá prostřednictvím `\@dblarg`.

Makro `\@dblarg` se používá právě v případech `\macro[arg0]{arg1}`, přičemž standardně má platit `arg0 = arg1`. Pokud za makrem není hranatá závorka, `\@dblarg` vytvoří správný nepovinný parametr. Příkaz

```
\@dblarg{\macro}{arg1}
```

pak bude expandován do tvaru

```
\macro[arg1]{arg1}
```

Je tedy zřejmé, že v případě `\normalpart` se nikdy nemůže otazník do nepovinného parametru dostat bez toho, že bychom jej explicitně uvedli.

21.4. Makro `\refstepcounter`

Často potřebujeme kapitoly číslovat a odvolávat se na ně pomocí makra `\ref`. Již jsme uvedli, že každá úroveň nadpisu má přidělen svůj čítač. Hodnotu tohoto čítače musíme před vytištěním zvýšit o jedničku. Šlo by to provést různými způsoby, ale musíme použít jediný: makro `\refstepcounter`. To kromě zvýšení hodnoty čítače ještě vytvoří další pomocná makra, která budou následně použita makrem `\label`. Pokud bychom použili `\stepcounter` nebo `\addtocounter`, nefungovaly by křížové odkazy.

Za výjimečných okolností můžeme vyžadovat nastavení čítače na hodnotu, která není o jedničku vyšší než předchozí. Pokud mají i v tomto případě fungovat křížové odkazy, musíme příkazem `\setcounter` vložit hodnotu o jedničku nižší a následně použít `\refstepcounter`.

21.5. Makra `\markboth`, `\markright` a přátelé

V odborných knihách máme často živé záhlaví, které obsahuje název aktuální kapitoly a podkapitoly, případně můžeme uvádět název knihy. T_EX pro tyto účely

zavádí primitiv `\mark`. Ten je v \LaTeX u rozšířen tak, aby umožnil vložení dvou značek, levé (`\leftmark`) a pravé (`\rightmark`). Ty reprezentují živá záhlaví na levé a pravé části dvoustrany při oboustranném tisku.

Príslušné značky vytváříme pomocí `\markboth` a `\markright`. První z nich má dva parametry a vytváří obě značky. Druhé makro má jediný parametr. Mění pouze hodnotu pravé značky, levou značku ponechá beze změny.

Nyní se vrátíme na chvílku zpět k definici `@startsection`. Prvním parametrem tohoto makra je jméno příkazu pro nadpis, který vytváříme. Toto jméno se použije též při vytváření příslušné značky. Při expanzi pak `\section` volá `\sectionmark`, `\paragraph` zavolá `\paragraphmark` apod. Všechna tato makra mají jeden parametr a standardně jsou ekvivalentní makru `@gobble`, které tento parametr pouze spolknou. Je tedy ponecháno na uživateli, aby tato makra případně předefinoval s použitím `\markboth` nebo `\markright`.

V jednom z dalších pokračování se zaměříme na živá záhlaví. Tam si pak ukážeme, jak se informace z `\leftmark` a `\rightmark` využijí.

22. Příklady

Již jsme uvedli příklad maker pro nadpisy v tomto Zpravodaji. Nyní si předvedeme několik ukázek z různých knih. Některé z nich byly sázeny ještě ve starém \LaTeX u 2.09, případně byly pro právě přicházející \LaTeX 2_ε upraveny „za pochodu“. Proto jsou některá makra programována jinak, než bylo před chvílí teoreticky vysvětleno. Nebudeme zpětně zasahovat do již odladěných maker, pouze na některých místech upozorníme, jak to jde nyní udělat pohodlněji.

Zde publikované ukázky nejsou přesnou kopií citovaných knih. Byly vysazeny znovu s použitím jiných fontů a do změněného zrcadla, aby se vešly do formátu Zpravodaje.

22.1. Jednoduché nadpisy s živým záhlavím

Začneme s knihou Pataňdžalího jógasútra [12]. Ta má pouze jednu úroveň nadpisů a pro ně bylo zvoleno makro `\chapter`. Hlavní kapitoly jsou číslovány, některé čísla nemají. Pro ty zvolíme variantu s hvězdičkou. Všechny kapitoly se ale musí objevit v obsahu.

Číslovaná kapitola je uvozena slovním označením, např. KAPITOLA PRVNÍ, její název je na dalším řádku. Text je vždy centrováný. Nečíslovaná kapitola má pouze název. Živé záhlaví má na levé straně označení kapitoly (např. KAPITOLA I), vpravo je název. Nečíslované kapitoly mají vlevo název knihy. Podobný případ nastává u poslední stránky kapitoly, pokud se objeví na levé straně. Číslo kapitoly by zde nevypadalo dobře, proto také vysadíme název knihy. Počáteční stránka kapitoly žádné záhlaví nemá.

V makru použijeme nejprve `\secdef`.

```
\def\chapter{\par
  \ifodd\c@page \else
    \markboth{Pataňďžaliho jógasútra}{\rightmark}
  \fi
  \clearpage
  \thispagestyle{empty}\global\@topnum\z@
  \@afterindentfalse
  \secdef\@chapter\@schapter}
```

Všimněte si, jak ošetříme změnu živého záhlaví. Odstavcovou zarážku zrušíme makrem `\@afterindentfalse`. Vnitřní čítač `\@topnum` zakáže plovoucí obrázky a tabulky na horním okraji stránky, aby se nemohly objevit nad nadpisem. V této knize ale žádné plovoucí objekty nejsou, takže příkaz byl vlastně zbytečný.

Hvězdičková verze musí vysadit název kapitoly a zapsat informaci pro obsah. Musí též nastavit značky pro živé záhlaví.

```
\def\@makeschapterhead#1{%
  \vspace*{3ex}%
  {\centering\sc\Large
   #1\par}\nobreak\vskip 4ex minus 7\p@}
```

```
\def\@schapter#1{%
  \edef\scratch{#1}%
  \ifx\scratch\contentsname \else
    \addcontentsline{toc}{schapter}{\uppercase{#1}}\fi
  \chaptermark{#1}\@makeschapterhead{#1}\@afterheading}
```

V minulé části L^AT_EXové kuchařky [7] jsme si řekli, že obsah je obvykle uvozen příkazem `\chapter*{\contentsname}`. Vypadalo by komicky, kdyby obsah začínal odkazem na sebe. Pokud je tedy názvem kapitoly `\contentsname`, žádná informace pro tvorbu obsahu se nezapíše.

Knih začíná obsahem, což je technicky nečíslovaná kapitola, a nečíslovanou předmluvou. Další kapitoly jsou číslovány. Můžeme tudíž `\chaptermark` definovat jednoduchým způsobem.

```
\def\chaptermark#1{\ifnum\c@chapter=\z@ \markright{#1}\else
  \markboth{Kapitola \Roman{chapter}}{#1}\fi}
```

Standardní verze nadpisového makra se liší jen v detailech. Všimněte si, že pro jeho definici je použit způsob z plain T_EXu. Makro `\@chapapp`, které se v L^AT_EXovém kernelu dost používá, expanduje na `\chaptername`, což je v českém

prostředí definováno jako „Kapitola“. Zde jsme jej předefinovali na tentýž text psaný verzálkami.

```
\def\@makechapterhead#1{%
  \vspace*{3ex}%
  {\centering\sc\@chapapp{}} \thechapter\par
  \vskip 1ex
  \Large#1\par}\nobreak\vskip 4ex minus 7\p@}

\def\@chapter[#1]#2{\refstepcounter{chapter}%
  \addcontentsline{toc}{chapter}{\protect
  \numberline{\Roman{chapter}}{\uppercase{#1}}}%
  \chaptermark{#1}\@makechapterhead{#2}\@afterheading}
```

Zbývá už jen definice čítače. Potřebujeme převedení čísla na slovní vyjádření. Použijeme k tomu primitiv `\ifcase`.

```
\def\thechapter{\ifcase\c@chapter \or PRVNÍ\or DRUHÁ\or TŘETÍ\or
  ČTVRTÁ\or PÁTÁ\or ŠESTÁ\or SEDMÁ\or OSMÁ\else
  \@latexerr{Too many chapters}\fi}
```

22.2. Netradiční vzhled kapitol

Nyní se zastavíme u knihy Milaräpa, velký tibetský jógin [13]. Věnovali jsme se jí i v minulé části pojednávající o tvorbě obsahu. Kniha obsahuje tři úrovně nadpisů, `\part`, `\chapter` a `\section`. Právě `\chapter` je pro nás nejzajímavější, další typy nadpisů zde pomineme.

V knize máme dva typy kapitol. Nečíslované kapitoly mají pouze centrovný název, což nečiní žádné problémy. Číslované kapitoly jsou uvozeny slovem Kapitola a římským číslem, pod nímž je uveden název. Dále je kurzívou vysázen stručný obsah a po vynechaném řádku následuje text, přičemž první slovo je vysázeno kapitálkami. Příklad jedné takové kapitoly vidíte na obrázku 1.

Makro pro tisk hodnoty čítače kapitol je konvenční a nevyžaduje podrobnější vysvětlení.

```
\def\thechapter{\Roman{chapter}}
```

Dále si nadefinujeme logický přepínač a token registr. Budeme je brzy potřebovat.

```
\newif\ifnewpage \newpagetrue
\newtoks\chap@toc
```

Makra pro sazbu nadpisů budou mít neobvyklý tvar, proto zde nelze použít `\secdef`. Přítomnost hvězdičky budeme zjišťovat makrem `\@ifstar`. To má dva

KAPITOLA II CHUŤ STAROSTÍ

Smrt a poslední vůle Milaräpova otce; protiprávní přisvojení si majetku strýcem a tetou z otcovy strany; z toho vzniklé starosti, jež musel Milaräpa snášet se svou matkou a sestrou.

OPĚT pozvedl Rächung svůj hlas: „Důstojný učiteli, pověz nám milostivě o svých útrapách a nesnázích po smrti svého otce.“

Obrázek 1: Příklad nadpisu kapitoly z knihy Milaräpa, velký tibetský jógin.

parametry. První se použije v případě, že následujícím znakem je hvězdička, v opačném případě se použije druhý parametr.

```
\def\chapter{\ifhmode \par \fi \@ifstar{\s@chapter}{\n@chapter}}
```

V knize chceme dosáhnout jednotného vzhledu nadpisů. Proto `\s@chapter` ani `\n@chapter` nic netisknou, pouze připraví údaje pro další příkazy.

```
\def\s@chapter#1{%  
  \def\chap@head{#1}  
  \chap@toc{#1}\zw@chapter}
```

```
\def\n@chapter#1{%  
  \refstepcounter{chapter}  
  \def\chap@head{KAPITOLA \thechapter\|#1}  
  \chap@toc{\numberline{\thechapter}#1}\zw@chapter}
```

Veškerou práci provádí dosti složitě makro `\zw@chapter`.

```
\newcommand\zw@chapter[1][\relax]{%  
  \ifnewpage \clearpage \fi \newpagetrue  
  \def\zw@{#1}\def\zw@next{\relax}  
  \ifx\zw@\zw@next  
    \let\zw@\relax  
    \edef\zw@next{\noexpand\addcontentsline{toc}{chapter}%  
      {\the\chap@toc}}  
  \else  
    \edef\zw@next{\noexpand\addcontentsline{toc}{chapter}%
```

```

    {\noexpand\protect\noexpand\cpt@entry{\the\chap@toc}{#1}}
\fi
\zw@next
\addvspace{45pt minus 7pt}
\begin{center}
  \Large\bfseries\chap@head\par
\end{center}
\nobreak\vskip 28dd minus 7pt
\ifx\zw@\relax \else
  \begingroup \interlinepenalty\@M
  \noindent \itshape#1\par
  \nobreak \vskip\baselineskip
  \endgroup
\fi
\@afterindenttrue\@afterheading}

```

Kapitola má začínat na nové stránce s výjimkou první kapitoly každé části. To se testuje logickým přepínačem `\ifnewpage`. Makro `\part` nastaví tento přepínač na `\newpagefalse`, ve všech ostatních případech máme `\newpagetrue`.

Nečíslované kapitoly zapisujeme pomocí `\chapter*{Nadpis}`, formát číslovaných kapitol je složitější. Stručný obsah uvádíme za názvem v hranatých závorkách, tedy `\chapter{Název}[Text.]`. Stručný obsah se tedy dostane do makra `\zw@chapter` jako nepovinný parametr. Hodnota tohoto parametru v případě nečíslované kapitoly bude `\relax`. Podle toho vložíme do `\zw@next` příslušný příkaz pro zápis informací do obsahu. Část jsme již předem schovali v token registru `\chap@toc`. Ten se expanduje právě jednou, proto před `\numberline` nepotřebujeme `\protect`. Některá makra však při definici expandovat nesmíme. Proto před ně vkládáme `\noexpand`.

Do makra `\chap@head` jsme již předem vložili text nadpisu. Nyní jej pouze obložíme formátovacími instrukcemi. Mezera nad nadpisem je zde definována pro případ první kapitoly v každé části. Ve všech ostatních případech vyjde na začátek stránky a \TeX ji pak ignoruje. V číslovaných kapitolách pak vysázíme obsah nepovinného parametru a skončíme magickou sekvencí `\@afterindenttrue \@afterheading`. Přepnutí prvního slova do kapitálek bylo řešeno ručně. Pokud by totiž věta začínala předložkou, měly bychom raději vytisknout kapitálkami dvě slova. To však nelze automaticky zajistit.

Makro `\cpt@entry` bylo popsáno v minulé části [7]. Zde se jím nebudeme zabývat.

22.3. Vložení obrázku

V knize *Châteaux et Légendes, Promenades au Royaume de Bohême et de Moravie* [14] chceme každou kapitolu ozdobit ilustrací. Příklad je na obrázku 2.



VYŠEHRAĐ

LE CHATEAU AU-DESSUS DE LA VLTAVA

Autrefois, dans les temps anciens, sur la Vltava, se tenait un château avec des fortifications et un palais construits tout en bois. Plus tard, il avait reçu le nom de Vyšehrad. Il se dressait sur la rive droite de la rivière, en haut d'un piton rocheux, baigné par les flots agités. Le château était résistant, comme la volonté de ceux qui y gouvernaient.

Obrázek 2: Příklad nadpisu z knihy Châteaux et Légendes, Promenades au Royaume de Bohême et de Moravie.

V tomto případě bylo nutno vyřešit několik problémů. Ve francouzštině se nepíší akcenty nad verzálkami. Protože názvy kapitol budeme sázet verzálkami, ale v obsahu použijeme minusky, musíme místo `\uppercase` použít jiné makro. Mezi francouzskými slovy se ovšem vyskytnou i česká jména. Nakladatel si přál, aby akcenty u českých jmen zůstaly zachovány. Tyto požadavky byly ošetřeny následujícími makry.

```
\def\save@accents{\let\cz@hacek\v \let\cz@acute}'}
```

```

\def\CZ{\protect\CZ@accents}
\def\CZ@accents#1{\begingroup \let\v\cz@hacek
                  \let\'\cz@acute #1\endgroup}
\def\remove@accents{\let\'\empty \let\'\empty
                   \let^\empty \let\v\empty}
\def\protect@accents{%
  \def\'\{\protect\'}%
  \def\`{\protect\`}%
  \def\^{\protect\^}%
  \def\v{\protect\v}}

\save@accents

\def\Uppercase#1{\begingroup \let\zw@protect\protect
  \let\protect\noexpand
  {\protect@accents
  \xdef\zw@temp{#1}}\let\protect\zw@protect
  \expandafter\remove@accents\expandafter\uppercase
  \expandafter{\zw@temp}\endgroup}

```

V názvu kapitoly jsme uvedli `\CZ{Vyšehrad}`, čímž jsme obnovili původní význam maker pro diakritická znaménka. Vše funguje pouze proto, že $\text{T}_{\text{E}}\text{X}$ od nás dostává sedmibitový vstup s řídicími sekvencemi. Ve zmíněné knize toho bylo dosaženo použitím TCP tabulky, která mapuje příslušným způsobem osmibitový vstup. Zpravodaj však z vážných důvodů tuto metodu použít nemůže. Při sazbě ukázky jsme museli česká písmena dočasně udělat aktivními. Použili jsme k tomu neobratně napsané příkazy (ale bylo to rychlejší než psaní maker jako v balíku `INPUTENC`):

```

\def\TiFont#1{\catcode'#1\active \def}

\TiFont áá{\'a}
\TiFont ää{"a}
\TiFont čč{\v c}
\TiFont ďď{\v d}
...

```

Hlavní kapitoly sázíme pomocí makra `\hrad`, které má dva parametry. První z nich je definován jako nepovinný. Obsahuje označení obrázku, který se má vysázet. Protože text byl k dispozici dříve než obrázky (u nich byly pouze známy rozměry), byl zvolen právě tento přístup. Označení obrázků jsme nemuseli hned zadávat a bylo doplněno později.

```

\newcommand\hrad[2][blbost]{\clearpage \ZW@nbreaktrue
\chapterhead{#1}%
{\usefont{T1}{Antique}{m}{n}
\fontsize{17}{25pt}\selectfont
\leftskip \z@ plus 1fil \rightskip\leftskip
\parfillskip\z@ \noindent
\Uppercase{#2}\par}\vskip 6ex plus 2ex minus 2ex
\addcontentsline{toc}{hrad}{#2}
\@afterindentfalse \@afterheading}

```

Všimněte si, že nastavujeme logický přepínač `\ZW@nbreaktrue` podobně jako v minulém příkladě. Testujeme jej pak v nadpisu úrovně `\section`.

```

\def\section#1{\ifhmode \par \fi
\ifZW@nbreak \ZW@nbreakfalse \else \clearpage \fi
{\bfseries\leftskip \z@ plus 1fil \rightskip\leftskip
\parindent\z@ \parfillskip\z@
\Uppercase{#1}\par}\vskip 4ex plus 2ex minus 2ex
\addcontentsline{toc}{section}{#1}
\@afterindentfalse \@afterheading}

```

V obou typech nadpisů nastavujeme přímo rozměry `\leftskip`, `\rightskip`, `\parindent` a `\parfillskip`. Dosáhneme tím vycentrování nadpisů. Zde se nabízí otázka, proč nepoužijeme jednoduše prostředí `center`. Je to dáno tím, že v knize máme změněny některé rozměry, a proto standardní makra občas nefungují.

Usazení obrázků provádí makro `\chapterhead`. Jeho definice je trochu komplikovaná.

```

\newcommand{\chapterhead}[1]{%
\edef\cat@colon{\the\catcode '\:}\catcode '\:=12
\def\get@shift##1,##2,##3:%
\def\@@@lft{##1mm}\def\@@@fn{H:/EPSfiles/Martin/m##3.eps}%
\up@shft34.5mm \advance\up@shft##2mm\relax}
\ifZWdraft
\vbox to 34.5mm{\vss\offinterlineskip\hrule
\@@line{\vrule height 34mm\hss\vrule height 34mm}
\hrule}
\else
\makeatletter
\InputIfFileExists{H:/EPSfiles/Martin/m#1eps.tex}{\typeout{***** Cannot find picture: m#1.eps *****}}
\get@shift0,0,0:}

```

```

\makeatother
\ vbox to 34.5mm{\vss
\iftrue
  \moveleft \@@@lft\vbox to \up@shft
                {\noindent\includegraphics{\@@@fn}\vss}%
\fi}
\fi
\catcode ‘\:\cat@colon
\vskip 20.5mm plus 1.8ex minus 1.8ex % only this can have "plus"
}

```

Nejprve si musíme uvědomit, že sázíme francouzský text. V něm jsou některá interpunkční znaménka aktivní. Dvojtečku však potřebujeme v názvech souborů a musí mít kategorii 12. Původní kategorii (víme, že je to 13, ale napíšeme to obecněji) si tedy schováme do makra `\cat@colon`, abychom ji nakonec mohli obnovit.

Obrázky byly skenovány programem, který nedával těsný bounding box. Stejná kniha pak byla sázena v dalších jazykových mutacích. Abychom nedělali stejnou práci dvakrát, byl ke každému obrázku vytvořen jednořádkový soubor s makrem `\get@shift`, z něhož zjistíme korekční posunutí a skutečné jméno souboru s obrázkem. Makro `\get@shift` je definováno uvnitř `\chapterhead`.

Obrázky se sázejí příkazem `\includegraphics`. Pokud v `\documentclass` uvedeme `DRAFT`, pak místo obrázku dostaneme jen obdélník se jménem souboru. Podobná funkce je naprogramována pomocí přepínače `\ifZWdraft`. Jedná se o relikv z pradávých dob. Kód byl totiž opsán ze stylu pro sazbu velmi staré knihy, kdy ještě nebyl $\text{\LaTeX} 2_{\epsilon}$, obrázky se vkládaly makrem `\epsffile` a pracoval jsem v DOSu. Musel jsem tehdy obcházet neexistenci pohodlnějších nástrojů.

23. Křížové odkazy

O křížových odkazech jsme se již zmínili, když jsme si v podkapitole 20.3 ukazovali, jak správně modifikovat formát tisku čítače. Další poznatek jsme uvedli v podkapitole 21.4. Nyní si řekneme trochu více o tom, jak a proč odkazy fungují.

Nejprve bychom se měli zeptat, jak je vůbec možné, že \LaTeX ví, zda se odkazujeme na `\chapter`, `\section` nebo `\paragraph`. Odpověď je velmi jednoduchá. Zařídí to makro `\refstepcounter`. Uloží totiž jméno čítače do proměnné `\@currentlabel`. Makro `\label` podle toho pozná, na který čítač se odkazujeme, a запиše příslušnou informaci do pomocného souboru. Základní \LaTeX ové příručky uvádějí, že nesmíte nadpis kapitoly či podkapitoly vložit do skupiny bez textu. Má to jednoduchou příčinu. Po uzavření skupiny přestane platit na-

stavení proměnné `\@currentlabel` a obnoví se předchozí hodnota. Pokud pak použijete `\label`, budete se odkazovat na předchozí kapitolu.

L^AT_EX umí také vytvořit odkazy na obrázky, tabulky, rovnice i výčty. S plovoucími objekty však má řada uživatelů obtíže. Křížový odkaz jim směřuje na kapitolu, nikoliv na obrázek. Vysvětlení je prosté. Aby `\label` fungoval správně, musíme nejprve naplnit `\@currentlabel`. To provede, jak již víme, makro `\refstepcounter`, které je ale obsaženo v `\caption`. Proto musíme napsat `\label` až za `\caption`. Protože prostředí je vlastně skupina a po jejím uzavření se obnoví stará hodnota `\@currentlabel`, musí být `\label` uvnitř prostředí.

Představme si, že máme čítač pojmenovaný `CNT` a za určitých okolností provedeme `\refstepcounter{CNT}`. Když pak nadefinujeme `\label`, zapíše se do pomocného souboru hodnota získaná vyhodnocením výrazu:

```
\csname p@CNT\endcsname\theCNT
```

O makru `\theCNT` jsme se již zmiňovali. To definuje tiskovou podobu. Makro `\p@CNT` určuje prefix pro křížové odkazy. V makru `\label` se volá pomocí konstrukce `s \csname`. Ono to ani jinak nejde, ale má to navíc tu výhodu, že makro `\p@CNT` nemusíme definovat. Pokud není definováno, expanduje se příslušná konstrukce `s \csname` na `\relax`, což nám žádnou škodu nezpůsobí. Obvykle nejsou makra typu `\p@CNT` definována. Jiná je situace u výčtů (zde se `\refstepcounter` volá při expanzi `\item`). Například v třídě `ARTICLE` se definuje:

```
\renewcommand\p@enumii{\theenumi}  
\renewcommand\p@enumiii{\theenumi(\theenumii)}  
\renewcommand\p@enumiv{\p@enumiii\theenumiii}
```

24. Popisy obrázků a tabulek

Popis obrázků i tabulek vytváříme stejným makrem. L^AT_EX sám pozná, k jakému objektu `\caption` patří. Postarají se o to makra příslušného plovoucího prostředí. Jeho typ se nejprve vloží do `\@capttype`. Makro `\caption` pak provede `\refstepcounter` a zavolá `\@caption` pomocí známého `\@dblarg`. To zapíše informaci do seznamu příslušných plovoucích objektů a vyvolá `\@makecaption` s dvěma parametry: jménem makra pro tisk čísla (např. `\thefigure`) a popisem.

Někdy chceme neplovoucí objekt, ale potřebujeme `\caption`. Lze to jednoduše obejít. Vezměme si příkazy:

```
\makeatletter  
\newenvironment{mytable}{\ifhmode \par \fi  
  \edef\@capttype{table}}{\ifhmode \par \fi}  
\makeatother  
\begin{mytable}  
\caption{Stručný přehled zvířat}
```

```
\begin{center}
\begin{tabular}{|l|l|}\hline
```

```
...
```

Jejich výsledkem bude:

Tabulka 1: Stručný přehled zvířat

Zvíře	Identifikace
Pes	Baskervillský
Kocour	V botách

25. Literatura

Následující přehled obsahuje jak odkazy na zdroje citované v tomto článku a texty vhodné k dalšímu studiu, tak i knihy, z nichž byly převzaty ukázky vzhledu nadpisů kapitol.

1. Robert S. Sutor and Samuel S. Dooley: *TEX and L^AT_EX on the Web via IBM techexplorer*. TUGboat **19**(2), 157–161(1998).
2. Ross Moore: *Presenting mathematics and languages in Web-pages, using L^AT_EX₂HTML*. TUGboat **19**(2), 195–203(1998).
3. Peter Signell: *One-document scientific publishing for print and Web/CD*. TUGboat **19**(2), 208–213(1998).
4. Michael D. Sofka: *TEX to HTML translation via tagged DVI files*. TUGboat **19**(2), 214–222(1998).
5. Lewis Carroll: **The Annotated Alice**. Alice's Adventures in Wonderland and Through the Looking-Glass, strana 122. Penguin Books 1970.
6. Zdeněk Wagner: *L^AT_EXová kuchařka/1*. Zpravodaj Československého sdružení uživatelů T_EXu, **6** (2), 96–108 (1996).
7. Zdeněk Wagner: *L^AT_EXová kuchařka/3*. Zpravodaj Československého sdružení uživatelů T_EXu, **7** (3), 140–167 (1997).
8. M. Goossens, F. Mittelbach, A. Samarin: **The L^AT_EX Companion**. Addison Wesley, Reading 1994. ISBN 0-201-54199-8.
9. Johannes Braams: *To reset or not to reset*. TUGboat **15**(4), 496–497(1994).
10. D. E. Knuth: **The T_EXbook**. Addison Wesley, Reading 1984. ISBN 0-201-13448-9.
11. P. Olšák: **T_EXbook naruby**. Konvoj, Brno 1997. ISBN 80-85615-64-9. Elektronická verze je k dispozici na <http://math.feld.cvut.cz/olsak/tbn/>.
12. Květoslav Minařík: **Pataňdžaliho jógasútra**. Canopus, Praha 1994. ISBN 80-85202-25-5.
13. W. Y. Evans-Wentz, K. Minařík: **Milaräpa, velký tibetský jógin**. Canopus, Praha 1996. ISBN 80-85202-28-X.

14. Eduard Petiška: **Châteaux et Légendes, Promenades au Royaume de Bohême et de Moravie**. Martin, Praha 1996. ISBN 80-85955-02-4.

Zdeněk Wagner
wagner@mbox.cesnet.cz

Kam za TUGboat-mi?

ŠTEFAN PORUBSKÝ

Časopis TUGboat bol založený v roku 1980 združením T_EX Users Group (<http://www.tug.org/>). Zájemcovia o toto periodikum sa môžu prihlásiť do tejto záujmovej organizácie na adrese

TeX Users Group
1466 NW Front Avenue, Suite 3141
Portland, OR 97209, USA

email: office@tug.org
fax: +1-503-223-3960
voice: +1-503-223-9994

a dostanú ho ako členský časopis. Členský príspevok je \$60 ročne.

ČSTUG je kolektívnym členom TUG (\$350 / rok) a z tohto titulu v súčasnosti dostáva 7 exemplárov. Časopisy sú rozmiestnené podľa nasledujúceho rozdeľovníka:

- 1 ks Matematicko-informatické oddelení MFF, Knihovna Václava Hlavatého (Karlín), Sokolovská 83, Praha 8 Karlín, tel. 02-2191 3292. Výpožičná doba knižnice a študovňa je otvorená v čase:

Po	9.00 – 18.00
Ut	9.00 – 18.00
St	9.00 – 18.00
Št	9.00 – 18.00
Pi	9.00 – 15.00

K dispozícii sú tieto ročníky alebo čísla:

- 1 ks knihovna MÚ AV ČR, Žitná 25, Praha 1. Výpožičná doba a študovňa je otvorená v čase:

Po	8.00 – 12.00	13.00 – 15.00
St	8.00 – 12.00	13.00 – 15.00
Pi	8.00 – 13.00	

K dispozícii sú tieto ročníky alebo čísla vol.14/1-4; 15/1-4+1,2; 16/1,3; 17/1-4; 18/1,2+1,2.

• 1ks knihovna FI MU (2. patro v pravém křídle budovy), Botanická 68a, 602 00 Brno, <mailto:knihovna@fi.muni.cz>, <telnet://tinlib@knihovna.muni.cz>, katalog Informatika. Knižnica je otvorená v čase:

Po	8.00 – 17.00
Ut	8.00 – 17.00
St	8.00 – 17.00
Št	8.00 – 17.00
Pi	8.00 – 14.00

K dispozícii sú tieto ročníky alebo čísla: vol.11/1-4; 12/1-4; 13/1-4; 14/1,3-4; 15/1-4; 16/1-4; 17/1-4; 18/1-4, k číslu 2 je CD-ROM TeXlive II, 19/1-2, k číslu 1 je TeXlive III CD-ROM a 4AllTeX 4 dvou CD-ROM, k číslu 2 je CTAN trojCD-ROM.

• 1 ks České Budějovice.

K dispozícii sú tieto ročníky alebo čísla: vol. 14/1,3,4; 15/1,2; 17/1,2,3,4; 18/1.

• 1 ks knižnica Matematického ústavu SAV, Štefánikova 49, SK-814 73 Bratislava. Knižnica je otvorená každý pracovný deň (okrem jedného mesiaca dovolenky v lete) od 8.00 do 14.30.

Po	8.00 – 14.30
Ut	8.00 – 14.30
St	8.00 – 14.30
Št	8.00 – 14.30
Pi	8.00 – 14.30

• 1 ks knižnica Matematicko-fyzikálnej fakulty, Pavilón informatiky, Mlynská dolina, SK-842 15 Bratislava. Jednotlivé čísla sú požičiavané len prezenčne v čase (k dispozícii je kópírka):

Po	8.00 – 18.00
Ut	8.00 – 18.00
St	8.00 – 18.00
Št	8.00 – 18.00
Pi	8.00 – 18.00

K dispozícii sú tieto ročníky alebo čísla: 17/1-3

Štefan Porubský
Ústav matematiky VŠCHT,
Technická 5, 166 28 Praha 6
e-mail: porubsk@vscht.cz

Zpravodaj Československého sdružení uživatelů T_EXu

ISSN 1211-6661

Vydalo: Československé sdružení uživatelů T_EXu
vlastním nákladem jako interní publikaci

Obálka: Bohumil Bednář

Počet výtisků: 710

Uzávěrka: 15. září 1998

Odpovědný redaktor: Zdeněk Wagner

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

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

fax: 05-412 125 68

e-mail: cstug@cstug.cz

Zřízené poštovní aliasy sdružení ČSTUG:

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 ČSFAQ

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