OBSAH

Organizátoři úvodem: Sborník z T _E Xperience 2008!	97
Redakce Zpravodaje: Errata ke Zpravodaji 1–2/2008	98
Program T _E Xperience 2008	102
Jiří Rybička: Typografie a T _E X	104
Jan Přichystal: Inovace a rozšíření systému T _E XonWeb	110
Petr Sojka, Michal Růžička: Publikování z jednoho zdroje v odlišných formátech pro různá výstupní zařízení	116
Petr Olšák: DocBy.T _E X – dokumentování zdrojových textů T _E Xem	130
Tomáš Hála: Značkovací styl pro rychlou sazbu bibliografických citací . .	142
Petra Talandová: Možnosti tabulkové sazby	151
Zdeněk Wagner: L ^A T _E X v sazečské praxi	161
Sbohem T _E Xperience 2008! Buď vítána T _E Xperience 2009!	175

Zpravodaj Československého sdružení uživatelů T_EXu 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, nejlépe jako jeden archivní soubor (.zip, .arj, .tar.gz). Postupujte podle instrukcí, které najdete na stránce <http://bulletin.cstug.cz/>. 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, formáty Macintosh 1.44 MB a EXT2 jsou též přijatelné. Nezapomeňte přiložit všechny soubory, které dokument načítá (s výjimkou standardních součástí L^AT_EXu), zejména v případě, kdy vás nelze kontaktovat e-mailem.

ISSN 1211-6661 (tištěná verze)
ISSN 1213-8185 (online verze)

Vážené T_EXistky a T_EXisté,

v rukou držíte Zpravodaj 3/2008 a zároveň sborník z prvního ročníku mezinárodní konference T_EXperience 2008, kterou pořádali Československé sdružení uživatelů T_EXu a Ústav statistiky a kvantitativních metod Fakulty managementu a ekonomiky Univerzity Tomáše Bati ve Zlíně. Akce se konala ve dnech 3. – 5. října 2008 na turistické chatě Jestřábí na Rusavě. Snahou organizátorů bylo po několika letech navázat na zdařilou sérii konferencí SLT.

V našem zájmu je v této konferenci každý rok pokračovat. Proto ještě dřív, než se zahlubáte do poutavých příspěvků, tak zveme na ročník příští. Pozvánku najdete v závěru sborníku.

Naši příjemnou povinností je poděkovat všem. Organizátorům Zdeňku Wagnerovi, Pavlu Střížovi a Vítu Zýkovi, ČSTUGu, ÚSKM FaME UTB ve Zlíně, kolegům Miroslavě Dolejšové a Martinu Kovářikovi, zvaným i nezvaným přednášejícím, tiskařům, všem skvělým autorům, recenzentům, korektorům Martinu Střížovi a Davidu Cattovi, grafikům Petru Nevřivovi a Igoru Novozámskému, pracovníkům turistické chaty Jestřábí a rodině Střížově, hostům a studentům FaME UTB ve Zlíně, i vám, příchozím na konferenci, a vašim jistě všetečným dotazům a názorům. Neméně děkujeme České statistické společnosti za možnost přetisknout ve sborníku materiály o Rusavě, které byly publikovány ve sborníku Stakanu z roku 2003 plus několik dalších obrázků s T_EXovou tematikou z Informačního Bulletinu 4/2003.

Těm z vás, kteří na konferenci být nemohli, přikládáme i program konference, abyste mohli zaplakat nad tím, o co vše jste přišli. :-) Snad na viděnou příště?! Na webu vyvěšeném nebo v brzkém Zpravodaji lze očekávat DVD z této akce.

Získávání zkušeností a nových poznatků kolem T_EXu a jeho přátel zdar!

Praha, 1. října 2008.

Organizátoři T_EXperience 2008

Jak nás upozornili čtenáři Zpravodaje, bylo by vhodné zdůraznit následující. My tak s radostí konáme.

Redakce Zpravodaje se omlouvá za chybnou informaci v článku Karla Píšky ve Zpravodaji 1–2/2008 Testování LM-fontů s ohledem na čs. sazbu. Místo věty „ ζ TUG dosud neměl k dispozici pokračovatele CM-fontů ve formátech Type 1 a OpenType.“ mělo být uvedeno „ ζ TUG dosud neměl k dispozici pokračovatele CM-fontů ve formátu OpenType.“, neboť Type 1 ζ -varianty z dílny Petra Olšáka jsou již dávno k dispozici a hojně používány zvláště uživateli pdf $\text{\TeX}$ u. Petru Olšákovi za ně patří velké poděkování.

Pravdou též je, že důsledné a až mravenčí testování Karla Píšky objevilo v ζ -fontech mnoho nekonzistencí. Protože oprav v ζ -fontech se zatím žádný dobrovolník neujal, zatímco všechny chyby písem LM byly opraveny i byla zapracována většina českých typografických specifik, a protože LM jsou nadmnožinou ζ , doporučuje ζ TUG k používání s českými i slovenskými texty LM rodinu fontů.

Děkujeme také České statistické společnosti (ČStS) za další upozornění a vylepšení. Tímto děkujeme autorům a redakční radě ČStS za souhlasy k přetištění následujících článků:

- Prezentace v $\text{\TeX}$ u (Luboš Prchal a Pavel Schlesinger),
- Poster v $\text{\TeX}$ u (Luboš Prchal a Pavel Schlesinger),
- Ukázkám posterů (Luboš Prchal, Roman Kasal a kolektiv),
- Velkovýroba tabulek pomocí AWK (Zdeněk Hlávka).

První tři příspěvky byly publikovány v mimořádném českém čísle Informačního Bulletinu ČStS roku 2005. Poster pana Prchala byl k vidění na CompStatu 2004 a Robustu 2006, poster pana Kasala na Robustu 2006. Článek pana Hlávky byl publikován v Informačním Bulletinu ČStS 3/2008.

VÍTĚZNÁ OCHRANO MORAVY

ZŮSTAŇ MATKOU LIDU SVĚMU

VÍTEZNÁ OCHRANO MORAVY ZŮSTAŇ MATKOU LIDU SVÉMU

Po ničivé a svým rozsahem největší válečné smrti, která se během třiceti let několikrát přehnala Evropou (1618–1648), soustředili věřící svoji naději do úcty mariánské. A tak ve 2. polovině 17. století vznikají mariánské poutní místa nebývalou měrou. Procesí putují k Boží rodičce na mnohé posvátné hory a kopce, opředené často zbožnými tradicemi a pověstmi o zázračných událostech a uzdraveních.

K takovým posvátným horám patří i Svatý Hostýn – nejnavštěvovanější mariánské poutní místo na Moravě a po Velehradě snad nej památnější vůbec. Již po tři století je hostýnská hora se svojí mariánskou svatyní cílem tisíců poutníků z blízkého okolí i vzdálenějších krajů, kteří zde nacházejí útočiště ve svých strastech a potřebách.

Hostýnské vrchy, nazývané podle památné hory Hostýna (735 m n. m.) jsou nejzápadnějším výběžkem moravských Karpat. Hostýn tvoří dva vrcholy. Severnější s rozhlednou je nejvyšším místem hory, zatímco druhé temeno (718 m n. m.) u západního okraje je zastavěno poutním chrámem a dalšími objekty. Hora Hostýn s charakteristickou siluetou kostela je již zdaleka viditelná díky rovinaté krajině úrodné Hané, která se západně od Hostýnských vrchů široce rozprostírá kolem řeky Moravy.

Někteří autoři opírajíce se o pověsti a tradici, ale spíše svým zbožným přáním, rádi by vyšperkovali historii Hostýna nejen „zázrakem“ s Tatary ve 13. století, ale nejraději by posunuli počátky křesťanské tradice až do doby velkomoravské k sv. Cyrilu a Metodějovi, ba dokonce až do slovanského dávnověku, kdy zde měla být uctívána modla Hostýna či Radegasta. Hostýn je však poprvé spolehlivě historicky doložen teprve v 16. století. Tím však není řečeno, že nejstarší osídlení tohoto místa nesahá hluboko před jeho psané dějiny. Pravěké osídlení hostýnské výšiny dokládají četné architektonické nálezy. Plastičtější a úplnější obraz pravěkého a časně historického osídlení Hostýna může však v budoucnu podat jen systematický archeologický výzkum.

Mezi stovkami poutních míst v Evropě i jinde ve světě stěží najdeme obraz podobný obrazu hostýnskému – „Matku s blesky“. Panna Maria s Ježíškem, který metá na nepřátele křesťanů blesky. Tento originální výjev není sice původní, vznikl teprve na základě barokní pověsti o Tatarech, natolik je však již srostlý s pojmem Svatého Hostýna, že jedno od druhého nelze již vůbec oddělit. Hostýnskou pověst nemáme doloženu v soudobých pramenech, poprvé se o ní zmiňují až po polovině 17. století jesuitští historiografové Bohuslav Balbín a Jiří Crugerius. Nebudeme proto daleko od pravdy, když vznik pověsti o zázračném zachránění Moravanů proti moři Tatarů položíme až do doby rekatolizace po třicetileté válce.

Za vpádu Mongolů, lidově zvaných Tatařů, na Moravu v roce 1241, hledali lidé pro sebe a svůj nejskromnější majetek útočiště v lesích a na horách. Takovým útočištěm se mohl zřejmě stát i Hostýn, zvláště když zde již existovaly mohutné valy pravěkého osídlení. Traduje se, že Moravané utekli před Tatary na Hostýn a za navršenými záseky statečně odrazili jejich útoky. V horkém létě dolehl však na obránce nedostatek vody. V nejvyšší tísní se prý obrátili s důvěrou k Matce Boží, na jejíž přimluvu vytryskl hned pod temenem hory pramen. Nato Tataři vidouce marnost svého úsilí obrátili se k Olomouci.

Boj křesťanů proti Tatarům v roce 1241 nebyl tak důležitý po stránce historické věrohodnosti, jako spíše symbolem boje křesťanské Evropy proti tureckému nebezpečí, které po pádu Cařihradu roku 1453 až do bitvy u Vídně v roce 1683, smrtelně ohrožovalo evropskou civilizaci. V době, kdy se rodí hostýnská pověst o Tatarech, objevuje se desetitisícová turecká armáda na východní Moravě. V roce 1663 Turci, lidově opět nazývaní Tataři, třikrát vpadli na naše území a napáchali na moravsko-uherském pomezí nedozírné škody, kruté násilí a nepředstavitelné utrpení zejména prostému venkovskému lidu. Někteří badatelé právě v tomto tureckém vpádu viděli skutečný zdroj vzniku hostýnské pověsti.

Neodmyslitelnou součástí každého poutního místa je také pramen zázračné uzdravující vody. Pramen je nejen důležitým symbolem, ale i skutečným občerstvením znavených poutníků a provází ho také uzdravující pověst. Na Hostýně tryská takový pramen ze skály na západním svahu přímo pod chrámem. Podle této tradice vytryskl při tatarském ležení v roce 1241 k záchraně obležených křesťanů a víra v zázračnou moc hostýnské vody má ve zbožném moravském lidu hluboké kořeny podnes.

Ať je náš pohled na Svatý Hostýn jakýkoliv, přesto můžeme přijmout pověst o Tatarech jako symbol hledání nadpřirozené pomoci tam, kde naše prostředky byly již dávno vyčerpány. Vždyť tuto zkušenost jsme si mohli ověřit již tolikrát i v dobách minulých a lidstvo si ji nese i v našem, třetím tisíciletí.

Program T_EXperience 2008

Čtvrtek, 2. října 2008, odpoledne

Po 15:00, příjezdy a registrace čtvrtečních účastníků.

20:00–21:00, večere, 60 minut.

Pátek, 3. října 2008, dopoledne

08:00–09:00, snídaně čtvrtečních příchozích, 60 minut.

Od 09:00 a během dne, registrace ostatních příchozích na konferenci.

Pátek, 3. října 2008, odpoledne

13:00, *začátek konference.*

13:00–14:00, oběd, 60 minut.

14:00–14:10, Orgové (organizátoři), 10 minut, oficiální zahájení konference.

Blok *Obecná typografie a značkování*, předsedající Karel Horák

14:10–14:30, Jiří Rybička, 20 minut, T_EX a typografie.

14:30–15:30, Vít Zýka, 60 minut, Příprava dokumentů pro formátování.

15:30–15:45, přestávka, 15 minut.

Blok *Makrojazyk T_EXu I+II+III*, předsedající Pavel Stríž

15:45–17:15, Petr Olšák, 90 minut. Makrojazyk T_EXu.

17:15–18:15, večere, 60 minut.

18:15–19:45, Petr Olšák, 90 minut. Makrojazyk T_EXu.

19:45–20:00, přestávka, 15 minut.

20:00–21:30, Petr Olšák, 90 minut, Makrojazyk T_EXu.

Společenské akce, uvede Pavel Stríž

Od 22:00, Taneční skupina Kirké, španělské tance;

Z. Wagner, křest knihy pohádek s neformální diskusí;

Taneční skupina Kirké, orientální tance;

Na dobrou noc, předpůlnoční autorské čtení z knihy pohádek

Sobota, 4. října 2008, dopoledne

08:00–09:00, snídaně, 60 minut.

Blok *Výuka I*, předsedající Jiří Rybička

09:00–09:30, Jan Přichystal, 30 min., Inovace a rozšíření systému T_EXonWeb.

09:30–10:00, Michal Růžička, 30 minut, Publikování z jednoho zdroje
v odlišných formátech pro různá výstupní zařízení.

Blok *Fonty*, předsedající Petr Olšák

10:00–10:45, Karel Horák, 45 minut, Typografie a nová písma v mezinárodních projektech.

10:45–11:15, Michal Růžička, 30 min., Nástroje použité v projektu DML-CZ: Tralics, náhrada bitmapových fontů při konverzi PS do PDF.

Blok *ConT_EXt I*, předsedající Petr Olšák

11:15–12:00, Vít Zýka, 45 minut, tutoriál: Logo CON_T_EX_Tem.

Sobota, 4. října 2008, odpoledne

12:00–13:00, oběd, 60 minut.

13:00–18:00, výlet na Svatý Hostýn a přilehlou Rozhlednu (7+7 km).

18:00–19:00, večere, 60 minut.

Blok *Výuka II*, předsedající Petr Sojka

19:00–19:15, student Michal Polášek, 15 minut, Výuka L^AT_EXu paralelně s OpenOffice.org Writerem/Microsoft Wordem či HTML.

Blok *ConT_EXt II*, předsedající Petr Sojka

19:15–20:45, Vít Zýka, 90 minut, tutoriál: Článek CON_T_EX_Tem.

20:45–21:00, přestávka.

Blok *T_EX zkušené*, moderátor Zdeněk Wagner

21:00–22:30, panelová diskuze: kulatý stůl účastníků konference.

Témata: T_EXLive, L^AT_EX_UG a volené téma dle zájmu.

Cestovatelské zkušenosti, uvede Pavel Stříž

22:30–23:45, Milan Štourač, 75 minut, Cestování po Evropě.

Snad také kolování cestopisu vysázeného v L^AT_EXu.

Neděle, 5. října 2008

08:00–09:00, snídaně, 60 minut.

Blok *Praktici*, předsedající Jiří Rybička, po přestávce Karel Horák

09:00–09:45, Petr Olšák, 45 minut, DocBy.T_EX – dokumentování zdrojových textů T_EXem.

09:45–10:00, Tomáš Hála, 15 minut, Značkovací styl pro rychlou sazbu bibliografických citací.

10:00–10:15, přestávka, 15 minut.

10:15–11:00, Petr Sojka, 45 minut, Jak jsem sázel knihu Knuthovu.

11:00–11:45, Zdeněk Wagner, 45 minut, L^AT_EX v sazečské praxi.

11:45–12:00, Good-bye T_EXperience 2008! Welcome T_EXperience 2009!

12:00–13:00, oběd, 60 minut.

Po obědě, plynulé odjezdy účastníků z konference.

/Změny programu vyhrazeny. Budou vyvěšeny na dveřích při vstupu do hlavní budovy./

Počítačová sazba je velmi rozšířenou aplikací běžných uživatelů. Pro zhotovení kvalitního dokumentu je potřebné nejen ovládat příslušný programový produkt, ale především aplikovat typografické zásady. Jak řešit tento problém v T_EXu a jeho nadstavbách? Quo vadis typografie v T_EXu?

1. Úvod

S tvorbou kvalitních dokumentů jsou pevně spjaty pojmy typografie a T_EX. První z nich odpovídá na otázku *proč*, druhý odpovídá na otázku *jak*. Přesto však toto spojení není tak průzračné, abychom mohli pokaždé v praxi říci, že jedno bez druhého není možné. Kvalitní dokument dnes představuje *jazykovou správnost, typografickou správnost a efektivní technickou realizaci v elektronické podobě*. K poznání všech souvislostí vede určitá cesta, kterou pravděpodobně každý tvůrce dokumentů musí projít.

2. Od začátku ...

2.1. Typografické bezvědomí

Zpracovávat dokumenty počítačem bylo možno začít až v okamžiku, kdy se do našich pracovišť rozšířily stroje vybavené elementárními programy. Jednalo se tehdy logicky jen o počítačový strojopis, textové editory WordStar, MikroStar, QEdit, později i Text602. Velkým průlomem bylo vytváření předloh sborníků a skript tiskem na jehličkových tiskárnách. Pro dostatečně kvalitní přípravu takových tiskovin stačily informace získané v kursu psaní strojem.

2.2. První náznaky

Teprve při setkání s programem, v němž se objevují na první pohled záhadné pojmy, jako například kurzíva, sazba do bloku, Bookman apod., se vnímavý uživatel začne zajímat, odkud to pochází a k čemu by to mohlo být dobré.

Z vlastní zkušenosti mohu potvrdit, že když jsem se v roce 1988 poprvé setkal s T_EXem, bylo nutné začít uvažovat trochu jiným způsobem. Možnosti, které se mi tím nabízely, vyžadovaly daleko hlubší znalosti, než bylo v tehdejší době zvykem.

Všeobecné povědomí o typografii podporované počítačem však bylo prakticky nulové. Podařilo se mi zcela náhodou nahlédnout do jedné z mála knih – Computers and Typesetting – a otevřely se mi nové obzory. Velkým posunem byla rozhodně spolupráce na knize Jak publikovat na počítači, kde kapitolu o počítačové typografii napsal Jan Rajlich ml. Setkání s tímto grafikem a typografem pro mě znamenalo nový pohled na celou věc.

2.3. Typografie a L^AT_EX

Jednou z mých nejzajímavějších činností byla příprava knihy L^AT_EX pro začátečníky. V prvním vydání jsem se plně věnoval popisu systému z technického hlediska – řešil jsem tedy jen otázku, „jak“ dokument připravit. Druhé vydání jsem již zcela přepracoval tak, aby výklad akcentoval otázku „proč“ – připojil jsem typografická pravidla, která dávají všem těm technickým zákonitostem pravý smysl.

Samotná znalost typografických pravidel a schopnost orientovat se v nabídkách a službách určitého programu však kupodivu taky nestačí. K těmto dvěma podmínkám je potřebné přidat ještě jednu – schopnost optimálního strukturního značkování a vhodné reprezentace dokumentu v požadovaném formátu. To vede k zobecnění pohledu na dokument, k osamostatnění typografie od technického zpracování a k dalekosáhlým možnostem znovupoužití již zpracovaného a zkorigovaného textu. Dostáváme se tím nad rámec jednoho formátovacího typografického nástroje. Typografie přirozeně nabývá pozice primárního prvku, jehož pravidla zcela rozhodují o dalším zpracování.

3. Co vidíme kolem sebe?

Cesta poznání od typografického bezvědomí k obecným strukturně značkováným dokumentům není jednoduchá. To s sebou automaticky přináší nutnost podpory jednotlivých částí ve výuce na různých stupních škol a v další osvětové činnosti. Stav, který vidíme všude kolem sebe, nás však určitě nenaplnuje pocitem uspokojení. Pokusme se shrnout alespoň vybrané aspekty.

3.1. Typografická pravidla

„Tak nám zabili... pomlčku, řečenou pauzu,“ píše Karel Dostál ve svém svěžím a zajímavém článku z roku 1994. Změnila se nějak situace za uplynulých 14 let?

Obecné povědomí o typografických pravidlech můžeme charakterizovat spíše jako bezvědomí. Pokud jsou někde dodržována základní typografická pravidla, děje se tak často zcela bezděčně, když náhodou použitý program nahradí oboustranně mezerovaný spojovník za pomlčku, vloží automaticky nezlomitelnou mezeru za

neznělé jednoznačkové předložky nebo správně vysadí odstavec za použití předdefinovaných parametrů.

V kontrastu se systematickou výukou ortografie zcela absentuje výuka typografie. Základní zásady patří zcela jistě již na stupeň základních škol, kde se žáci poprvé oficiálně setkávají s programovým vybavením pro zpracování textů. Paradoxní je, že přestože se například učitelé mateřského jazyka rádi ohánějí Pravidly českého pravopisu, vynechávají z nich ve výuce řadu pasáží nutných ke správnému zápisu hladkého textu v jakémkoliv počítačovém prostředí. Všichni určitě víme, že tuto dovednost získanou na základní škole mohou později téměř všichni žáci bohatě využít.

Lze bohužel také konstatovat, že neexistuje relevantní norma, která by byla pro úpravu sázených dokumentů použitelná. Oborová norma ON 88 2503 „Základní pravidla sazby“ zanikla spolu se všemi oborovými normami a nebyla za ni vytvořena žádná náhrada. Zato norma ČSN 01 6910 „Úprava dokumentů připravených textovými editory nebo psaných strojem“ je stále živá, byla poměrně nedávno novelizována a tvoří základ pro strojopisné dokumenty – tím se ovšem stává pro většinu uživatelů podkladem pro sázené dokumenty, pro něž vůbec není určena a v řadě případů poskytuje při této aplikaci zcela nesprávné zásady. Některé příklady dokumentů obsažené v přílohách této normy navíc nešťastně nepoužívají strojopisné písmo, ale proporcionální bezserifové písmo (asi jako demonstraci výstupu z textového editoru), což bezpochyby běžného čtenáře zcela utvrdí ve správnosti aplikace na sázené dokumenty. Samotným příkladům nelze nic vytknout – náhodou (nebo zcela záměrně?) zde není jediný prvek, který by přímo odporoval pravidlům sazby, to ovšem neznamená, že se v praxi lze takovým prvkům úspěšně vyhnout.

Na základě strojopisné normy pak vznikají různá doporučení, či dokonce striktní instrukce pro přípravu dokumentů, například kvalifikačních prací na vysokých školách. V nich se vesele míchají pravidla strojopisná s pravidly sazby, čímž jsou uživatelé přímo nuceni vytvářet nechutné slátaniny typu „písmo Times New Roman, 60 znaků na řádku, řádkování 1,5 a mezera na začátku odstavce 1,25 cm“.

3.2. Systémy WYSIWYG

Většina tvůrců dokumentů silně preferuje grafické rozhraní operačního systému a také veškerého programového vybavení pro zpracování textů. Ať už jsou příčiny tohoto stavu jakékoliv, není důvod se domnívat, že by v dohledné době došlo k nějaké zásadní změně. S uživatelským rozhraním tohoto typu souvisí pojem WYSIWYG – tvůrce má pocit, že na obrazovce vidí výsledný tvar dokumentu. Víme, že obrazovka a tiskárna mají odlišný technický způsob získávání obrazu, nelze tedy tak jednoduše vidět totéž na obou zařízeních. Zaměříme-li se ovšem na nezkušené laické uživatele s tímto reklamním tahem, je komerční úspěch takového programu zaručen.

Grafické rozhraní s sebou přirozeně nese vizuální manipulace systémem „označ a klikni“. Vůbec zde nejsou primárně nabízeny efektivní možnosti strukturního značkování, natož nějaká solidní typografická pravidla. V příručkách a v nápovědách se nanejvýš dozvíme, kterak efektivně označit a kam kliknout.

3.3. Proč ne $\text{\TeX}$?

Z výukových zkušeností vyplývá, že všeobecný trend k povrchnímu vzdělávání uvedenému neefektivnímu a nesprávnému způsobu práce s dokumenty jen nahrává. Například ve specializovaném volitelném předmětu orientovaném na seznámení se systémem $\text{\LaTeX}$ a s některými principy samotného $\text{\TeX}$ u, si každoročně zhruba 120–160 studentů naší fakulty zkusí vytvořit běžné dokumenty včetně matematických výrazů, tabulek a dalších prvků. Přibližně 90 % těchto studentů se však později při tvorbě své závěrečné práce rozhodne použít jiný systém.

Faktorů podílejících se na tomto stavu je bezpochyby více (někteří studenti si mohli předmět zvolit kvůli něčemu jinému než seznámení s $\text{\TeX}$ em), ale všeobecně lze říci, že o genialitu $\text{\TeX}$ u není zájem – její využití je podmíněno příliš velkým intelektuálním vkladem, který nejsou studenti většinou ochotni vynaložit. Žádného začátečníka asi nelze přesvědčit, že například tvorba tabulek a vkládání obrázků v systému $\text{\LaTeX}$ je jednodušší než v programu Microsoft Word. V interaktivním prostředí totiž uživatel snadno může metodou pokus-omyl zkoušet, ve kterém místě a v jaké velikosti bude vkládáný obrázek „vypadat dobře“ a tabulku si „nalinkuje“ podobně jako tužkou na papíře – nemusí vůbec přemýšlet o typografické správnosti a reálné ovladatelnosti získaného výsledku, dokument si ke své plné spokojenosti pěkně „splácá“.

Dalším pádným důvodem pro použití jiných systémů je cílový formát dokumentu – v řadě případů je vyžadováno, aby dokument byl v elektronické podobě předán zadavateli tak, aby jej bylo možno dále upravovat, slučovat s jinými dokumenty do větších celků apod. Je zcela iluzorní se domnívat, že takovým požadovaným výstupním formátem bude $\text{\TeX}$. Jsme svědky masového publikování dokumentů v proprietárních formátech komerčních programů v internetovém prostředí, a to i například ze strany ministerstva školství. Příkladem asi největšího fiaska je příprava skript do matematiky a statistiky v programu Microsoft Word – jak vyžaduje třeba ediční středisko.

Určité rezervy lze vidět i v samotné podpoře běžných typografických prvků v $\text{\TeX}$ u ($\text{\LaTeX}$ u). Opravdu velmi těžko se například vysvětluje začátečníkům, kteří jsou v určitém okamžiku seznámeni se základními pravidly sazby, proč pro nejčastější zúženou mezeru, tedy čtvrtčtverčík, neexistuje předdefinovaný příkaz... Je samozřejmé, že všechno si lze vytvořit, ale ne všechno jde tak snadno jako zmíněná čtvrtinka a (to hlavně) proč to má každý uživatel pořád dokola definovat?

4. Ven z ulity T_EXu

Není asi příliš příjemné konstatovat, že technologie – o které jsme přesvědčeni, že je vynikající, dává nedostižně nejlepší výsledky a je schopna realizovat všechny typografické nuance – je málo rozšířena, je rekvizitou nadšenců, přičemž není pravděpodobné, že by se tento stav radikálně v budoucnosti změnil. Můžeme do toho nějak zasáhnout?

Máme-li prospět systému T_EX a jeho příbuzenstvu, je nezbytné překročit jeho hranice a pokusit se jej svázat s okolním světem. Podívejme se například, jak úspěšným krokem bylo a je přímé generování formátu PDF místo původního DVI. Znamená zcela novou dimenzi a zcela jistě významně posiluje pozici T_EXu.

Nezbytným požadavkem je rovněž schopnost pracovat se všemi fonty instalovanými v systému, zpřístupnit formát Open Type a kódování Unicode. Řešení tohoto aspektu již také existuje.

Za možná kacířskou, ale podle mého názoru stejně významnou, lze považovat myšlenku generování formátu DOC (DOCX). Úkol dříve neřešitelný je dnes ve zcela jiném světle, neboť specifikace formátu DOC je již volně k dispozici. Jak radikálně by se tím zvýšila kompatibilita světa T_EXu a světa komerčních i dalších volně dostupných programů... Je podle mého názoru krátkozraké tvrdit, že to není potřeba. Úsilí vynaložené do této jistě netriviální konverze se však určitě vyplatí.

A také naopak – představme si editační nástroj v podobě uživatelského rozhraní komerčního programu (třeba Microsoft Word), z něhož lze exportovat kvalitní zdrojový text pro některou variantu (nastavbu) systému T_EX. O kolik by se usnadnil život začátečníkům, kteří s počátečním nadšením chtějí tvořit kvalitní dokumenty... Musíme si uvědomit, že příčinou špatného dokumentu není uživatelské rozhraní, ale především rozsah znalostí a schopností uživatele.

Prosazování kvalitní typografie nemůže zůstat otázkou jen jednoho systému (nebo jedné rodiny systémů). Typografická osvěta se musí týkat všech rozšířených programů pro zpracování textů.

5. Závěr

„Chce to jen probudilost a dobrou vůli,“ píše Karel Dostál ve zmíněném článku o prosazování kvalitní typografie. K těmto nezbytným podmínkám lze dnes snad ještě dodat jednu – vůli překročit hranice jedné technologie.

Pokud jsme před čtrnácti lety byli shledáni jako sdružení, „kde se o tomto lepším příště nikoli sní, ale kde se už v tuto chvíli s velkým nasazením a na vysoké profesionální úrovni mění ve skutečnost,“ neměli bychom ani dnes zklamat.

Pomlčka skutečně není starý kráp – kvalitní typografie je součástí celkové kultury.

Summary: T_EX and Typography

Computer typesetting is a very widespread application commonly used with personal computers. It is necessary to handle appropriate programs but it is also very important to apply typographic rules. This paper deals with the question of how to solve this problem in T_EX and its formats? Quo vadis typography in T_EX?

Literatura

1. *ČSN 01 6910 – Úprava písemností zpracovaných textovými editory nebo psaných strojem*. Praha: Český normalizační institut, 2003.
2. DOSTÁL, K. O jedné vraždě a jejím možném odčinění. *Zpravodaj ČSTUG*, č. 4, 1994, s. 163–173. ISSN 1211-6661.
3. RÁBOVÁ, Z. a kol. *Jak publikovat na počítači*. Veletiny: Science, 1996. ISBN 80-901475-7-7.
4. *ON 88 2503 – Základní pravidla sazby*. Praha: Ústav pro normalizaci a měření, 1968.

Príspevek popisuje nové vlastnosti webového rozhraní pro tvorbu dokumentů – systému T_EXonWeb, který vznikl z potřeby jednoduchého a rychlého použití systému T_EX. Jde především o nové uživatelské rozhraní, uživatelské profily a práci s dokumenty. Jsou zde také zmíněny vlastnosti, které teprve budou implementovány a směr, jakým se bude vývoj dále ubírat.

1. Úvod

Vzpomínám si, že když jsem před lety začínal se systémem T_EX, měl jsem velkou radost z prvního dokumentu, který se mi podařilo vysadit. Oproti jiným systémům, se kterými jsem pracoval do té doby, jsem musel pro dosažení výsledku vyvinout nemalé úsilí. Nebylo to způsobeno ani tak složitostí zápisu – byl to v podstatě pouze text „Ahoj, jak se máš“ – jako spíše problémy s instalací a konfigurací. Na počítači jsem měl nainstalovanou Linuxovou distribuci RedHat 6.2 a cesta k získání T_EXu a doprovodných utilit byla poměrně jednoduchá. Vše bylo součástí distribučních balíčků. Jaké bylo však mé překvapení, když jsem zjistil, že po vytvoření jednoduchého dokumentu a jeho přeložení jsem nedostal žádný výsledek. Vysvětlení je jednoduché. V konfiguračním balíčku se v té době vyskytovala implicitní volba výstupu na tiskárnu. Tu jsem bohužel neměl k počítači připojenou, takže jsem ani neměl možnost obdržet výsledek. Dnes by pro mě vyřešení podobného problému nepředstavovalo větší překážku, ale pro začátečníka to jednoduché nebylo. Podobných problémů se vyskytlo více a během let se i průběžně obměňovaly. Novější verze řešily staré potíže, jiné problémy se však zase objevovaly. Například dodnes je v distribuci Fedora balíček **tetex**, který je implicitně nastaven tak, aby překlad **pdf_latexem** vygeneroval výstup do **.dvi**. Čert ví proč.

Při práci se systémem T_EX jsem nezůstal pouze uživatelem. Jako vyučující předmětu Zpracování textů na počítači jsem se snažil přiblížit tento systém i dalším zájemcům. Připravit v učebně pracovní prostředí pro studenty nebyl problém. Horší to ovšem bylo, když si studenti chtěli nainstalovat T_EX doma a vyzkoušet si ho – vytvořit v něm závěrečný úkol nebo napsat diplomovou práci. Často to obnášelo několik kol rozhovorů nad instalačním CD T_EXLive a občas to dopadlo i tak, že student psal závěrečnou práci raději v něčem jiném (v Microsoft Word). To jsem chápal jako své selhání a snažil se tento stav řešit. Inspirován prací ve vývojovém týmu Univerzitního informačního systému na Mendelově

univerzitě, rozhodl jsem se vytvořit jednoduché a snadno dostupné řešení pro sazbu dokumentů v $\text{\TeX}$ u. Výsledek práce – systém $\text{\TeX}$ onWeb, který umožňuje překlad $\text{\TeX}$ ových dokumentů pouze s použitím webového prohlížeče – jsem pak prezentoval na posledním setkání SLT v roce 2004 ve Znojmě (Přichystal – Rybička, 2004).

V současné době je $\text{\TeX}$ onWeb poměrně hojně využíván jak studenty naší univerzity, tak i lidmi odjinud. Dlouhodobým používáním se objevila řada nedostatků a také nápadů, jak tento systém udělat výkonnějším a příjemnějším na používání. Na začátku roku 2008 jsem se rozhodl, že je vhodná doba udělat další krok směrem k vytvoření kvalitního webového rozhraní pro tvorbu dokumentů. Jakým směrem se $\text{\TeX}$ onWeb ubírá popisuji v tomto článku.

2. Uživatelské rozhraní

První, na co uživatel narazí, ještě než vůbec začne s $\text{\TeX}$ onWeb pracovat, je vzhled uživatelského rozhraní. Naší snahou je modifikovat současný stav tak, aby bylo přehlednější, snadno konfigurovatelné a poskytovalo uživateli co nejvíce prostoru pro vlastní práci. Co je současné verzi hojně vytýkáno, je poměrně velké záhlaví obsahující logo systému a nadpis stránky. Tyto objekty nejsou pro vlastní práci příliš důležité, proto jsme se rozhodli je minimalizovat. Lišta s logem zůstala zachována, ale v mnohem menší velikosti. Tato změna se týká implicitního nastavení.

Protože je vzhled vytvářen pomocí kaskádového stylu, má uživatel možnost vybrat si z několika předdefinovaných vzhledů podle vlastní chuti. K dipozici je nový vzhled preferující větší prostor na pracovní ploše, původní klasický styl a minimální styl vhodný pro pomalejší připojení nebo kapesní počítače. Další styly budou přibývat a v budoucnu bude mít každý uživatel možnost definovat si svůj vlastní styl.

Dalším omezujícím faktorem uživatelského rozhraní je velikost pracovního textového pole, které je vlastně základem celého systému. Právě do něj uživatel zapisuje zdrojový text dokumentu. Textové pole je v nové verzi upraveno tak, aby si uživatel mohl sám jednoduše nastavit jeho velikost pomocí aktivního pravého rohu. Zároveň si může v nástrojové liště určit velikost písma zdrojového textu.

Toto jsou věci patrné na první pohled, nikoliv však nejdůležitější. Hlavní těžiště redesignu systému spočívá ve změně přístupu k systému. $\text{\TeX}$ onWeb vždy nabízel možnost pracovat buď pod vlastním uživatelským jménem a využívat tak možnost jednoduché správy dokumentů a konfigurace pracovního prostředí, nebo pracovat jako anonymní uživatel s implicitním nastavením. Problémy právě s anonymním přístupem a změny, které se objevují v nové verzi, nás vedly k rozhodnutí preferovat autentizovaný přístup. Neznamená to, že již nebude možné používat $\text{\TeX}$ onWeb bez přihlášení. Uživatelé, kteří si nebudou chtít

vytvořit pracovní účet, mohou stále T_EXonWeb používat, ale množina nabízených funkcí bude omezena na pouhý zápis a překlad dokumentu. Ostatní funkce, jako třeba možnost ukládat vytvořenou práci na server, budou dostupné pouze po přihlášení. Bude tak nadále zachována možnost rychlého a bezproblémového vytvoření krátkého dokumentu v T_EXu, ale pro dlouhodobější práci bude vhodné využít vlastní účet.

Co vlastně bude autentizovaný přístup nabízet? Je to možnost vylepšené práce se soubory spočívající především v:

- vytvoření vlastní adresářové struktury pro ukládání souborů,
- mazání, přejmenovávání a přesouvání souborů,
- přímé otevření uloženého souboru v textovém poli,
- download a upload souborů na lokální počítač a
- překlad souboru uloženého na lokálním počítači.

Každý přihlášený uživatel si také bude moci nastavit pracovní prostředí tak, aby vyhovovalo jeho potřebám. Může ovlivnit vzhled pomocí výběru z několika předdefinovaných vzhledů a rozhodnout o tom, které prvky (tlačítka, menu, lišty) se mají zobrazovat a které nikoliv. Stejně tak lze určit implicitní nastavení překladače, které se použije při překladu dokumentu. Je možno ovlivnit nastavení počtu průchodů překladače, možnost vložení nezlomitelných mezer a výběr výstupního formátu dokumentu.

Autentizovaná část systému nabízí i různé podpůrné nástroje usnadňující práci. Jde o možnost zapnutí zvýrazňování syntaxe a číslování řádků pro snazší orientaci ve zdrojovém kódu, možnost použít kontrolu pravopisu, tvorba rejstříku a nástrojová lišta umožňující vkládat pomocí ikon některé příkazy T_EXu.

3. Podpůrné nástroje

Při zápisu zdrojového kódu dokumentu v T_EXu mají začátečníci často problém vyhledat vhodnou značku a správně ji použít. Mnoho z nich je zvyklých používat WYSIWYG systémy, které jim nabízí různá menu a průvodce. Inspirováni tímto faktem, rozhodli jsme se začlenit podobné nástroje i do systému T_EXonWeb. Základem je v podstatě nástrojová lišta obsahující nejpoužívanější příkazy T_EXu uspořádané do logických skupin. Každý příkaz je dostupný přes intuitivní ikonku. Ne všechny příkazy jsou však tak jednoduché jako například vyznačení kurzívního řezu. Při návrhu tabulky je potřeba zapsat poměrně složitou strukturu a leckdy použít i přídatné definiční soubory. V této oblasti mají začínající uživatelé obvykle hodně problémů, protože jsou zvyklí právě na „vizuální“ návrh.

Nástrojová lišta nabízí tedy i možnost použití průvodce pro zápis složitějších objektů. Jedná se především o vkládání obrázků a tabulek, definici rozměrů stránky atd. Průvodce pro tvorbu tabulek umožní uživateli navrhnout strukturu a vzhled tabulky interaktivním způsobem. Protože problematika tabulek v T_EXu

není jednoduchá a existuje řada různých balíčků pokrývajících různé části sazby tabulek, vycházeli jsme při tvorbě průvodce z diplomové práce Petry Talandové (Talandová, 2006), která provedla sumarizaci jednotlivých existujících balíčků a navrhla, jak by mohl takový průvodce vypadat.

Při tvorbě dokumentů se často setkáváme i s potřebou zkontrolovat text, zda neobsahuje různé překlepy nebo pravopisné chyby. Proto T_EXonWeb nabízí možnost využití spellcheckeru aspell, který problematická slova najde a vyznačí ve zdrojovém textu. Z důvodu neinteraktivity tohoto přístupu neposkytuje prozatím možnost přidávání vlastních slov do slovníku, ale snad se brzy podaří i tento nedostatek odstranit.

Systému T_EXonWeb také chyběla podpora tvorby rejstříku. Tento problém stále není definitivně rozřešen, ale pracujeme na tom.

4. Styly dokumentů

I když je T_EXonWeb primárně určen spíše pro nouzové použití, kdy uživatel nemá možnost použít lokální instalaci T_EXu nebo chce napsat pouze kratší text, postupem doby začal být využíván i pro zpracování textů složitějších. Příkladem mohou být studenti naší univerzity, kteří zde překládají své seminární a diplomové práce.

Protože se ukázalo, že jistá uživatelská nepřítulnost není překážkou, rozhodli jsme se tuto tendenci podpořit a nabídnout uživatelům předpřipravené styly a usnadnit jim tak tvorbu různých typů dokumentů. V současné době systém T_EXonWeb nabízí využití stylu pro závěrečné práce, který připravil Jiří Rybička. K dispozici jsou i styly pro posudky závěrečných prací a to jak pro vedoucí, tak i oponenty. Další styl lze použít pro psaní dopisů. Průběžně vznikají styly nové a naším cílem je vytvořit co nejširší skupinu typických dokumentů. Výhoda použití těchto stylů je určitá typografická kvalita, která často dokumentům od nepoučených autorů chybí, a i když uživatel nemusí použít přímo uvedený styl, alespoň se jím může inspirovat.

5. Technické pozadí

Se zvyšující se kvalitou systému T_EXonWeb se zvyšuje i počet uživatelů, kteří tento nouzový způsob práce s T_EXem využívá. V exponovaných chvílích, jako je například konec semestru doprovázený odevzdáváním seminárních a závěrečných prací, se ukazuje, že používaný hardware již nedostačuje svým výkonem. Proto jsme se rozhodli posílit i tuto stránku systému a přecházíme z Intel Celeron 1 GHz, 512 MB operační paměti a 20 GB pevný disk na výkonnější čtyřprocesorový Intel Xeon 3 GHz, 2 GB operační paměti a 200 GB pevný disk. Slibujeme si od toho

zachování dobré přístupnosti a rychlé odezvy systému i při velkém zvýšení počtu uživatelů.

V souvislosti s přechodem na nový hardware jsme se rozhodli využít i kvalitnější operační systém a přešli z *Fedora Linux* na *CentOS*, který je v podstatě volně dostupnou variantou *RedHat Enterprise Linux*. To nám však způsobilo pár problémů s kompatibilitou kódu systému *TeXonWeb*, který je celý psán v programovacím jazyku Perl a využívá různé bezpečnostní prvky zamezující diskreditaci systému. Problémy se objevily především u nástroje *chroot*, který vytváří virtuální adresářovou strukturu pro běh systému.

6. Závěr

Systém *TeXonWeb* pravděpodobně nikdy nebude moci množinou ani kvalitou svých funkcí konkurovat profesionálním editorům, které se běžně používají pro zápis zdrojových textů *TeXu* na lokálních počítačích. Nicméně v situacích, kdy je člověk mimo své pracoviště a potřebuje si vysadit jednoduchý dokument nebo se jedná o začínajícího uživatele, který nemá s *TeXem* zkušenosti, tehdy může *TeXonWeb* prokázat platnou službu.

Faktorem, který nejvíce omezuje plynulý vývoj systému *TeXonWeb* je různorodost webových prohlížečů. Pro práci uživateli sice dostačuje obyčejný webový prohlížeč, ale těch je spousta a každý má jiné renderovací jádro. To však uživatele nezajímá, chce aby vše fungovalo, jak má. Čím více se snažíme zpříjemnit a usnadnit práci s *TeXonWeb* použitím moderních technologií, tím více narážíme právě na tuto rozdílnost. Problémy nastávají hlavně při použití Ajaxu, kaskádových stylů nebo JavaScriptu. Často jsme nuceni hledat řešení, které sice nebude splňovat všechny naše představy, ale alespoň je bude moct použít většina uživatelů. I z toho důvodu není postup prací tak rychlý, jak bych si přál.

Lepší než spousta slov je možnost vyzkoušet si, co a jak vlastně funguje. Protože se systém *TeXonWeb* nevyvíjí skokově, ale plynule, neobjevují se nové vlastnosti vždy najednou a kompletně hotové. Snažíme se přesto nové vlastnosti důkladně testovat a uveřejňovat plně funkční verze. Ne vše, co jsem v tomto článku popsal, je proto již k dispozici v ostré verzi. Vývojová větev, která není úplně stabilní a bezchybná, je však k dispozici všem zájemcům, kteří se chtějí seznámit s novinkami a případně přispět svými připomínkami. Testovat lze na adrese <http://tex.mendelu.cz/devel/>. Ostrá verze systému *TeXonWeb* je k dispozici na adrese <http://tex.mendelu.cz/>.¹

Vývoj systému je čas od času podpořen grantem. Například v letošním roce je součástí projektu IG180631 Interní grantové agentury Provozně ekonomické

¹Poznámka redakce. K vyzkoušení lze doporučit i experimentální webové stránky Hanse Hageny <http://live.contextgarden.net/>, nabízející k 1. září 2008 pdf_{TeX}, X_q_{TeX} a Lua_{TeX} (beta-0.29.0).

fakulty Mendelovy zemědělské a lesnické univerzity v Brně.

I když je systém $\text{T}_{\text{E}}\text{XonWeb}$ nejčastěji prezentován mou osobou, nejsem sám, kdo se na jeho vývoji podílí. Cenné rady a nápady dodává Jiří Rybička, na problematice tabulek spolupracuje Petra Talandová a významný podíl na kódu má Jan Prudil.

7. Literatura

- PŘICHYSTAL, J., RYBIČKA, J. Webové rozhraní pro sazbu dokumentů. In OLŠÁK, P. *SLT 2004 – sborník semináře o Linuxu a T_EXu*. 1. vyd. Brno: Konvoj, ČSTUG, CZLUG, 2004, s. 73–78. ISBN 80-7302-068-8.
- TALANDOVÁ P., *Přístupy ve zpracování tabulek v systémech DTP*, diplomová práce, MZLU 2006.

Summary: Innovation and Enhancement of System $\text{T}_{\text{E}}\text{XonWeb}$

The paper describes the new features of a web interface for typesetting – system $\text{T}_{\text{E}}\text{XonWeb}$. This system has arisen because of the need for the fast and smooth usage of $\text{T}_{\text{E}}\text{X}$. It is first of all about a new user interface, user profiles and document management. New features that will be implemented in the future and the way it develops are also discussed.

Publikování z jednoho zdroje v odlišných formátech pro různá výstupní zařízení

PETR SOJKA, MICHAL RŮŽIČKA

T_EX je tradičně používán jako autorský nástroj pro publikování vědeckých textů a učebnic. V dnešní době jsou z mnoha důvodů čtenáři vyžadovány elektronické publikace souběžně vydávané nejen ve webovém formátu, ale i ve formě pro prohlížení na obrazovce. Tento článek se zabývá způsoby publikování z jednoho zdrojového textu autorsky pořizovaného ve formátu L^AT_EX a ukazuje příklady několika skript publikovaných touto cestou. Zvláštní důraz je přitom kladen na webové dokumenty generované buď do HTML nebo XHTML s matematickými výrazy převedenými do jazyka MathML.

Zmíněno je také „on-the-fly“ generování dokumentů s JBIG2 komprimovanými obrázky pro potřeby digitální matematické knihovny projektu DML-CZ.

Motivace

Discover the outer logic of the typography in the inner logic of the text.

–Robert Bringhurst [4]

Dokumenty předávající informace mají svůj *obsah* a *formu*. Formu (vzhled) by měl zohledňovat *design*, který by měl užívat grafické prostředky konzistentně. Možnosti formy dokumentu jsou vymezeny *výstupním zařízením* (papír, LCD monitor, PDA).

Je dobře známým, ale již méně respektovaným faktem, že design dokumentu musí být znovuvytvořen pro každé nové výstupní zařízení. Mnoho dokumentů vyladěných T_EXovým systémem pro čtení z výtisku na papíře (mikrotypografie atd.) je hrdě vystaveno na webu bez přizpůsobení designu pro konkrétní výstupní zařízení (LCD obrazovka, PDA), pro konkrétní účel a pro specifické požadavky čtenáře. To je v rozporu s cílem, který měl na mysli Knuth např. při návrhu fontů v METAFONTu – dokonce i maličké detaily rasterizace ovlivněné různými tiskárnami by měly být vyladěny správným nastavením pro konkrétní tiskárnu v souboru parametrů `modes.mf`.

Autoři, kteří užívají systém s otevřeným zdrojovým kódem na bázi T_EXu, mají značné možnosti k ovlivnění každého aspektu formy při psaní svých článků a knih. Pokud autoři při psaní používají *pouze* logické značkování, je možné nezávisle

Děkujeme za podporu České akademie věd granty AV1ET208050401 a AV1ET200190513, stejně jako za cestovní příspěvek Československého sdružení uživatelů T_EXu.

zvolit odlišnou typografickou úpravu – respektující obsah a odpovídající různým možnostem výstupních zařízení – změnou vizuální reprezentace jednotlivých logických částí textu. Důsledné oddělení obsahu a formy je možné téměř vždy, pouze s ojedinělými výjimkami, jako je sazba básní Christiana Morgensterna. Je však nutná disciplinovanost autorů při vyhýbání se přímého užití vizuálních příkazů jako je např. `\skip`: důsledná lokalizace vizuálního formátování do stylu sazby je téměř nezbytností.

Jak roste kvalita a rozlišení zobrazovacích zařízení, stále delší texty jsou čteny přímo z obrazovky počítače či z mobilních telefonů nebo PDA. Autoři přirozeně vyžadují, aby obsah jejich dokumentů *byl* přizpůsoben pro širokou řadu různých výstupních zařízení. I když návrh designu dokumentu je podstatně obtížnější pokud musí být vzata v úvahu celá řada těchto výstupních zařízení a prohlížečů, je to přesně to, co je požadováno čtenáři.

V tomto článku zmíníme několik publikačních projektů: dvě matematická skripta [1, 2] a databázové publikování v DML-CZ [11]. Ve všech projektech jsou ukázány výhody plynoucí ze striktního oddělení formy a obsahu pro užití jednoho vstupního zdroje vytvořeného autorem nebo vygenerovaného na požádání z databáze pro různé typy výstupů.

Publikování z jednoho zdroje

If the only tool we know is a word processor, everything looks like a print document.

– Peter Meyer [9]

Autor chce předávat informace. A jediná věc, kterou *musí* udělat, je označit logické prvky v textu. To umožní designerovi vynutit konzistenci: vizuální zobrazení stejných logických částí by mělo být důsledně stejné. Vyznačení logických prvků v textu jeho autorem umožní publikování pro různá výstupní zařízení pouhým přepnutím mezi různými designy. Údržba textu je pak mnohem jednodušší a levnější než údržba různých verzí obsahově stejného textu, které se liší pouze zamýšleným způsobem použití. Také to snižuje počet chyb, zlepšuje konzistenci a/nebo šetří náklady na překlad. Pro tento typ pořizování a zpracování dokumentů se užívá termín *publikování z jednoho zdroje* (*single-source publishing* nebo *single-sourcing*) – http://en.wikipedia.org/wiki/Single_source_publishing. Dobře známým systémem a sadou DTD umožňujících publikování z jednoho zdroje je DocBook, který obsahuje podporu pro konverzi v něm označovaných dokumentů do XHTML, DVI, PS nebo PDF. A to buď za pomoci XSL, XSL-FO, nebo L^AT_EXem.

Publikování z jednoho zdroje často vychází z XML jako zdroje obsahu [9]. Pro mnoho autorů však není dost pohodlné zapisovat XML přímo, a to ani s užitím „chytrého“ XML editoru. Technické rukopisy plné matematiky budou i nadále pořizovány v nějaké formě T_EXu díky své kompaktnosti, přehlednosti

a čistotě matematického zápisu $\text{T}_\text{E}\text{X}$ u. Produktivita autora znatelně vzrůstá s „autor-centrickými“ systémy [8], v případě matematiky např. s $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{L}\text{T}_\text{E}\text{X}$ em.

V akademické oblasti autoři připravují učební materiály pro své předměty sami a chtějí je publikovat ve formátech, které upřednostňují jejich studenti. Metodou publikování z jednoho zdroje jsme proto připravili do různých výstupních formátů dvě matematická skripta [1, 2] ze zdrojového textu autorsky pořízeného v $\text{L}\text{T}_\text{E}\text{X}$ u. V dalších částech popíšeme naše zkušenosti s těmito projekty.

Značkování a konverzní nástroje

Data cannot be used at a finer grain than it is marked up at.

–Rick Jelliffe

Abychom umožnili publikování z jednoho zdroje, museli jsme podstatně pročistit zdrojové soubory, protože ty původně vůbec nebyly psány s cílem publikování do více odlišných formátů. I Donald Ervin Knuth, DEK, psal velmi „nízkoúrovňový“ kód ve svém $\text{T}_\text{E}\text{X}$ booku:

```
&\elevenit I\kern.7ptllustrations by\cr
&DU\kern-1ptANE BIBBY\cr
\noalign{\vfill}
&\setbox0=\hbox{\manual77}%
\setbox2=\hbox to\wd0{\hss\manual6\hss}%
\raise2.3mm\box2\kern-\wd0\box0\cr % A-W logo
&ADDISON\kern.1em--WESLEY\cr
%&PUBLISHING COMP\kern-.13emANY\kern-1.5mm\cr
```

Neočekával totiž, že by kód sloužil pro cokoli jiného než pro přípravu sazby pro tisk na fotosázecím stroji – s danými fonty, kerningem pro danou velikost atd. Pro potřeby publikování z jednoho zdroje musí být hlavní text napsán bez ladění pro konkrétní výstupní zařízení/tiskárnu a nízkoúrovňové značkování musí být nahrazeno vysokoúrovňovým značkováním, které umožňuje více odlišných definic použitých maker pro různé výstupy. Značkování musí být zvoleno s nejlepší možnou granularitou, převoditelné na odpovídající nastavení designu pro každý typ výstupu. Není to nová myšlenka a je užita také ve světě XML (rozdílné nastavení zobrazení pomocí kaskádových stylů CSS pro odlišná zařízení nebo prohlížeče).

Určili jsme několik typů výstupních formátů, které naši studenti požadovali. Vedle standardní verze vhodné pro tisk na papír byla požadována prohledávatelná verze dokumentu optimalizovaná pro LCD obrazovku s poměrem stran 4:3. Pro některé účely byla potřebná i (X)HTML verze dokumentů pro platformy a zařízení bez PDF prohlížečů. Nakonec jsme připravili také XHTML+MathML verzi skript, což je vhodný formát např. pro nevidomé.

Existuje mnoho nástrojů a pomůcek pro konverzi $\text{\TeX}$ ových dokumentů do různých výstupních formátů – seznam některých z nich je dostupný na webové stránce TUGu *TeX Resources on the Web* (<http://www.tug.org/interest.html>). PDF $\text{\LaTeX}$ s balíčky **hyperref** a **crop** je vhodná kombinace pro tiskový výstup. Pro verzi dokumentu pro prohlížení na obrazovce jsme zvolili balíček **pdfscreen** v kombinaci s PDF $\text{\TeX}$ em a balíčkem **hyperref**.

PDF verze

Make all visual distinctions as subtle as possible, but still clear and effective.

– Edward R. Tufte [17]

Pro každou verzi výstupu byly parametry designu a maker zapsány v oddělené podmíněné větvi. Ukázka jednoduchého zdrojového kódu:

```
\newif\ifprint
\printfalse % Obrazovková verze.
%\printrtrue % Tisková verze.

\ifprint
  \hypersetup{colorlinks=false, pdfborder={0 0 0}}
  % Centrovaný zrcadlený dokument na papíru
  % formátu A4 s přidáním ořezových značek.
  \usepackage[cam,a4,center,mirror]{crop}
\fi
```

PDF pro tisk

Pro potřeby tisku je vhodné připravit výstup v odstínech šedi. K tomuto účelu je často užíván balíček **hyperref** s odpovídajícím nastavením (**colorlinks=false**, **pdfborder={0 0 0}**).

Balíček **crop** (dostupný na CTAN) je dobrá volba pro přidání ořezových značek do dokumentu. Balíček **crop** je také schopen provést některé transformace dokumentu, jako je jeho zrcadlení apod. Výstup je ukázán na obrázku 1 na straně 123.

Výhodou balíčku **crop** je to, že tyto transformace je schopen s dokumentem provádět dodatečně, bez redefinice geometrie dokumentu ze strany uživatele. Je snadné vzít již existující, finálně zalomený dokument, a přidáním jediného řádku vytvořit jeho tiskovou variantu bez nutnosti hlubších zásahů, a aniž by bylo nutné cokoli znovu ladit nebo by hrozilo nebezpečí změny zlomu dokumentu.

PDF pro prohlížení na obrazovce

Typický LCD monitor je úplně odlišné výstupní zařízení než tiskový výstup. V porovnání s dnešními domácími tiskárnami s rozlišením 1 200 a více DPI

má obrazovka řádově nižší rozlišení. U elektronické publikace můžeme využít interaktivních možností, které nám tento formát nabízí – v dokumentu je možné používat hypertextové odkazy, barvy, navigační lišty atd.

Pro obrazkovou verzi učebnic jsme použili balíček `pdfscreen` (dostupný na CTAN). Kód pro `pdfscreen` byl spolu s definicemi prostředí a maker pro obrazovku opět v oddělené podmíněné větvi `stylopisu`.

Abychom se vyhnuli náročnému opakovanému ladění řádkového zlomu, definovali jsme geometrii tiskového zrcadla obrazkové verze dokumentu tak, aby byl řádkový zlom identický jako v tiskové verzi. Vzhledem k značně odlišné geometrii stránky je však odlišný stránkový zlom, jak je ukázáno na obrázku 2 na straně 124. Zachování stejného řádkového zlomu znamenalo velkou úsporu času a námahy sazeče, bohužel neexistuje způsob, jak toto provést automaticky. Nastavení odpovídající geometrie tiskového zrcadla tedy stejně vyžadovalo jisté úsilí.

Webové verze dokumentu

Hitem dneška je XML – a jednou z oblastí využití XML technologií je dnešní web, který je také místem, kam ve formátu (X)HTML míří značná část dnes produkovaných dokumentů. Pro vytváření prohledávatelných a škálovatelných matematických zápisů pro web je pak vhodným standardem XML jazyk MathML.

Pro převod $\text{\TeX}$ ových dokumentů do webových formátů je k dispozici několik nástrojů. Patří mezi ně např.:

- $\text{\TeX}$ 2page (<http://www.ccs.neu.edu/home/dorai/tex2page/>),
- Tralics (<http://www-sop.inria.fr/apics/tralics/>),
- $\text{\TeX}$ 4ht (<http://www.cse.ohio-state.edu/~gurari/TeX4ht/>),
- $\text{\LaTeX}$ XML (<http://dlmf.nist.gov/LaTeXML/>) nebo
- $\text{\LaTeX}$ 2HTML (<http://www.latex2html.org/>).

Každý nástroj má některé výhody, ale i nevýhody; my zde nechceme rozebírat všechny z nich. Po otestování některých z těchto nástrojů jsme pro přípravu skript vybrali program $\text{\TeX}$ 4ht [3,6], který nabízí převod jak do HTML, tak do XHTML. Navíc podporuje i převod matematiky do značkování MathML.

Značnou výhodou $\text{\TeX}$ 4ht je, že využívá nativní překladač $\text{\TeX}$ u pro přípravu standardních DVI výstupních souborů, které však obsahují `\special` příkazy pro následné zpracování. Díky tomu nehrozí nebezpečí „nepochopení“ nepodporovaných příkazů jako např. v případě $\text{\LaTeX}$ 2HTML. Teprve z takto obohaceného DVI jsou následně vygenerovány (X)HTML stránky dalšími nástroji $\text{\TeX}$ 4ht.

HTML

Nejobtížnější částí konverze byla příprava webového výstupu. V případě komplexních dokumentů je obvykle nutné provést některé změny ve zdrojovém kódu. Již existující logické značkování L^AT_EXu bylo využito kdekoli to bylo možné. Tyto modifikace a příkazy specifické pro T_EX4ht byly opět v odděleném stylovém souboru.

Základní použití T_EX4ht je velmi jednoduché. Když máme zdrojový kód ve formátu L^AT_EX s načteným T_EX4ht stylem, můžeme vyzkoušet provést konverzi dokumentu. Nejdříve můžeme vyzkoušet příkaz:

```
htlatex dokument.tex 'html'
```

Pokud T_EX4ht úspěšně dokončí svou práci, výsledkem je HTML podoba našeho původního dokumentu. S tímto nastavením jsou komplikovanější matematické vzorce vykresleny do PNG obrázků, jak můžete vidět na obrázku 3 na straně 125. Tato cesta je oproti MathML zápisu jednoduchá a bezpečná pro čtení všemi webovými prohlížeči, včetně jejich starších verzí.

Pokud není specifikováno jinak, tak je celý dokument převeden do jediného HTML souboru. T_EX4ht je však schopen dlouhý dokument automaticky rozdělit do stromové struktury vzájemně propojených webových stránek. Volání překladu příkazem `htlatex dokument.tex 'html,2'` vygeneruje dokument s každou kapitolou v odděleném souboru. Navigace mezi kapitolami je pak možná pomocí lišty s odkazy, která je přidána na horní a spodní kraj každé stránky.

Při generování HTML výstupu může být užitečné mít možnost do dokumentu přímo vložit nějaký HTML kód. To provedeme příkazem `\HCode{Nějaký HTML kód.}`. Ten může být použit také pro vložení CSS kódu.

Pro definici CSS jsme užili příkaz `\Css{Definice CSS}`. CSS atributy jsou mapovány na patřičné elementy dokumentu pomocí kódu vkládaného do výstupu pomocí obecného příkazu `\HCode`. Příkaz `\ConfigureEnv` pak zajišťuje pohodlnou definici kódu vkládaného před a za obsah odpovídajícího L^AT_EXového okolí, a to bez nutnosti zásahu přímo do nativní definice okolí.

Jako trošičku složitější příklad definice CSS může posloužit prostředí `veta`:

```
\newtheorem{veta}{Věta}[chapter]
...
\ifweb % Při vytváření webového výstupu...
  \Css{% Definice CSS kódu
    .veta { background-color: \#FFFFFF;
            border: 1px solid;
            border-color: \#0000FF; } }
% Ve značkování výsledného HTML dokumentu je před každé prostředí 'veta'
% umístěna značka '<div class="veta">', za prostředí je umístěna
% značka '</div>'.
\ConfigureEnv{veta}
  {\HCode{<div class="veta">}}
```

```

{\HCode{</div>}}
{}{}

\fi
...
% V dokumentu je užito stejné LaTeXové značkování pro všechny verze
% prostředí 'veta'.
\begin{veta}
    Funkce~ $f(x,y)$  je spojitá v-bodě- $[x_{\{0\}},y_{\{0\}}]$ .
\end{veta}

```

XHTML + MathML

$\TeX$ je velmi často používán pro sazbu vědeckých textů, ve kterých je užito velké množství matematických zápisů. Pokud mají být výstupní XHTML dokumenty maximálně využitelné, tak mnohem zajímavější alternativou k HTML s matematikou v obrázcích je rozšíření XHTML o značkování XML jazyka MathML. Nejkomplikovanější konverzní proces, který jsme na matematických skriptech vyzkoušeli, byla právě konverze do XHTML + MathML.

Bohužel, při užití MathML jsou zde v současnosti jisté obtíže nejen pro autory, ale i uživatele. Zaprvé, implementace MathML ve webových prohlížečích je roztržštěná. Nejlepších výsledků jsme s $\TeX$ 4ht dosáhli při použití volby překladu **mozilla** v kombinaci s webovým prohlížečem Mozilla Firefox (nebo jiným prohlížečem založeným na jádře Gecko) použitým pro prohlížení výsledného dokumentu. Zadruhé, uživatelé musí mít na svém počítači nainstalovány odpovídající matematické fonty. Pro uživatele prohlížeče Mozilla Firefox jsou informace o potřebných fontech, odkazy k jejich stažení a instalační pokyny dostupné z webové stránky *MathML in Mozilla* (<http://www.mozilla.org/projects/mathml/fonts/>).

Velkou výhodou $\TeX$ 4ht je mimo jiné také možnost MathML výstupu, který je velmi užitečný v případě matematických textů – generování XHTML + MathML je přitom velmi obdobné generování HTML:

```
htlatex dokument.tex 'xhtml,mozilla'
```

Pokud vše proběhne, jak má, obdržíme po skončení překladu XML soubor obsahující XHTML kód, který používá jazyk MathML pro vyjádření matematických zápisů. Výsledek, plně škálovatelný dokument, můžete vidět na snímku okna prohlížeče Mozilla Firefox na obrázku 4 na straně 126.

Komplikací pro autory je mimo jiné to, že $\TeX$ 4ht je při generování MathML velmi citlivý na čistotu zápisu matematiky ve zdrojovém textu dokumentu. Např. zápis

```
 $M=\{x|x\$ \text{ je liché } \$\}\$$ 
```

je korektní $\TeX$ ový zápis. $\TeX$ 4ht ale v tomto případě nemá informaci o párování složených závorek. Je tedy nutné použít správnější zápis:

```
 $M=\{x|x \ \mbox{je liché}\}\}$ 
```

Toto byla značná komplikace při převodu matematických skript, která původně nebyla psána s ohledem na tuto potřebu překladače.

Pro $n = 2$ budeme místo $f(x_1, x_2)$ psát $f(x, y)$ a pro $n = 3$ místo $f(x_1, x_2, x_3)$ píšeme $f(x, y, z)$.

Příklad 1.1. i) Zobrazte v rovině definiční obor funkce

$$f(x, y) = \sqrt{\left(x^2 + \frac{(y-2)^2}{4} - 1\right)(x^2 + y^2 - 6x)}.$$

Řešení. Výraz pod odmocninou musí být nezáporný, tj. musí být splněna podmínka

$$\left(\frac{(y-2)^2}{4} + x^2 - 1\right)(x^2 + y^2 - 6x) \geq 0.$$

To nastane, právě když

$$\frac{(y-2)^2}{4} + x^2 - 1 \geq 0 \quad \text{a} \quad (x^2 + y^2 - 6x) \geq 0$$

nebo

$$\frac{(y-2)^2}{4} + x^2 - 1 \leq 0 \quad \text{a} \quad (x^2 + y^2 - 6x) \leq 0.$$

Rovnice $\frac{(y-2)^2}{4} + x^2 = 1$ je rovnicí elipsy se středem v bodě $[0, 2]$ a poloosami délek $a = 1$ a $b = 2$, rovnice $x^2 + y^2 - 6x = 0$ je rovnicí kružnice se středem v bodě $[3, 0]$ a poloměrem $r = 3$, neboť tuto rovnici lze převést na tvar $(x-3)^2 + y^2 = 9$. Množina všech bodů $[x, y] \in \mathbb{R}^2$ splňující výše uvedené nerovnosti, tj. definiční obor funkce f , je znázorněna na vedlejším obrázku. Je to uzavřená množina v $\mathbb{R}^2$.

ii) Zobrazte v rovině definiční obor funkce

$$f(x, y) = \arccos(x^2 + y^2 - 1) + \sqrt{|x| + |y| - \sqrt{2}}.$$

Řešení. Definičním oborem funkce $\arccos$ je interval $[-1, 1]$, první sčítanec je tedy definován pro $[x, y]$ splňující nerovnosti

$$-1 \leq x^2 + y^2 - 1 \leq 1,$$

Obrázek 1: Tiskový výstup (bez zrcadlení stránek)

To nastane, právě když

$$\frac{(y-2)^2}{4} + x^2 - 1 \geq 0 \quad \text{a} \quad (x^2 + y^2 - 6x) \geq 0$$

nebo

$$\frac{(y-2)^2}{4} + x^2 - 1 \leq 0 \quad \text{a} \quad (x^2 + y^2 - 6x) \leq 0.$$

obr. 1.1 Definiční obor funkce f

Rovnice $\frac{(y-2)^2}{4} + x^2 = 1$ je rovnicí elipsy se středem v bodě $[0, 2]$ a poloosami délek $a = 1$ a $b = 2$, rovnice $x^2 + y^2 - 6x = 0$ je rovnicí kružnice se středem v bodě $[3, 0]$ a poloměrem $r = 3$, neboť tuto rovnici lze převést na tvar $(x - 3)^2 + y^2 = 9$. Množina všech bodů $[x, y] \in \mathbb{R}^2$ splňující výše uvedené nerovnosti, tj. definiční obor funkce f , je znázorněna na obrázku 1.1. Je to uzavřená množina v $\mathbb{R}^2$.

[Titulní strana](#)

[Obsah](#)

[Výsledky cvičení](#)

[Rejstřík](#)

◀◀

▶▶

◀

▶

Strana 27 z 407

[Zpět](#)

[Vpřed](#)

[Zavřít](#)

[Konec](#)

Obrázek 2: Obrazovkový výstup

Ve složitých případech může být přepis matematického zápisu velmi komplikovaný, nebo dokonce nemožný. Pro tyto vzácné, ale přesto hrozící případy $\text{\TeX}$ 4ht nabízí nouzové řešení – uživatel si může u určitého úseku kódu vynutit jeho obrazkovou reprezentaci ve výstupním dokumentu.

```
\ifweb
  \Picture*{}
\fi
$M=\{x|x\$ je liché \$\}$
\ifweb
  \EndPicture
\fi
```

Toto řešení není omezeno pouze na zápis matematiky, ale může být užito pro libovolný objekt s problematickou reprezentací v (X)HTML/MathML.

$$\left(\frac{(y-2)^2}{4} + x^2 - 1\right)(x^2 + y^2 - 6x) \geq 0.$$

To nastane, právě když

$$\frac{(y-2)^2}{4} + x^2 - 1 \geq 0 \quad \text{a} \quad (x^2 + y^2 - 6x) \geq 0$$

nebo

$$\frac{(y-2)^2}{4} + x^2 - 1 \leq 0 \quad \text{a} \quad (x^2 + y^2 - 6x) \leq 0.$$

Rovnice $\frac{(y-2)^2}{4} + x^2 = 1$ je rovnicí elipsy se středem v bodě $[0,2]$ a poloosami délek $a = 1$ a $b = 2$, rovnice $x^2 + y^2 - 6x = 0$ je rovnicí kružnice se středem v bodě $[3,0]$ a poloměrem $r = 3$, neboť tuto rovnici lze převést na tvar $(x-3)^2 + y^2 = 9$. Množina všech bodů $[x,y] \in \mathbb{R}^2$ splňující výše uvedené nerovnosti, tj. definiční obor funkce f , je znázorněna na vedlejším obrázku. Je to uzavřená množina v $\mathbb{R}^2$.

ii) Zobrazte v rovině definiční obor funkce

$$f(x, y) = \arccos(x^2 + y^2 - 1) + \sqrt{|x| + |y| - \sqrt{2}}.$$

Řešení. Definičním oborem funkce $\arccos$ je interval $[-1,1]$, první sčítanec je tedy definován pro $[x,y]$ splňující nerovnosti

$$-1 \leq x^2 + y^2 - 1 \leq 1,$$

tj.

$$0 \leq x^2 + y^2 \leq 2,$$

Obrázek 3: HTML výstup

Publikování digitalizovaných textů

Druhý projekt, kde je využívána cesta publikování z jednoho primárního zdroje, je projekt DML-CZ <http://dml.cz> [11, 15]. Cílem projektu není jen digitalizace 250 000 stran českých a slovenských matematických časopisů, ale také poskytnutí nástrojů pro souběžné generování tiskových a pro web optimalizovaných verzí nových, digitálně pořizovaných čísel časopisů.

Zpracování začíná naskenováním převážně bitonálních TIFF obrázků stránek dokumentů v rozlišení 600 DPI. Následný OCR proces je prováděn v několika fázích za užití programů FineReader (OCR textu) a InftyReader (OCR matematiky) [12, 14]. FineReader pro výstup OCR umí využít uložení více vrstev do PDF, a pod naskenovaný obrázek stránky umí uložit rozpoznaný text (pro vyhledávání, indexování). InftyReader [16] umí takové PDF načíst a je scho-

$$\left(\frac{(y-2)^2}{4} + x^2 - 1\right)(x^2 + y^2 - 6x) \geq 0.$$

To nastane, právě když

$$\frac{(y-2)^2}{4} + x^2 - 1 \geq 0 \quad \text{a} \quad (x^2 + y^2 - 6x) \geq 0$$

nebo

$$\frac{(y-2)^2}{4} + x^2 - 1 \leq 0 \quad \text{a} \quad (x^2 + y^2 - 6x) \leq 0.$$

Rovnice $\frac{(y-2)^2}{4} + x^2 = 1$ je rovnicí elipsy se středem v bodě $[0, 2]$ a poloosami délek $a = 1$ a $b = 2$, rovnice $x^2 + y^2 - 6x = 0$ je rovnicí kružnice se středem v bodě $[3, 0]$ a poloměrem $r = 3$, neboť tuto rovnici lze převést na tvar $(x-3)^2 + y^2 = 9$. Množina všech bodů $[x, y] \in \mathbb{R}^2$ splňující výše uvedené nerovnosti, tj. definiční obor funkce f , je znázorněna na vedlejším obrázku. Je to uzavřená množina v $\mathbb{R}^2$.

ii) Zobrazte v rovině definiční obor funkce

$$f(x, y) = \arccos(x^2 + y^2 - 1) + \sqrt{|x| + |y| - \sqrt{2}}.$$

Rěšení. Definičním oborem funkce $\arccos$ je interval $[-1, 1]$, první sčítanec je tedy definován pro $[x, y]$ splňující nerovnosti

$$-1 \leq x^2 + y^2 - 1 \leq 1,$$

Obrázek 4: MathML výstup

pen exportovat text včetně matematických strukturních vzorců v L^AT_EXu nebo v MathML spolu s pozičními informacemi o umístění objektů. To případně umožní vysázet rozpoznaná L^AT_EXová a/nebo MathML data jako další vrstvy PDF pod obrázky, například PDFL^AT_EXem za použití standardního prostředí `picture`. Pro vytvoření prohlédávatelných dokumentů v různých jazycích je nutné použít balíček `cmap`.

Velikost vygenerovaných souborů je důležitá – pro tisk je obvykle požadováno rozlišení tiskového zařízení, pro prohlížení na obrazovce je dostačující nižší rozlišení. PDF nabízí odlišné typy kompresních filtrů – od verze 1.4 PDF standardu je podporována komprese bitonálních obrázků metodou JBIG2. PDF_TE_X ve svých posledních verzích JBIG2 komprimované obrázky podporuje také. Navíc Adam Langley napsal open-source konverzní program `jbig2enc` jako projekt podporovaný společností Google [7].

JBIG2 nabízí jak bezztrátovou, tak ztrátovou kompresi. Pro naskenovaný obsah je nejvhodnějším postupem k získání nejlepšího kompresního poměru přijmutí drobných chyb v obrazových datech užitím symbolického kódování, kde variace vznikají z tiskových chyb. Kompresní poměr JBIG2 závisí na velikosti kontextu – lepší výsledky jsou dosahovány, pokud se komprimují celé rozsahy stran místo jediné oddělené stránky. Při užití lehce ztrátové komprese a velké velikosti kontextu jsme schopni generovat PDF obrázky, které mají zhruba asi jen 10–20% velikost oproti CCITT kódovaným a LZW komprimovaným obrázkům. Parametry programu `jbig2enc` umožňují finální vyladění regenerace obrázků, a to i pro jejich „on-the-fly“ generování [5], které bylo implementováno pro OSS systém Kramerus.

Nové články časopisů u digitální knihovny se již samozřejmě naskenují, ale přebírají sazenu (T_EXem). Pro sazbu časopisu *Archivum Mathematicum* [10] se zvažuje i generování nových článků v XHTML+MathML spolu s tiskovou verzí v PDF.

Jelikož se jak formát PDF, tak požadavky a možnosti čtenářů mění, základním principem je uchovávat (a editovat) pouze *jedna primární data* a metadata, a všechny aktuálně vystavované výstupy z nich generovat automatizovanými procedurami a programy. Tak jsou například před závěrečným importem do digitální knihovny DSpace stránky článku spojeny do jednoho PDF, přigenerována úvodní stránka s metadaty a citačními informacemi, PDF je optimalizováno do aktuálně vhodné verze PDF s již široce akceptovanými kompresními filtry a výsledné PDF určené k šíření je digitálně podepsáno prostředky PKI infrastruktury. To vše se děje skripty, které se dají spouštět opakovaně po rozsáhlejších úpravách primárních (meta)dat, případně se dají vytvářet varianty generujících skriptů pro různá užití výstupů (tagged PDF, PDF optimalizovaná pro web ap.).

Závěr

Na příkladech několika projektů jsme ukázali, že na T_EXu založené pořizování autorského textu a publikování z jednoho zdroje či jedné primární dat je velmi přirozená a efektivní cesta přípravy a personalizace dokumentů pro různá výstupní zařízení. T_EX4ht je velmi konfigurovatelný nástroj pro webové publikování z T_EXových zdrojů, JBIG2 formát pro bitonální obrázky značně šetří diskové a přenosové kapacity bez újmy na kvalitě, pokud se generuje velké množství obrázků jako tomu je v případě digitalizačních projektů jako DML-CZ.

Reference

- [1] Zuzana Došlá and Ondřej Došlý. *Metric Spaces: Theory and Examples (in Czech)*. Masaryk University in Brno, second edition, 2000.

- [2] Zuzana Došlá, Roman Plch, and Petr Sojka. Matematická analýza s programem Maple: 1. Diferenciální počet funkcí více proměnných (Mathematical Analysis with Program Maple: 1. Differential Calculus). CD-ROM, <http://www.math.muni.cz/~plch/mapm/>, December 1999.
- [3] Michel Goossens, Sebastian Rahtz, Ross Moore, and Bob Sutor. *The L^AT_EX Web Companion*. Addison-Wesley, Reading, MA, 1999.
- [4] Philip Babcock Gove and Merriam Webster. *Webster's Third New International Dictionary of the English language Unabridged*. Merriam-Webster Inc., Springfield, Massachusetts, U.S.A, January 2002.
- [5] Miroslav Grabovský. Generování PDF pro systém Kramerius. Master's thesis, Masaryk University, Brno, Faculty of Informatics, January 2007.
- [6] Eitan M. Gurari. TeX4ht: L^AT_EX and T_EX for Hypertext. <http://www.cse.ohio-state.edu/~gurari/TeX4ht/>, February 2005.
- [7] Adam Langley and Dan S. Bloomberg. Google Books: Making the public domain universally accessible. In *Proceedings of SPIE — Volume 6500, Document Recognition and Retrieval XIV*, pages 1–10, San Jose, CA, January 2007. The International Society of Optical Engineering. <http://www.imperialviolet.org/binary/google-books-pdf.pdf>.
- [8] Peter Meyer. Planning a single source publishing application for business documents, 2005. http://www.elkera.com/cms/articles/seminars_and_presentations/.
- [9] Peter Meyer. Introduction to single source publishing, 2006. http://www.elkera.com/cms/articles/technical_papers/.
- [10] Michal Růžička. Automated Processing of T_EX-typeset Articles for a Digital Library. In Sojka [13], pages 167–176.
- [11] Petr Sojka. From Scanned Image to Knowledge Sharing. In Klaus Tochtermann and Hermann Maurer, editors, *Proceedings of I-KNOW'05: Fifth International Conference on Knowledge Management*, pages 664–672, Graz, Austria, June 2005. Know-Center in coop. with Graz Uni, Joanneum Research and Springer Pub. Co.
- [12] Petr Sojka. Towards Digital Mathematical Library: Optical Character Recognition on Mathematical Texts. In Julius Štuller and Zdenka Linková, editors, *Inteligentní modely, algoritmy a nástroje pro vytváření sémantického webu*, pages 110–113, Praha, Czech Republic, 2006. Ústav informatiky AV ČR.
- [13] Petr Sojka, editor. *Towards Digital Mathematics Library—Proceedings of DML 2008*, Birmingham, UK, July 2008. Masaryk University.
- [14] Petr Sojka, Radovan Panák, and Tomáš Mudrák. Optical Character Recognition of Mathematical Texts in the DML-CZ Project. Technical report, Masaryk University, Brno, September 2006. presented at CMDE 2006 conference in Aveiro, Portugal.
- [15] Petr Sojka and Jiří Rákosník. From Pixels and Minds to the Mathematical

- Knowledge in a Digital Library. In Sojka [13], pages 17–27.
- [16] Masakazu Suzuki, Fumikazu Tamari, Ryoji Fukuda, Seiichi Uchida, and Toshihiro Kanahori. INFITY — An integrated OCR system for mathematical documents. In C. Vanoirbeek, C. Roisin, and E. Munson, editors, *Proceedings of ACM Symposium on Document Engineering 2003*, pages 95–104, Grenoble, France, 2003. ACM.
- [17] Edward R. Tufte. *Visual Explanations*. Graphics Press LLC, 1997.

Summary: Parallel Electronic Publications

T_EX is traditionally used as an authoring tool for the paper publishing of scientific texts and textbooks. Parallel electronic publications that are meant for on-screen viewing and web delivery are also demanded by readers for many reasons today. This paper discusses the ways to single-source author publishing from a L^AT_EX source file, and it shows examples of several textbooks published by this approach. Special attention is given to the web document generation either to HTML or XHTML markup with a notation translated to MathML. Also discussed is a personalised automated document generation for a digital library project DML-CZ.

*Fakulta informatiky Masarykovy univerzity
Botanická 68a, 602 00 Brno, Česká republika
<http://www.fi.muni.cz/usr/sojka/>
sojka@fi.muni.cz, xruzick7@fi.muni.cz*

DocBy.T_EX je makro T_EXu, které umožní jednoduše dokumentovat zdrojové texty v programovacích jazycích, například v jazyku C. Obsahuje možnost vkládání vymezených úseků zdrojových textů a díky encT_EXu umí automaticky zvýraznit dokumentovaná slova v těchto textech a udělat z nich aktivní odkazy. Automaticky vytváří obsah a rejstřík. K vytvoření cílové dokumentace v PDF formátu je potřeba pouze pdfT_EX doplněný encT_EXem. Není nutné použít žádné preprocesory.

1. Úvod, motivace, zadání

Můj syn Mirek píše ročníkovou práci z programování a je zřejmé, že pokud to bude chtít odevzdat, měl by své zdrojové kódy nějak dokumentovat. Nabídl jsem se mu, že mu pro tyto účely napíšu T_EXové udělátko. Snažil jsem se splnit následující cíle

- Formát dokumentace je klikací PDF. Dokumentaci ve formátu html negenerujeme.
- Věc by měla být co nejjednodušší, tj. pokud možno jen použití T_EXu. Případné preprocesory by práci zbytečně komplikovaly.
- K napsanému zdrojovému kódu se může psát dokumentace zvlášť do „vedlejšího“ T_EXového souboru, v němž budou příkazy na vkládání vymezených úseků zdrojového kódu podle potřeby. Kód bude tedy vložen v době zpracování dokumentace T_EXem aktuální. Tj. přesně ten, který načítá kompilátor při překladu programu.
- Zdrojový kód je členěn do modulů. Kompilátor je překládá do objektů *.o a ty pak sestavuje do výsledného programu. Jeden modul z pohledu programátora řeší jeden ucelený problém a stojí za to jej dokumentovat jako „samostatnou jednotku“. T_EXové soubory s dokumentací tedy respektují strukturu členění na moduly.
- Pomocí encT_EXu je možné zařídit, že libovolný výskyt dokumentovaného slova v začleněném zdrojovém kódu se stane automaticky aktivním odkazem na místo, kde je slovo dokumentováno. Dokumentovaným slovem může být funkce, struktura, proměnná nebo cokoli jiného.
- DocBy.T_EX sám vytvoří pod čarou na stránce, kde je slovo dokumentováno, seznam všech stránek, kde se slovo vyskytuje napříč celou dokumentací.

Podobné seznamy stránek vznikají v rejstříku.

- Rejstřík vzniká ze všech dokumentovaných slov zcela automaticky uvnitř $\text{\TeX}$ u. Tj. bylo potřeba implementovat makrojazykem $\text{\TeX}$ u abecední třídění.
- Členění na sekce a podsekce a automatické vytvoření obsahu, který je klikací, je samozřejmostí.
- Věc funguje v csplainu, protože já i syn používáme tento formát.

Přiznám se bez mučení, že možnosti současných nástrojů na dokumentování zdrojových kódů jsem moc nestudoval. Je tedy možné, že v některých aspektech znovu vynalézám kolo a že některé vlastnosti, které se běžně při dokumentování používají, chybí. Je to taková „rodinná“ zakázka na míru programu, který syn odevzdává jako ročníkovou práci. Nicméně jsem se snažil nástroj udělat aspoň trochu obecně použitelný a dovolím si jej na konferenci $\text{\TeX}$ perience 2008 předvést. Posluchači sami posoudí, zda jim to může být k něčemu dobré. Obecnou použitelnost deklaruji například tím, že dokumentaci k DocBy. $\text{\TeX}$ u (včetně technické dokumentace s rozбором všech maker) píšu v DocBy. $\text{\TeX}$ u.

Zvažoval jsem aspoň na vteřinu doporučit synovi literární programování, jak jej vytvořil Donald Knuth (tzv. WEB). Ovšem, velmi rychle jsem od toho upustil. Domnívám se, že doba, kdy Knuth vytvářel své dílo, byla poznamenána poněkud jinými počítači, než jaké máme před sebou dnes. Dnes je běžné grafické rozhraní, ve kterém je možné současně otevřít více editorů vedle sebe v různých oknech s různými texty. V jednom třeba může být text určený pro počítač a v druhém text, který je určen pro člověka. Vše je vidět naráz, jak může vidět i čtenář tohoto článku na obrázku 1. Není tedy nutné tyto informace slučovat do jediného souboru, aby byly blízko sebe, a poté je preprocesory (tangle, weave) zase oddělovat. Dnešní programátoři také obvykle přemýšlejí v intencích, jak kód zpracovává kompilátor. Trhání tohoto kódu na úseky (jako v Knuthově WEBu) asi moc nemilují.

Poznamenávám ještě, že jsem Knuthův WEB nezavrhl z neznalosti. Sám jej velmi dobře znám. Vrtal jsem se poměrně do hloubky ve zdrojových kódech $\text{\TeX}$ u a také jsem napsal jednoduchoučký program vlna v CWEBu. Synův program je ovšem asi stokrát složitější než program vlna, takže rozhodování o CWEBu ani moc dlouho netrvalo. CWEB jsme nepoužili.

Protože syn bude odevzdávat svou ročníkovou práci až v lednu roku 2009, nečekejte nyní hotové dílo. Jeho program je ve stádiu zrodu a DocBy. $\text{\TeX}$ se také zatím vyvíjí podle požadavků programátora. V době, kdy se snažím aspoň částečně dodržet termín odevzdání tohoto příspěvku (srpen 2008) tedy není DocBy. $\text{\TeX}$ ve stádiu, kdy bych jej mohl zveřejnit na internetu. Ani ukázek zatím moc není. Syn má naprogramováno v tuto chvíli asi 20 modulů svého programu a bude přidávat ještě mnoho dalších. Dokumentovány DocBy. $\text{\TeX}$ em má moduly dva. Na obrázcích 2, 3 můžete posoudit, jak vypadá současná verze dokumentace jeho programu. Na obrázku 4 pak vidíte stránku z dokumentace

Obrázek 1: Pracovní plocha s více editory

samotného DocBy.T_EXu.

Až DocBy.T_EX dospěje do nějaké rozumně stabilní verze, najdete jej na internetu na <http://www.olsak.net/docbytex.html>.

2. Příklad použití DocBy.T_EXu

Předpokládá se, že program je členěn na moduly. Každý modul obsahuje skupinu funkcí, která řeší jeden ucelený problém. Předpokládejme teoretický program lup, který má tři moduly: main v souboru main.c, win v souborech win.c a win.h a konečně modul base v souborech base.h a base.c. Dokumentace k takovému programu vznikne tak, že autor napíše hlavní T_EXový soubor (například lup.tex) a z něj bude číst další T_EXové soubory, které obsahují dokumentaci k jednotlivým modulům: main.d, win.d a base.d. V lup.tex může být nějaký rozverný úvod a dále příkazem „\module“ jsou načítány jednotlivé dokumentační soubory k mo-

```
10: #define HASHSIZE 233 // velikost hashového pole
```

hash.c

Délka pole se podle doporučení volí jako prvočíslo zhruba tak velké, jak velký očekáváme slovník. Hodnoty pole jsou pointery do struktury `hash`:

```
14: static struct hash{
15:   char typ; // informace, na zaklade ktere se hleda
16:   char *string; // informace, na zaklade ktere se hleda
17:   kq_pin informace; // informace, ktera se hleda
18:   struct hash *next; // dalsi policko hashe (je pouzito separate chaining)
19: } *hashtable[HASHSIZE];
```

hash.c

Každý klíč dostane přiděleno pomocí hashovací funkce číslo v rozsahu $0..HASHSIZE-1$, které je indexem v poli `hashtable`. Pokud se sejdou dva nebo více různých klíčů se stejnou hodnotou hashovací funkce, je nutno řetězit jednotlivé struktury `hash`, jak ukazuje obrázek. Pole `hashtable` o velikosti `HASHSIZE` je na obrázku zakresleno svisle vlevo. Např. klíč1 má na obrázku hashovací index 3 stejně jako klíč5. Má-li se vyhledat hodnota ke klíč5, program vyhodnotí index 3 a dále projde „řádek“ na obrázku sekvencně.

Funkce `inithash` vynuluje `hashtable` a použije se na začátku programu.

```
32: void inithash () // Vynuluje hashove pole.
33: {
34:   int i;
35:   for(i=0; i<HASHSIZE; i++){
36:     hashtable[i] = NULL;
37:   }
38: }
```

hash.c

Funkce `hashfunc` přidělí ke klíči hashovací hodnotu, tj. index do pole `hashtable`. Protože klíč je v našem případě dvojice `[typ, slovo]`, má funkce odpovídající parametry `typ, s`.

```
42: static int hashfunc (int typ, char *s) // Spocita cislo ze stringu
43: {
44:   int i, vysledek;
45:
46:   vysledek=typ*5;
47:   for(i=0; (unsigned char)s[i]; i++)
48:     vysledek = (2*vysledek+(unsigned char)s[i])%HASHSIZE;
49:   return vysledek;
50: }
```

hash.c

```
void inithash(): 2   int hashfunc(): 2-3
```

Obrázek 2: Ukázka z Mirkovy dokumentace, modul hash

```

44: }
45:
46: if(*pom){
47:     cas += (*pom)->zbyva; // cas jsem prestrelil
48:     (*pom)->zbyva -= cas; // upravim zbyva v nasledujicim timeru
49: }
50:
51: tim->zbyva = cas;
52: tim->next = *pom;
53: *pom = tim;
54: }
55:
56: static void vyradtimer (kq_timer *tim)
57: /*******/
58: {
59:     kq_timer **pom;
60:
61:     if(tim->pause) pom = &pausetimers;
62:     else pom = &prvtimer;
63:
64:     for(; *pom != tim; pom = &(*pom)->next)
65:         if(!*pom){
66:             warning("vyradtimer(): timer nenalezen");
67:             return;
68:         }
69:
70:     *pom = tim->next;
71:     if(tim->next || !tim->pause) tim->next->zbyva += tim->zbyva;
72: }

```

Funkce `pausetimer` práci se seznamy řeší samostatně. Při vyřazení totiž musí spočítat zbývající čas a zařazení k zapausovaným timerům je příliš snadné na to, aby na to byla volána funkce.

4 Rejstřík

BINDTEXT: 1, 1, 3	kq_timer* newtimer: 7, 4, 7
BINDWIN: 1, 1, 3	void pausetimer(): 5, 4-6, 10
void deleteltimers(): 6, 4, 6-7	void unpausetimer(): 5, 4, 6
void deletetimer(): 7, 4	void vypistimer(): 8, 4, 8
void dotimers(): 4, 4-5	void vypistimery(): 8, 4
void hash_add(): 3, 3	void vyradtimer(): 9, 5-6, 10
kq_pin* hash_find: 3, 3	WORD: 1, 1, 3
int hashfunc(): 2, 2-3	WORDCAT: 1, 1, 3
HASHSIZE: 1, 2	WORDPART: 1, 1
hashtable: 1, 2-3	void zaradtimer(): 9, 5-7, 9
void inithash(): 2, 2	int zjistitimer(): 8, 4-5, 8-9
kq_timer: 4, 4-10	void zmentimer(): 5, 4-5

Obrázek 3: Ukázka z Mirkovy dokumentace, rejstřík

Obsah

1	Úvod	1
2	Dokumentování zdrojových kódů v jazyce C	1
2.1	Příklad dokumentace modulu	2
	<code>dvojice ... 3, uzasna_funkce() ... 3</code>	
2.2	Vymezovací text	4
2.3	Ignorování nebo vypisování vymezujících řádků	4
	<code>\skipping ... 4</code>	
2.4	Vkládání zdrojových textů „po částech“	4
2.5	Načtení prototypu	5
2.6	Lokální deklarace, jmenné prostory	6
2.7	Generování rejstříku a obsahu	6
2.8	Módy DocBy.T _E Xu	6
3	Reference příkazů systému DocBy.T _E X	7
3.1	Kapitoly, sekce, moduly	7
	<code>\module ... 7, \modulename ... 7</code>	
3.2	Vkládání zdrojových textů	7
	<code>\ifirst ... 7, \ins ... 7, \inext ... 7</code>	
3.3	Automatické generování aktivních odkazů a vyznačování komentářů	7
	<code>\commentreset ... 7, \setcomment ... 7</code>	
3.4	Deklarování cílů odkazů	8
	<code>\d ... 8, \n ... 8, \ifig ... 8, \figdir ... 8, \dotoc ... 8, \doindex ... 8,</code> <code>\titmodule ... 8, \tittoc ... 8, \titindex ... 8</code>	
4	Jmenné prostory	8
	<code>\namespace ... 8, \ds ... 9</code>	
5	Aplikační dokumentace	9
	<code>\begapi ... 9, \endapi ... 9</code>	
6	Rejstřík	9

1 Úvod

DocBy.T_EX umožňuje jednoduše dokumentovat pomocí T_EXu zdrojové kódy programu v jazyce C. Je možné jej použít i k dokumentování něčeho jiného.

Na rozdíl od Knuthova literárního programování tento nástroj nepoužívá žádné preprocesory nebo filtry pro oddělení informace pro člověka a pro počítač. Vycházím z toho, že programátor je zvyklý psát tyto informace odděleně a chce mít věci pod vlastní kontrolou. Rovněž mnozí programátoři uvítají, že mohou psát dokumentaci dodatečně, a přitom skoro nezasahovat do už napsaného (a možná odladěného) zdrojového kódu. Doba, kdy Knuth navrhoval literární programování, pokročila a tvůrce dokumentace dnes může mít zároveň ve více oknech otevřeno více textů. Některé jsou určeny pro člověka a jiné pro počítač. Nevidím tedy tak hlasitou potřebu tyto informace slučovat do jednoho souboru, jako tomu bylo kdysi.

2 Dokumentování zdrojových kódů v jazyce C

Předpokládá se, že zdrojové kódy jsou členěny na moduly. Každý modul je myšlenkově samostatná záležitost. Alespoň pro programátora. Každý modul má své jméno (například `cosi`) a je napsán v souborech `cosi.h` a `cosi.c`, případně v dalších. Tyto soubory se kompilují, aby vznikl `cosi.o` a v závěru kompilace se linkují všechny kompilované moduly do výsledného programu.

dulům. V těchto souborech se autor dokumentace soustředí na problém řešený v daném modulu (něco jako sekce v Knuthově WEBu) a postupně vkládá do dokumentace vymezené úseky kódu ze souborů *.c a *.h pomocí příkazu „\ins“. Dokumentovaná slova vymezuje příkazem „\dg“ (dokumentace globálního slova). Vztah mezi jednotlivými soubory při načítání kompilátorem a při načítání T_EXem je naznačen na obrázku 5.

Obrázek 5: Vztahy mezi zdrojovými soubory

Obsah souboru `lup.tex` může vypadat třeba takto:

```
\input docby.tex
```

```
\title Program lup - dokumentace ke zdrojovým textům
```

```
\author Progr a Mátor
```

```
\dotoc % tady bude obsah
```

```
\sec Členění zdrojových textů
```

Zdrojové texty programu "lup" jsou rozděleny do tří modulů. V "base.c" jsou definovány pomocné funkce a v "base.h" jsou jejich prototypy. Podobně ve "win.c" jsou funkce pro okenní záležitosti a "win.h" obsahuje jejich prototypy. Konečně "main.c" obsahuje hlavní funkci programu.

```

\module base
\module win
\module main
\doindex % v tomto místě bude sestaven rejstřík
\end

```

V tomto příkladě jsme se rozhodli čtenáře dokumentace seznamovat s programem „zdola nahoru“, tedy od elementárních funkcí až k hotovému programu. Někdo možná preferuje cestu „shora dolů“ a může mít v dokumentaci na konci souboru `lup.tex` třeba napsáno:

```

\module main
\module win
\module base
\doindex
\end

```

Oba přístupy jsou možné, protože dokumentace je automaticky provázána hyperlinky. Čtenář se kdykoli může podívat na dokumentaci té funkce, jejíž použití zrovna čte, a obráceně může projít výskyty veškerého použití funkce, když čte její dokumentaci.

Podívejme se nyní podrobněji například do souboru `base.d`, kde je soustředěna dokumentace modulu `base`. Část tohoto souboru může vypadat třeba takto:

```

Struktura \dg dvojice se používá jako návratová hodnota funkce
"uzasna_funkce" a sdružuje dvě hodnoty typu "float".
\ins c dvojice

```

```

Funkce \dg [struct dvojice]uzasna_funkce() si vezme jeden parametr
"p" a vrátí ve struktuře "dvojice" dvojnásobek a trojnásobek
tohoto parametru.
\ins c uzasna_funkce

```

Příkazem „`\ins c dvojice`“ jsme dali najevo, že potřebujeme ze souboru se stejným jménem jako je aktuální název modulu (tj. `base`) a s příponou `.c` přečíst část kódu, který obsahuje deklaraci zmíněné struktury. Pro vymezení částí z načítaného souboru je potřeba do zdrojového kódu vložit vymezuující poznámky tvaru `//: slovo`. Příklad části kódu ze souboru `base.c` vše osvětlí:

```

#include <stdio.h>

//: dvojice

struct dvojice {
    float x, y;

```

```

};

//: uzasna_funkce

struct dvojice uzasna_funkce (float p)
{
    struct dvojice navrat;
    navrat.x = 2*p; // tady nasobim p dvema
    navrat.y = 3*p; // tady nasobim p tremi
    return navrat;
}

```

Příkaz „\ins c slovo“ tedy načte část kódu od vymezující poznámky tvaru `//: slovo` po výskyt poznámky tvaru `//:` nebo až po konec souboru. Do zdrojového kódu je tedy potřeba vložit vymezující poznámky.

Na pořadí úseků, které zahrnujeme ze zdrojového textu do dokumentace, nezáleží. Klidně jsme mohli dokumentaci začít od povídání o úžasné funkci (včetně vložení jejího kódu) a potom ještě dopsat, co to je ta struktura `dvojice` a následně vložit deklaraci této struktury.

Na obrázku 6 můžete vidět zhruba výsledek zpracování našeho příkladu DocBy.T_EXem (první stránka). Níže sice píšu „všimněte si“, „za povšimnutí stojí“ atd., ovšem je to potřeba brát s rezervou. Je možné, že tisk obrázku 6 bude jen černobílý a pravděpodobně dost zmenšený. Omlouvám se tedy preventivně čtenáři, pokud ho nutím, aby si všiml, že něco zmodralo nebo zezelenalo, když asi obrázek 6 mu tuto informaci jednoznačně neposkytne.

Všimněme si, že T_EX zapsal čísla řádků přesně podle toho, jak jsou ve zdrojovém kódu. Tj. počítal i přeskakovaný řádek `#include <stdio.h>` i přeskakované prázdné a vymezující řádky. Komentářový řádek s vymezením `//: slovo` není do dokumentace zahrnut a pokud následuje za řádkem s vymezením prázdný řádek, ani ten není do dokumentace zahrnut.

Za povšimnutí stojí použití příkazu „\dg“. Za ním následuje slovo (separované mezerou), které dokumentujeme. Toto slovo se v dokumentaci výrazně označí (v PDF verzi červenou barvou navíc v barevném rámečku) a jakýkoli jiný výskyt takového slova ve zdrojovém textu nebo mezi uvozovkami „...“ bude automaticky označen modrou barvou a bude klikací. Kliknutí na modrý výskyt slova kdekoli v dokumentaci vrátí čtenáře na červený výskyt, kde je slovo dokumentováno.

Všechny Céčkové komentáře ve vloženém zdrojovém kódu automaticky zezelenaly.

Palcové uvozovky „...“ vymezují kusy kódu uvnitř odstavce. Text takto uvozený je psán strojopisem a pokud se v něm vyskytují dokumentovaná slova, tato slova automaticky modrají. Text mezi těmito uvozovkami je navíc přepisován ve „verbatim“ módu T_EXu, tj. žádné znaky nemají speciální vlastnosti (s výjimkou

Program lup – dokumentace ke zdrojovým textům

Progr a Mátor

Obsah

1	Členění zdrojových textů	1
2	Modul base	1
	dvojice ...1, uzasna_funkce() ...1	
3	Modul win	1
	uzasne_wokno(), ... 1	
4	Modul main	1
	main() ...1	
5	Rejstřík.....	2

1 Členění zdrojových textů

Zdrojové texty programu lup jsou rozděleny do tří modulů. V `base.c` jsou definovány pomocné funkce a v `base.h` jsou jejich prototypy. Podobně ve `win.c` jsou funkce pro okenní záležitosti a v `win.h` obsahuje jejich prototypy. Konečně `main.c` obsahuje hlavní funkci programu.

2 Modul base

Struktura `dvojice` se používá jako návratová hodnota funkce `uzasna_funkce` a sdružuje dvě hodnoty typu `float`.

```
5: struct dvojice {  
6:     float x, y;  
7: };
```

`base.c`

Funkce `uzasna_funkce` si vezme jeden parametr `p` a vrátí ve struktuře `dvojice` dvojnásobek a trojnásobek tohoto parametru.

```
11: struct dvojice uzasna_funkce (float p)  
12: {  
13:     struct dvojice navrat;  
14:     navrat.x = 2*p; // tady nasobim p dvema  
15:     navrat.y = 3*p; // tady nasobim p tremi  
16:     return navrat;  
17: }
```

`base.c`

3 Modul win

Hlavním obsahem tohoto modulu je implementace funkce `uzasne_wokno` která založí okno programu lup. Návratová hodnota této funkce obsahuje inicializovanou strukturu `win` tohoto okna.

```
3: win uzasne_wokno ()  
4: {  
5:     win navrat;  
6:     ...  
7:     return navrat;  
8: }
```

`win.c`

4 Modul main

Funkce `main` našeho programu nejprve přečte parametry příkazové řádky, pak v rychlosti spočítá výsledek, přičemž využije funkci `uzasna_funkce`. Nakonec pomocí funkce `uzasne_wokno` zobrazí uživatelsky přívětivým způsobem výsledek.

```
dvojice: 1      struct dvojice uzasna_funkce(): 1-2      win uzasne_wokno(),: 1-2      int main(): 1-2
```

Obrázek 6: Ukázka ukázkového výstupu

koncové palcové uvozovky).

Na stránce, kde je slovo dokumentováno (pomocí „\dg“), je v poznámkách pod čarou slovo znovu zmíněno a vedle této zmínky je seznam všech stránek, na kterých se kdekoli v textu vyskytuje použití tohoto slova. Dále jsou všechna dokumentovaná slova zahrnuta do závěrečného abecedního rejstříku, který odkazuje jednak na stránku, kde je slovo dokumentováno, i na stránky se všemi výskyty slova.

Pozorný čtenář si jistě všiml, že v poznámce pod čarou a v rejstříku je `uzasna_funkce` zmíněna včetně jejího návratového typu a navíc je slovo ukončeno dvojicí závorek `()` a tím je naznačeno, že se jedná o funkci. Je to díky zápisu „\dg [struct dvojice]uzasna_funkce()“ v místě dokumentace funkce. Nepovinný parametr s návratovou hodnotou funkce se přepíše před jméno funkce pod čáru i do rejstříku, ale rejstřík při abecedním řazení tento parametr ignoruje.

3. Další možnosti

DocBy.T_EX umožňuje vyznačit vkládané úseky kódu dalšími způsoby: je možné kontrolovat, zda se vloží nebo nevloží vymezující řádky, nebo jen vymezující řádek na jedné straně, je možné navazovat vkládáním od místa, kde vkládání naposledy skončilo, je možné vložit prototyp funkce, je možné vkládat od libovolně vymezeného místa ve zdrojovém kódu do dalšího vymezeného místa (není tedy nutné vymezovat text jen výše uvedenými poznámkami), je možné vložit jediný řádek s prvním výskytem slova nebo s dalším výskytem slova od místa posledního vložení.

Je možné nastavit jiné vzory, podle kterých zelenají ve vložených souborech komentáře. Implicitně DocBy.T_EX považuje za komentář úsek od `//` do konce řádku a úsek mezi `/*...*/`.

V DocBy.T_EXu může autor dokumentace založit sekce a subsekcce, přitom implicitně příkaz „\module“ založí novou sekci. Je možné odkazovat pomocí lejblíků na sekce, subsekcce. Je možné vkládat obrázky a odkazovat na ně.

DocBy.T_EX implementuje jmenné prostory, takže je možné dokumentovat slovo lokálně v rámci úseku dokumentace, která vymezuje určitý jmenný prostor. Nastavení jmenných prostorů je poměrně flexibilní a umožňuje na lokálně dokumentované slovo odkazovat globálně při dlouhém výpisu slova včetně specifikace jmenného prostoru.

DocBy.T_EX rozlišuje mezi PDF a DVI módem, dále mezi enc a non-enc módem a konečně mezi csplain a plain módem.

Pro zpracování dokumentace je ideální použít pdfT_EX s aktivovaným výstupem do PDF (tj. PDF mód) a s encT_EXem (tj. enc mód) a s formátem csplain nebo plain (tj. csplain nebo plain mód). V takovém případě fungují barvy, odkazy i automatické tvoření aktivních odkazů ve vkládaném zdrojovém textu.

Není-li aktivován PDF výstup, DocBy.T_EX o tom napíše varování na terminál a přejde do DVI módu. V tomto módu nefungují barvy a odkazy nejsou klikací.

Není-li aktivován encT_EX, DocBy.T_EX přejde do non-enc módu a napíše o tom varování na terminál. V takovém případě dokumentovaná slova ve vkládaném zdrojovém textu automaticky nemodrají a nestávají se klikatelnými odkazy. Ani se jejich výskyty neuvádějí na správných stránkách v rejstříku a v poznámkách pod čarou. Taktéž komentáře automaticky nezelenají.

Při csplain módu generuje DocBy.T_EX některé názvy česky. Jedná se o název kapitoly s obsahem, rejstříkem a modulem. V plain módu jsou tyto názvy anglicky. Pracujete-li s jiným jazykem, můžete předefinovat odpovídající makra.

Generování rejstříku i obsahu probíhá v DocBy.T_EXu zcela automaticky. Pro vytvoření rejstříku není nutné používat externí program (DocBy.T_EX si slova abecedně zatřídí sám). Stačí tedy vložit na požadovaná místa příkazy „\dotoc“ a „\doindex“. Rejstřík ani obsah nejsou správně vygenerovány po prvním průchodu T_EXu. Je potřeba T_EXovat dvakrát nebo třikrát. Pro generování obsahu i rejstříku si DocBy.T_EX zakládá pomocný soubor s příponou .ref, v našem příkladě tedy lup.ref.

Summary: DocBy.T_EX – documenting source codes by T_EX

DocBy.T_EX is a T_EX macro software product which gives the possibility to document simply your source codes written in various programming languages, for example written in C. You can include parts of your source code into your documentation. All occurrences of documented words in your included source code are automatically made as active links if encT_EX and pdfT_EX is in progress. To make PDF output, you need no more than pdfT_EX with encT_EX. The table of contents and the index are also created automatically. The sorting of the words in the index is implemented at T_EX macro level.

Značkovací styl pro rychlou sazbu bibliografických citací

TOMÁŠ HÁLA

Práce předkládá stručný přehled možností sazby bibliografických citací. Běžně dostupné metody však neřeší problémy, které nastávají při zpracování citací v textech jiných autorů, například v rozsáhlejších sbornících. Článek představuje styl **bib.sty** umožňující jednoduché a přehledné strukturní značkování běžně používaných druhů bibliografických citací a jejich prvků. Součástí je komentovaný soupis použitých makropříkazů.

1. Úvod

Pojem **bibliografická citace** není potřeba příliš představovat, neboť se jedná o denní potřebu většiny vědeckopedagogických pracovníků. Úvodem si však dovolíme krátký přehled zdrojů, jež souvisejí s bibliografickými citacemi.

2. Přehled zdrojů

2.1. Normy

ISO 690 (1987), oficiální mezinárodní norma platná v současnosti, je převzata do soustavy českých norem pod označením **ČSN ISO 690** (1996) s doplněním tzv. národní přílohy. Určuje šest základních typů publikací, na něž (a na žádné jiné) smí být aplikována. Bibliografické citace elektronických dokumentů popisuje rozšíření **ISO 690-2** (1997), převzaté jako **ČSN ISO 690-2** (2000).

ČSN 01 0197 (1985) je starou, dnes již neplatnou normou pro citování. Obsahuje však podrobný popis sedmnácti druhů dokumentů, čímž téměř plně pokrývá celou škálu tištěných dokumentů, a je proto dodnes užitečným vodítkem pro tvorbu citací ostatních, v normě ČSN ISO 690 nezmíněných druhů.

Základní rozdíl mezi těmito dvěma českými normami z hlediska sazby spočívá v tom, že tzv. „stará norma“ určovala nejen obsah a strukturu citace, ale také typografické konvence. Současná norma výslovně uvádí, že interpunkce, řezy a další typografické záležitosti nestanovuje závazně.

2.2. Druhy publikací

Druhy publikací, které popisují zmíněné normy, jsou uspořádány v tabulce 1.

Tabulka 1: Hlavní druhy publikací

	ČSN ISO 690 (1996, 2000)	ČSN 01 0197 (1985)
A	monografická publikace	jednosvazkové dílo vícesvazkové dílo jako celek
B	seriálová publikace	periodikum jako celek ročník periodika
C	část monografické publikace	dílo na pokračování (nepravé periodikum) jako celek jednotlivý svazek vícesvazkového díla část díla na pokračování (nepravé periodikum)
D	příspěvek v monografické publikaci	stať (příspěvek, článek) ve sborníku
E	článek v seriálové publikaci	článek v časopisu
F	patentová dokumentace	patentový spis

Pozn. Písmeno v prvním sloupci slouží k pracovnímu označení druhu publikace a je následně užito v názvech makropříkazů.

Kromě toho norma starší uvádí dalších sedm položek, které nemají svůj obraz v normě nové (tabulka 2).

2.3. Nástroje pro práci s citacemi

Základními prostředky v systému $\text{\LaTeX}$ je prostředí `thebibliography`, které pracuje podobně jako výčtová prostředí. Uvnitř je definován makropříkaz `\bibitem`, uvozující položku soupisu citací. Citace jsou číslovány, případně pojmenovány a tato označení lze použít v textu (`\cite`).

Repertoár možností sazby citací rozšiřuje celá řada nadstaveb (stylů), jejichž podrobný popis uvádějí Mittelbach a Goossens (2004). Za zmínku stojí například `natbib` obsahující několik variant příkazu `\bibitem`. Odkazy jsou rozděleny na `\citet` – jméno autora v pozici větného členu, rok v závorce – a na `\citep` – autor i rok v závorce. Styl `bibentry` uvádí úplné citace uvnitř textu v místě odkazu, styl `jurabib` se zabývá obecně zkrácenými citacemi, dále citacemi právnických dokumentů, umožňuje řešit odkazy v případě různých rodů a čísel a umožňuje relativně snadnou úpravu vzhledu citace (fonty, sloupce). Další styly dokáží rozdělit citace podle kapitol (`bibunits`) či témat (`bibtopic`).

2.4. Nástroje pro správu báze citací

Pro komplexnější zpracování citací lze v prostředí systému $\text{\LaTeX}$ použít `BibTeX`. Jedná se o nástroj vyžadující databázi v předepsaném formátu, založeném na uspořádaných dvojicích typu identifikátor–hodnota.

Tabulka 2: Další druhy publikací

	ČSN ISO 690	ČSN 01 0197
T	--	habilitační, disertační, diplomové a podobné práce
P	--	recenze bez zvláštního názvu záznam v referátovém časopisu
N		technická norma firemní literatura výzkumné, vývojové a podobné zprávy cestovní zprávy ostatní druhy dokumentů

Skutečnost, že se za posledních více než 10 let tento systém nijak nerozvíjí, vedla další autory k pokusům o nová řešení – bibulus, $\text{BIB}_{\text{TeX}}++$, $\text{MIBIB}_{\text{TeX}}$.

Program **bibulus** (Widmann, 2003, 2008) je řešení postavené na jazyce XML. Data mohou být vytvářena buď ručně, v libovolném textovém editoru, nebo lze použít jakýkoliv editor formátu XML. Kromě toho existuje program pro převod databáze ve formátu pro BIB_{TeX} do XML. Výhodou je obecnost – data lze celkem snadno převádět i do HTML či jiných podob, Vlastní operace jsou řešeny v jazyce Perl (modul Biblus.pm). Systém však není zcela připraven, autor sám nedoporučuje jeho použití: „... but for the time being Bibulus is unfortunately really only for developers.“

MIBibTeX (Hufflen, 2003, 2004, 2005) jde směrem k vícejazyčné podpoře (Ml = multilingual), jejímž cílem je vytvářet bibliografické soupisy odpovídající národním zvyklostem (ustálené zkratky, datové údaje, uvozovky aj.). Řešení opět využívá jazyka XML, s využitím vlastního DTD. Systém se nachází ve stádiu vyšší připravenosti, je kompatibilní se systémem BIB_{TeX} , na který navazuje drobnými rozšířeními.

$\text{BibTeX}++$ je napsán v jazyce Java kvůli přenositelnosti a objektově orientovaným vlastnostem. Namísto BST používá formát DSN. Jedná se opět o parametizovaný překladač citací s možností výstupu do systému $\text{L}_{\text{AT}}_{\text{E}}_{\text{X}}$ i do dalších formátů (OpenOffice, DocBook).

Zajímavým řešením značkování bibliografických citací je Wagnerova (2006) metoda použitá ve stylu **cnbwp.sty** určeným pro publikace České národní banky. Využití stylů **natbib.sty** a **keyval.sty** umožňuje zápis podobný systému BIB_{TeX} , definice jsou však vloženy přímo do makropříkazů systému $\text{L}_{\text{AT}}_{\text{E}}_{\text{X}}$. Alternativní zpracování systémem BIB_{TeX} a připraveným stylem **abbrvcnb.bst** data narozdíl od obvyklého použití neformátuje na výstup, ale jen vypíše pomocí vlastních maker.

3. Styl pro rychlé značkování bibliografických citací

Z hlediska vedení osobní či podnikové agendy se jedná nesporně o velmi užitečné nástroje. Pro běžnou práci sazeče jsou však tato řešení příliš silná – nelze očekávat, že pro jednorázovou sazbu publikace typu sborník příspěvků bude sazeč nejprve vytvářet databázi publikací, následně je použije v seznamu citací a případně ještě v textu bude nahrazovat stávající odkazy na citace jinými odkazy.

Z tohoto důvodu vznikla potřeba nástroje, který:

- se oprostí od potřeby tvorby databáze;
- bude mít jednoduché rozhraní pro uživatele, tj. pro autora či pro sazeče-neprogramátora;
- upřednostní strukturní značkování přímo v jazyce systému $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$;
- lze rozšiřovat či modifikovat;
- vytvoří tvar citace nevyžadující další korektury, především v otázce úplnosti a správného pořadí užitých prvků.

3.1. Popis stylu `bib.sty`

Historie stylu sahá do roku 1996, kdy poprvé vyvstala potřeba zpracovat hromadně větší množství bibliografických citací, avšak bez tvorby specializované databáze. Veřejně byl styl `bib.sty` avizován již dříve (Hála, 2001), kde čtenář nalezne i přehled a příklady základních druhů bibliografických citací.

Aktuální verze stylu nese označení 2.30 (Hála, 2008) a je nezávislá na vnějším prostředí, starší verze se odvolávaly na určité prvky obecnějšího stylu pro sazbu sborníků. Od verze 2.00 je již běžně používán uživateli, kteří připravují sazbu sborníků (např. na ústavu informatiky PEF MZLU v Brně, v nakladatelství KONVOJ).

3.1.1. Připojení stylu a vytvoření prostředí

Styl se připojuje v preambuli příkazem `\usepackage{bib}`, je však možné jej vložit i příkazem `\input{bib.sty}`. V místě soupisu bibliografických citací vytvoříme prostředí dvojicí příkazů `\bbib` a `\ebib`.

3.1.2. Seznam makropříkazů typů publikací

Ve shodě s pracovním označením jednotlivých druhů publikací (viz tabulky) byly definovány makropříkazy `\publA`, `\publB`, `\publC`, `\publD`, `\publE`, `\publF`, `\publT` a `\publP`. Jejich jediným parametrem je posloupnost makropříkazů zvolených prvků dle následujícího textu. Dále existuje makropříkaz `\publN`, který je odvozen z `\publA` a slouží pro citaci norem.

Pro vlastní práci s těmito makropříkazy je potřeba připomenout, že citace každého druhu publikace obsahuje určitou sadu prvků povinných a může obsahovat

vat některé prvky volitelné. Všechny povinné prvky jsou kontrolovány a uživatel obdrží chybové hlášení v případě nepřítomnosti kteréhokoliv z nich. Nepatří-li použitý prvek mezi prvky povinné ani volitelné, ignoruje se bez dalšího varování.

3.1.3. Seznam makropříkazů prvků

Seznam je rozčleněn tematicky. Až na výjimky, výslovně zmíněné v textu, mají všechny makropříkazy jeden parametr, kterým je příslušná hodnota daného prvku.

Jména autorů – `\autor` pro autora citovaného díla. V případě sborníků či dalších děl, složených z více částí různých autorů, se pro autora publikace jako celku použije makropříkaz `\autordok`. Tyto makropříkazy se použijí obvykle jen pro jednoho autora, pro více autorů se příkaz použije opakovaně. Podřízená odpovědnost (autoři ilustrací, fotografií, překladatel apod.) se uvede do makropříkazu `\podpov`.

Název díla – `\nazev` pro citované dílo, `\nazevdok` pro název souborného díla. Analogicky se použije `\podnazev` a `\podnazevdok` v případě, že název díla je složen z více částí.

Údaje o vydávání a vydavateli – `\rok` rok vydání, `\vyd` číslo vydání u monografických publikací, `\rozsah` v počtech stran nebo jiných jednotkách u monografických publikací, `\cast` pro označení části vícesvazkového díla, ať již monografické či seriálové publikace, `\umist` umístění části (článku, příspěvku, kapitoly) v hlavním díle, udávají se dva parametry hodnoty „od strany“ a „do strany“, přičemž se kontroluje, zda první číslo je menší než druhé. Pro strany uvedené římskými číslicemi (hodnota se však zadává arabskými číslicemi) je k dispozici dvojice `\rumist` a `\Rumist`, pro ostatní případy lze použít textově orientovaný makropříkaz `\tumist`, v němž se nekontrolují hodnoty. Místo vydání a název vydavatele jsou dva parametry makropříkazu `\nakl`.

Standardní čísla – `\isbn`, `\issn`, `\ismn` se uvádějí bez písmenné zkratky a včetně spojovníků.

Ostatní položky – edice se zaznamenává do `\edice`, t. č. se tento údaj o edici ve výstupním tvaru, a dva druhy poznámek `\pozn` a `\private`.

Dokumenty on-line – `\online` obsahuje datum citace, tvar není předepsán, ale doporučuje se použít mezinárodně uznávané pořadí „RRRR-MM-DD“, `\www` se použije pro zápis URI.

Závěrečné práce – `\bakpr`, `\dippr`, `\dispr`, `\habpr` – bezparametrické příkazy pro jednotlivé druhy závěrečných prací.

3.2. Doplnkové makropříkazy

Doplnkové makropříkazy slouží převážně k rychlejšímu zápisu určitých jevů: v tisku `\inpress`, s. (stran) `\stran`, listů `\Listu`, Roč. `\Rocnik`, red. (redaktor) `\red`, ed. (editor) `\ed`, eds. (editoři, v anglických textech) `\eds`, a kol. `\akol`.

Pro potlačení tečky za názvy děl, které končí otazníkem, příp. jiným interpunkčním znaménkem, se do parametru příkazů `\nazev` či `\nazevdok` přidá `\netecka`.

Další skupinou doplňkových makropříkazů jsou definice názvů a standardních čísel vybraných dokumentů, které autor stylu potřeboval velmi často.

3.3. Vyřešené problémy

Rozšíření pro dokumenty on-line se ve srovnání s rozsahem normy ukázalo jako velmi jednoduché. Původní definice byly rozšířeny pouhou dvojicí makropříkazů.

Více než tři autoři. V českých zemích a na Slovensku bývá zvykem delší seznam autorů zkracovat na jednoho až tři autory, následované zkratkou „a kol.“. Systém hlídá překročení počtu tří použití makropříkazu `\autor` a ohlásí chybu. Pokud však i přesto potřebujeme uvést všechny autory, což je obvyklé zejména v anglosaské literatuře, můžeme problém obejít tak, že do posledního, tj. třetího makropříkazu uvedeme i zbývající autory.

Dělení URI činilo při sazbě značné potíže. Proto zlom URI byl svěřen stylu `url.sty`, a lze předeslat, že makropříkaz `\www` je nadstavbou makropříkazu `url`.

Neúplné citace se staly skutečným problémem. Při sazbě souborných děl více autorů není žádná síla, která by donutila autory text doplnit. Dokonce i editoři jednotlivých svazků ignorují absenci základních pravidel. Pro tyto případy byl – ač se sebezapřením autora – připraven malý doplněk, pracovně nazvaný `bibprd.sty`, který potlačuje v podstatě všechny kontroly a upravuje vzhled citací s ohledem na chybějící údaje.¹

3.4. Co je uvnitř stylu

Základní konstrukci pro každý druh bibliografické citace tvoří trojice makropříkazů:

inicializační makropříkaz `\biblinit` je společný a jeho chování lze ovlivnit parametrem označujícím druh citace;

kontrolní makropříkazy `\bibltestX`, kde X je písmeno označující druh citace, provádí stanovené kontroly a zobrazuje chybová hlášení;

zobrazovací makropříkaz `\bibldoneX` uspořádá zadané údaje do určeného tvaru, doplní příslušná interpunkční znaménka a sestaví tělo citace.

Příklad kontrolního a zobrazovacího makropříkazu pro citaci monografické publikace uvádí následující výpis.

```
\def\publA#1{\biblinit{A}#1\bibltestA\bibldoneA}
```

¹Autor však na tomto místě výslovně prohlašuje, že tento doplněk používá jen na nátlak zákazníků.

```

\def\biblttestA{\biblif\@autortest\@nazevtest\@vydtest%
\@roktest\@nakltest\@isbntest}

\def\bibldoneA{\def\@tisk{\fontcitace{\fontautor\biblAutor}%
{\fontnazev\biblNazev} \biblVyd
\biblNakl{} \biblRok\biblCit
\if*\Rozsah.\else, \Rozsah\fi} %
\if*\Www\else{} \Www.{ } \fi%
\Isbn%
}\@bibakce{\@tisk}
}

```

Jako ukázkou použijeme jednu z citací z tohoto článku, výsledek pak lze prohlédnout v seznamu literatury.

```

\publA{
\autor{Mittelbach, F., Goossens, M.\akol}
\nazev{The \LaTeX{} Companion}
\isbn{0-201-36299-6}
\rok{2004}
\vyd{2}
\rozsah{1117\stran}
\nakl{Boston}{Addison-Wesley}
}

```

4. Závěr a pohled do budoucnosti

Styl, který zde byl představen, splňuje základní výchozí požadavky: nevyužívá žádných databázových ani jiných externích nástrojů, pro značkování jsou použity makropříkazy systému L^AT_EX. Je pravděpodobné, avšak nevyzkoušené, že styl je kompatibilní i s dalšími formáty. Repertoár lze snadno rozšiřovat i prostou analogií, od potřeby snadné modifikovatelnosti bylo prozatím ustoupeno. Několikaleté užívání tohoto stylu umožnilo otestovat jej na řadě různých zdrojů a lze konstatovat, že pro běžné použití je styl zcela použitelný.

Další kategorie publikací, které stojí za dopracování, jsou zákony a další podobné předpisy, vybraná firemní literatura, např. účetní výkazy, a z elektronických dokumentů nejsou dořešeny určité záležitosti týkající se citací zdrojů na nosičích. Zmínit lze také otázku vícejazyčného prostředí. Styl je konstruován obecně, s možností sazby v dalších jazycích se tudíž přímo počítá. Tato část však nebyla rovněž z časových důvodů dotažena. Navržené úpravy jsou však v plánu.

Summary: Markup Style for Fast Typesetting of Bibliographic References

The paper deals with the typesetting of bibliographic references. The introduction covers some important methods of styles and the systems for processing and typesetting bibliographic references. Basic problems of the proceedings in typesetting are dealt with. No one method alone is suitable for the typesetting of proceedings. A basic style and some extensions focus on designing cross references. Sophisticated database methods use up a lot of time while the database is being prepared and it can only be used once.

In conclusion a new style for faster markup and typesetting is created.

The input conditions are: (a) no database usage, (b) a simple interface for authors and/or typesetters, (c) complete markup in $\text{\LaTeX}$ macros, (d) extendable and modifiable when necessary, (e) the result does not need detailed proof.

The style `bib.sty` contains macros for the most frequently used types of bibliographic references and for elements of references. Some additional macros are described and electronic documents are also included.

Reference

- ČSN 01 0197 : *Bibliografické citace*. 5. vyd., Praha : Úřad pro normalizaci a měření, 1985, 16 s.
- ISO 690 : *Bibliographical references. Essential and supplementary elements*. 2. vyd., 1987.
- ISO 690-2 : *Bibliographical references. Electronic documents or parts thereof*. 1. vyd., 1997.
- ČSN ISO 690 : *Bibliografické citace. Obsah, forma, struktura*. 1. vyd., Praha : Český normalizační institut, 1996, 32 s.
- ČSN ISO 690-2 : *Bibliografické citace. Elektronické dokumenty nebo jejich části*. 1. vyd., Praha : Český normalizační institut, 2000, 24 s.
- DAGNAT, F., KERYELL, R., SASTRE, L. B., DONIN DE ROSIÉRE, E., TORNERI, N. European Bibliography Styles and MIBIB $\text{\TeX}$. *TUGboat.*, 2003, Vol. 24, N. 3, s. 472–488.
- HÁLA, T. Bibliografické citace včera, dnes a zítra. In KASPRZAK, J., SOJKA, P. (ed.) *SLT 2001*. 1. vyd., Brno : Konvoj, 2001, s. 85–98. (ISBN 80-7302-009-2.)
- HÁLA, T. bib.sty, v. 2.30. Značkovací styl pro rychlou sazbu bibliografických citací [on-line]. 2008 [cit. 2008-08-31]. Dostupné na: <http://www.konvoj.cz/style/biblio/2.30/bib.sty>.
- HUFFLEN, J.-M. European Bibliography Styles and MIBIB $\text{\TeX}$. *TUGboat.*, 2003, Vol. 24, N. 3, s. 489–498. (ISSN 0896-3207.)
- HUFFLEN, J.-M. A Tour around MIBIB $\text{\TeX}$ and its implementation(s). *Biuletyn Polskiej Grupy Użytkowników Systemu $\text{\TeX}$* , 2004, Zes. 20, s. 21–28. (ISSN 1230-5650.)
- HUFFLEN, J.-M. Bibliography Styles Easier with MIBIB $\text{\TeX}$. In *Euro $\text{\TeX}$ 2005*. 1. vyd., 2005, s. 105–119. Dostupné na: <http://www.dante.de/dante/events/eurotex/papers/WET02.pdf>.
- MITTELBACH, F., GOOSSENS, M. a kol. *The $\text{\LaTeX}$ Companion*. 2. vyd., Boston : Addison-Wesley, 2004, 1117 s. ISBN 0-201-36299-6.
- WAGNER, Z. [cnbwp.cls] LaTeX class for working papers of CNB, v. 1.51 [on-line]. 2006-04-02 [cit. 2008-09-20]. Dostupné na: http://www.cnb.cz/cs/vyzkum/vyzkum_publicace/download/cnbwp.zip.

WIDMANN, T. Bibulus – a Perl/XML replacement for BibT_EX. *TUGboat.*, 2003, Vol. 24, N. 3, s. 468–471. (ISSN 0896-3207.)
[WIDMANN, T.] [Bibulus] [on-line]. 2005 [cit. 2008-08-30]. Dostupné na: <http://www.nongnu.org/bibulus/>.

Tomáš Hála
thala@mendelu.cz

Příspěvek se zabývá problematikou sazby tabulek. Stručně popisuje balíky určené pro tabulky a pro úpravu jednotlivých vlastností tabulek. U vybraných balíků, které mohou být pro sazbu nejvíce přínosné, je provedena analýza jejich kompatibility. Ukázka sazby bez použití balíků a s nimi upozorňuje na rozsáhlé možnosti tabulkové sazby.

1. Úvod

Běžnou součástí textů nejrůznějšího druhu jsou tabulky. L^AT_EXna ně pamatuje s prostředím `tabular`, `tabular*`, `tabbing` a s plovoucím prostředím `table`. Pokud bychom ale chtěli tabulky typograficky dokonalé, podle pravidel [1, 2], tyto nástroje nám stačit nebudou. Pravidla pro správnou sazbu tabulek zahrnují některá specifika, jejichž cílem je dosáhnout především čitelnosti a přehlednosti údajů v tabulkách.

2. Tabulky z typografického hlediska

Tabulka se dělí na řádky, přičemž význačné postavení zaujímá hlavička, tj. první řádek tabulky. Ten tvoří nadpisky pro zbytek tabulky, tzv. nohy. Nohy se skládají ze sloupců. Na tabulku tak můžeme pohlížet jako na posloupnost řádků nebo sloupců. Díky způsobu, jakým se tabulky vytvářejí (například v T_EXu), můžeme tabulku vnímat jako jednotlivé řádky, které se dále dělí na jednotlivá pole. (Česká terminologie dává přednost výrazu *pole* nebo *oko* tabulky před doslovným překladem *buňka*.) Strukturu tabulky proto vymezuje uspořádání údajů do řádků a polí včetně možnosti spojovat pole vodorovně nebo svisle. Právě přehledné uspořádání je důvodem používání tabulkové sazby. Přehlednost by měla být podpořena správným užitím linek, vhodným množstvím světla a nastavením ostatních parametrů sazby podle typografických pravidel pro sazbu tabulek.

Linky jsou velmi výrazným prvkem v tabulce – jejich použití či nepoužití, síla linky a její obraz mohou mít výrazný vliv na čitelnost údajů. Dyntar (1955) uvádí detailní přehled druhů linek včetně pravidel pro jejich křížení. Moderní způsob sazby upouští od přehnaného používání linek, např. Felici (2003) uvádí, že vertikální linky „prakticky zmizely“. Tabulky jsou často sázené jako tzv. otevřené, bez linek, jen s vodorovnou linkou pod hlavičkou (případně i v patě tabulky)

nebo se využívá podtisku (každý druhý řádek je podbarven). Pokud se ale linky používají, měla by být dodržena typografická pravidla:

- Nejvyšší prioritu má dvojitá linka, poté jednoduché linky podle síly (tloušťky). Dvojitá linka nesmí být přerušena jednoduchou linkou.
- Nejsilnější by měla být obvodová linka a linka pod hlavičkou, vnitřní linky mají nižší důležitost a měly by být méně výrazné.
- Pro tabulky postačuje použití linky o tloušťce půl bodu až jeden bod, ostatní linky jsou již příliš silné a činí obsah tabulky nečitelným. (Některé textové procesory ovšem dovolují nastavit tloušťku linky až na devět bodů.)

Světlo v tabulce je důležitým faktorem, který má vliv na přehlednost údajů. Čitelnost vyžaduje, aby údaje nebyly v tabulce příliš nahuštěné a aby byly dostatečně vzdálené od linek (alespoň o dva body). Dodržení tohoto pravidla je poněkud problematické, např. tehdy, pokud se v tabulce vyskytují verzálky či sazba s exponenty. Řešení spočívá v rozšíření tabulkových polí či ve vkládání neviditelných „podpěr“ na problematická místa. Dále by tabulka měla být dostatečně vzdálená od okolního textu, neměla by jím být těsně sevřená. Kvůli optimálnímu množství světla se také doporučuje sázet text v tabulce menším stupněm písma (o 1–2 body).

Ostatní pravidla pro sazbu tabulek zahrnují především zarovnání údajů (opět kvůli přehlednosti), s tím, že zejména číselné sloupce by měly být zarovnané na desetinné znaménko. Ostatní pravidla pak hovoří o sazbě hlavičky, titulku (popisku) či o pravidlech sazby tabulky na šířku.

3. Základní vlastnosti tabulek

Sazba tabulek má tedy svá zvláštní pravidla, která by měla být v tabulce ošetřena. Složitá struktura tabulek, zvláště rozsáhlejších, je také díky své nepřehlednosti náročná na pozornost při sazbě. Jako reakce na řešení těchto požadavků vzniklo mnoho balíků, pomocí nichž lze vzhled tabulky zdokonalit (a mnohdy i zjednodušit sazbu).

Na tabulku můžeme nahlížet jako na celek, nebo si budeme všimnout jen některých jejích vlastností. Pro každý z těchto případů jsou k dispozici rozšiřující balíky, jejichž stručný přehled následuje. Podrobnější informace lze nalézt v [3–5].

3.1. Umístění tabulky

Umístění tabulky v rámci stránky lze řešit nejsnáze pomocí prostředí `table`, u vícestránkových tabulek ale musí nastoupit jiné řešení. Zřejmě nejpropracovanějším balíkem je `longtable`, který kromě jiného umí tabulku zarovnat a doplnit ji popiskem. Podobné možnosti nabízí balík `supertabular`, který ovšem interně používá prostředí `table` nebo `table*`. Jeho rozšířenou verzí je prostředí `xtabular` z balíku `xtab`.

Balík `ltxtable` rozšiřuje použití balíku `tabularx` i na dlouhé tabulky. Alternativou je balík `ltablex`, který kombinuje vlastnosti balíků `tabularx` a `longtable`. Rozšíření prostředí `tabular` nabízí balík `stabular` se snadno použitelnými prostředními `stabular` a `stabular*`. Balík `bigtabular` definuje prostředí `Tabular`, což je prostředí `tabular*`, které se může na konci stránky rozdělit.

Řešení je tedy více, ale nejvíce možností zřejmě nabízí balík `longtable`.

3.2. Rozměry tabulky a jejích částí

V některých případech můžeme chtít nastavit šířku tabulky. Pevnou šířku lze zařídit pomocí prostředí `tabular*` a dosažení daných rozměrů se řeší pomocí úpravy mezisloupcových mezer. Prostředí `tabularx` ze stejnojmenného balíku naopak definuje sloupec s označením `X`, jehož šířka se vypočítá automaticky podle požadované celkové šířky tabulky. Šířku lze upravit i použitím balíku `array`, který rovněž nabízí ovlivnění šířky mezisloupcových mezer.

3.3. Práce se skupinami polí

Častým požadavkem je sloučení tabulkových polí (horizontálně i vertikálně). Kromě standardního příkazu `\multicolumn` je k dispozici balík `multirow` a příkaz `\multirow`, který umožňuje vertikální spojování polí. Balík má několik neduhů, ale pro sloučení několika polí ve sloupci je velice užitečný.

3.4. Světlo v tabulce

Čitelnost údajů v tabulce závisí mj. na tom, zda je v tabulce dost prostoru, neboli světla. Zejména při použití verzálek nebo exponentů je potřeba rozměry tabulkových polí zvětšit a opět existuje několik řešení.

Příkaz `\extrarowheight` z balíku `array` způsobí zvětšení výšky řádku tabulky. Balík `bigstrut` poskytuje tzv. podpěry (struts), jimiž se dosahuje zvýšení řádků, a tedy docílení lepšího rozdělení světla v tabulce. K tomu slouží příkaz `\bigstrut`. S volitelným parametrem `t` přidává řádku výšku, s volitelným parametrem `b` přidává hloubku, příkaz bez parametru zajišťuje obojí. Obdobného efektu lze dosáhnout umístěním velkých (víceřádkových) oddělovačů do prostředí `tabular` a `array` za použití balíku `bigdelim`.

Balík `tabls` upravuje mezerování v prostředích `tabular` a `array`. Modifikuje tato prostředí tak, aby se text nedotýkal jiného textu nebo linek. Definuje nové parametry a předefinovává některá standardní makra. Balík `cellspace` zajišťuje pro pole s nadměrnou výškou alespoň minimální mezerování tak, aby se obsah nedotýkal linek. Je jednodušší alternativou k balíku `tabls`.

3.5. Ohraničení

K rozšíření možností sazby horizontálních linek v prostředích **tabular** a **array** lze použít balík **hhline**. Zavádí příkaz `\hhline`, který vytvoří obvyklou jednoduchou nebo dvojitou linku – rozdíl je však ve způsobu křížení s vertikálními linkami. Za příkazem se v povinném parametru specifikuje, zda bude linka jednoduchá, nebo dvojitá, a jakým způsobem bude křížení provedeno. $\text{\LaTeX}$ standardně vytváří souvislé linky pomocí příkazu `\hrule`, balík **hhline** linky sestavuje z mnoha malých segmentů. Tento balík je vhodným nástrojem, je-li třeba sázet dvojité linky. Jejich obraz se však může jevit jako ne zcela dobře vykreslený.

Balík **hvdashln** zavádí horizontální a vertikální čárkované linky v prostředích **tabular** a **array**. Rozšířením je balík **arydshln**. Pomocí parametrů lze nastavit délku čárky a délku mezery a čárkovaná linka se vytváří opakováním čárky s mezerou.

3.6. Zarovnání

Pro určení způsobu horizontálního zarovnání údajů ve sloupcích jsou k dispozici standardní prostředky. Balík **array** přidává ještě zarovnání na vertikální střed (parametr **m**) a dolů (**b**). Pro většinu údajů je to dostačující, sazba čísel ale vyžaduje speciální péči. Jednotlivé číselné řády mají být zarovnány pod sebou a čísla mají být vhodně umístěna vzhledem k ostatním údajům v tabulce, které nejsou číselné (například hlavička).

Balík **dcolumn** zařídí zarovnání čísel v tabulkách na desetinnou tečku (čárku) a pro tento účel definuje zvláštní sloupec s označením **D**. **dcolumn** používá pro číslice i oddělovače matematický mód a správně pracuje s čísly, které postrádají desetinnou nebo celou část, i s prázdnými záznamy.

Také balík **rccol** umožňuje zarovnání čísel. Údaje jsou zarovnány pod sebou a sázeny na střed vzhledem k ostatním údajům (například k hlavičce sloupce). Obdobně pracuje balík **warpcol**, sloupec čísel je ale umístěn na střed vzhledem k nadpisu.

Balík **numprint** tiskne čísla s oddělovačem po každých třech číslicích. Mnoho parametrů (desetinné znaménko, oddělovače tisíců atd.) si může uživatel nastavit, balík však nedovoluje zarovnávat čísla na desetinnou čárku či tečku.

3.7. Barvy

Balík **colortab** umožňuje stínování a vybarvení polí tabulky v prostředích **tabular** a **array**. Není však kompatibilní s balíkem **longtable**, proto se doporučuje používat spíše balík **colortbl**. Ten umožňuje obarvit řádky a sloupce i jednotlivá pole tabulky. Oproti balíku **colortab** pracuje odlišně; prvky, u nichž je požadováno obarvení, podkládá barevnými panely. Kromě toho balík zavádí i nové příkazy pro obarvení linek a obarvení prostoru uvnitř dvojité linky. Některé barvy jsou předdefinované, ostatní si může uživatel sám namíchat.

3.8. Ostatní

S ostatními požadavky mohou pomoci následující balíky:

Balík **ftcap** umísťuje popisky vždy nad tabulku (zatímco standardní řešení je umísťuje pod ni). Věnuje také pozornost správnému nastavení mezer okolo nadpisu. Podobnou funkci nabízí i balík **topcapt**.

Experimentální balík **blkarray** implementuje prostředí **blockarray**, které může být používáno stejně jako **tabular**, **array** nebo jejich rozšířené verze z balíku **array**. V matematickém módu se chová jako **array**, v ostatních případech jako **tabular**. Balík zavádí také vnořené prostředí **block**. To dovoluje, aby měl celý blok tabulky (například hlavička) jiné formátování. To lze zařídit například pomocí `\multicolumn`, ale **block** dovoluje specifikovat formát obvyklou syntaxí. **blkarray** je tedy zajímavým balíkem pro práci s různými druhy tabulek a matic. Vyznačuje se velkou variabilitou a dobrou kompatibilitou. Výborná je možnost formátovat specifikovaný úsek tabulky jednotně jako celek.

easytable je jednoduchý balík, který umí vytvářet tabulky se sloupci stejné šířky a/nebo řádky stejné výšky. Podporuje různé druhy linek a používá syntaxi ve stylu prostředí **tabular**. Balík zavádí prostředí **TAB**. Jeho parametry určují, zda a jak budou řádky a sloupce vyvážené (vztah výšky a šířky), jaká bude velikost tabulky a jaké je požadované zarovnání ve sloupcích a v řádcích.

Balík **booktabs** je založen na myšlence, že problém rozestupů řádků se nejčastěji objevuje při kolizích s linkami, které se používají až příliš často. Zvláštní příkazy v tomto balíku slouží k vytvoření vodorovných linek, ale starají se i o meziřádkové mezery. Dvojitě a svislé linky nejsou podporovány.

Balík **threeparttable** umožňuje tvořit tabulky, které mají tři části (název, tělo a poznámky) vždy o stejné šířce. Tělo tabulky může tvořit prostředí **tabular**, **tabular*** nebo **tabularx**. Pro poznámky je určeno prostředí **tablenotes**. Poznámky nejsou automatické, zvláště z důvodu, že v tabulkách často bývá vícenásobný odkaz na jednu poznámku.

Balík **csvtools** využívá automatizované zpracování údajů v souboru formátu CSV. Umožňuje opakovaně provádět sadu L^AT_EXových příkazů nad těmito soubory. Příkazy zpracují data v každém řádku zadaného souboru a použijí je k vytvoření L^AT_EXových dokumentů. Pomocí příkazu `\applyCSVfile` lze data vložit na požadovaná místa v dokumentu a vytvořit tak např. personalizované dopisy. Data ve formátu CSV lze samozřejmě převést i na tabulku. Balík k tomu poskytuje příkazy `\CSVtotabular` a `\CSVtolongtable`, které vytvoří příslušné prostředí a každý řádek z CSV souboru je zpracován na jeden řádek tabulky.

Balík **csvtools** bez problémů funguje s jinými balíky. Neposkytuje však možnost nastavit parametry pro jednotlivá pole či řádky tabulky, protože řádky se generují automaticky a všechny jsou formátovány stejně. Balík má i některá další omezení (například příkazy není možné vnořovat), jinak se ale jedná o velice užitečný nástroj pro hromadné zpracování dat a tabulek.

Tabulka 1: Balík csvtools

Obsah souboru vyrobky.csv:

```
Vyrobek,EUR,Kurs,CZK
A,120,24.64,2956.80
B,96,24.58,2359.68
C,56,24.32,1361.92
D,31,24.09,746.79
Celkem,303,--,7425.19
```

Zápis ve zdrojovém souboru:

```
\CSVtotabular{vyrobky.csv}{lrrr}
{Výrobek & Cena v~EUR & Kurs CZK/EUR & Cena v~CZK\\ \hline}
{\insertVyrobek & \insertEUR & \insertKurs & \insertCZK \\}
{\insertVyrobek & \insertEUR & \insertKurs & \insertCZK \\}
```

Výsledek:

Výrobek	Cena v EUR	Kurs CZK/EUR	Cena v CZK
A	120	24.64	2956.80
B	96	24.58	2359.68
C	56	24.32	1361.92
D	31	24.09	746.79
Celkem	303	–	7425.19

4. Kompatibilita tabulkových balíků

Podpora pro práci s tabulkami je dostatečná, ale jednotlivé dílčí problémy jsou řešeny pomocí samostatných balíků. Uživatel sice může využít několik těchto balíků, některé z nich ale mohou navzájem kolidovat.

Z uvedených balíků bylo vybráno dvanáct takových, které jsou pro zpracování tabulek nejprůběšnější. Těmito balíky lze obsloužit všechny uvedené oblasti sazby tabulek a zároveň zde nedochází k duplicitám (viz tab. 2).

Jistým problémem při použití většího množství balíků je kompatibilita. Některé balíky například předefinovávají určitá nastavení (každý balík svým způsobem), což pak znemožňuje jejich spolupráci. Kompatibilita balíků proto byla podrobena analýze, jejíž výsledky shrnuje tab. 3. Jednička v průsečíku řádku a sloupce znamená, že dané balíky spolupracují, nula vyjadřuje nemožnost spolupráce. Pomlčka označuje dvojici balíků, které ani spolupracovat nemohou (např. `tabularx` a `longtable`).

Zvláštní pozornost by měla být věnována kombinacím označeným nulou. Následující přehled uvádí, k jakým problémům zde dochází.

Tabulka 2: Přehled balíků pro jednotlivé vlastnosti tabulek

Vlastnost	tabularx	array	longtable	hhline	bigstrut	multirow	dcolumn	numprint	colortbl	ftcap	blkarray	csvtools
Umístění			1									
Rozměry	1		1									
Skupiny polí						1					1	
Zarovnání		1					1	1				
Světlo					1							
Ohraničení		1		1								
Barvy									1			
Popisek			1							1		

Tabulka 3: Přehled kompatibility L^AT_EXových balíků

	tabularx	array	longtable	hhline	bigstrut	multirow	dcolumn	numprint	colortbl	ftcap	blkarray	csvtools
tabularx		1	–	1	1	1	1	1	1	1	–	0
array			1	1	1	1	1	1	1	1	0	1
longtable				1	1	1	1	1	1	–	–	1
hhline					1	1	1	1	1	1	1	1
bigstrut						1	1	1	1	1	1	1
multirow							0	1	1	1	1	1
dcolumn								0	1	1	0	1
numprint									1	1	1	1
colortbl										1	0	1
ftcap											1	1
blkarray												0

Tabulka 4: Ukázka sazby bez balíků

Výrobek	Cena v EUR	Kurs	Cena v CZK
A	120	24.64	2956.80
B	96	24.58	2359.68
C	56	24.32	1361.92
D	31	24.09	746.79
Celkem	303	–	7425.19

- **dcolumn** a **multirow** – balík **multirow** sloučením polí ve sloupci naruší sloupec, je tedy narušeno i původní zarovnání. Oba balíky lze použít v jedné tabulce, ne však v jednom sloupci.
- **dcolumn** a **numprint** – každý balík pracuje s čísly zcela jiným způsobem, nelze tedy mít v jednom sloupci čísla zarovnaná na desetinnou čárku *a zároveň* s oddělovačem po třech číslicích (to lze pouze v případě, že čísla neobsahují desetinnou čárku). Je však možné použít tyto balíky v různých sloupcích téže tabulky.
- **blkarray** a **dcolumn**, **array**, **colortbl** – **blkarray** je experimentální balík, který nastavuje určité parametry jinak než balík **array**. Není tedy kompatibilní ani s **array**, ani s balíky **dcolumn** a **colortbl**, které jsou od **array** odvozeny.
- **csvtools** a **tabularx**, **blkarray** – balík **csvtools** obsahuje příkazy pouze pro vytvoření prostředí **tabular** a **longtable**, neumí tedy vytvořit prostředí **tabularx** a **blockarray** (z balíku **blkarray**).

Kromě výše uvedených nedostatků balíky pracují správně a lze je vzájemně kombinovat, což dává velké množství možností. To umožňuje vytvářet tabulky podle přání uživatele a přitom podle typografických pravidel. Ukázka 4 a 5 nastiňuje možnosti sazby tabulek, přičemž v prvním případě nejsou použité rozšiřující balíky, zatímco ve druhém případě bylo využito balíků **array**, **dcolumn**, **bigstrut**, **multirow**, **color** a **colortbl**.

5. Automatizovaná sazba tabulek

Z uvedeného přehledu je patrné, že možnosti sazby tabulek jsou sice rozsáhlé, ale zároveň nepřehledné. Použití rozšiřujících balíků, které by vedlo k optimálnímu výsledku, navíc po uživateli vyžaduje typografické znalosti sazby tabulek, což není vždy samozřejmé. (Naopak se mnohdy setkáváme se zcela nevhodným řešením.) Situaci se snaží řešit doplňky programových editorů, jako jsou průvodci

Tabulka 5: Ukázka sazby s balíky

Výrobek	Ceny výrobků		
	Cena v EUR	Kurs	Cena v CZK
A	120	24,64	2956,80
B	96	24,58	2359,68
C	56	24,32	1361,92
D	31	24,09	746,79
Celkem	303	—	7425,19

pro vytvoření tabulky. Zvláštním případem je editor LaTeX¹, který je specializovaný na vytvoření tabulky. Uživatel pracuje v grafickém rozhraní a program generuje zdrojový kód pro L^AT_EX. Editor však nabízí opět jen základní možnosti nastavení, další vylepšení již musí provést sám uživatel. Tento postup je sice bezproblémový pro zkušené uživatele, ale pro začátečníky by byl vhodnější nástroj, který by již pokročilé možnosti nastavení sazby tabulek zahrnoval a pokud možno automatizovaně uplatňoval.²

V současné době vzniká aplikace, která by měla sloužit jako průvodce tvorbou dokumentů, a to především pro začínající uživatele. Cílem je usnadnit proces tvorby dokumentů, přičemž důraz je kladen na typografická pravidla a celkovou úpravu. Jelikož obvyklou součástí dokumentů jsou tabulky, bylo by vhodné do aplikace zahrnout i prostředek pro návrh tabulky. Uživatelsky přívětivé prostředí by umožnilo uživateli nastavit parametry sazby v souladu s typografickými pravidly a o samotnou sazbu by se postarala vhodná kombinace L^AT_EXových balíků. Výsledkem by měl být vhodně upravený dokument s vhodně vysazenými tabulkami.

6. Závěr

Sazba tabulek, zejména rozsáhlejších, je komplikovanou záležitostí. Velké množství doplňujících balíků tuto situaci dále znepřehledňuje. Z dostupných balíků proto byly vybrány ty, které mohou být nejvíce nápomocné, a byla provedena analýza jejich kompatibility. Většina balíků bez problémů spolupracuje a umožňuje tak vytvářet přehlednější a lépe čitelné tabulky podle typografických pravidel. Tyto balíky se mohou v souladu s typografickými pravidly uplatnit v aplikaci, která bude sloužit pro efektivní přípravu dokumentů.

¹<http://www.ctan.org/tex-archive/help/Catalogue/entries/latex.html>

²Poznámka redakce. Za pozornost také stojí konvertory různých kvalit z tabulkových procesorů, např. Excel2L^AT_EX, viz <http://www.ctan.org/tex-archive/support/excel2latex/>.

Příspěvek byl zpracován za podpory grantu Interní grantové agentury MZLU v Brně č. 3/2008.

Reference

- [1] DYNȦAR, J. A KOL. *Technologie ruční sazby I*. 1. vydání. Praha: Státní nakladatelství technické literatury, 1955.
- [2] FELICI, J. *The Complete Manual of Typography*. Berkeley: Peachpit Press, 2003. ISBN 0-321-12730-7.
- [3] GOOSSENS, M., MITTELBACH, F., SAMARIN, A. *The L^AT_EX Companion*. Reading: Addison-Wesley, 1994. ISBN 0-201-54199-8.
- [4] TALANDOVÁ, P. *Přístupy ve zpracování tabulek v systémech DTP*. Diplomová práce. Brno: MZLU v Brně, 2006.
- [5] WILLIAMS, G. *The T_EX Catalogue Online* [online]. 2006 [cit. 2008-08-08]. Dostupné z <http://ftp.cstug.cz/pub/tex/CTAN/help/Catalogue/bytopic.html>.

Summary: Typesetting Possibilities for Tables

The paper deals with the typesetting of tables. It briefly describes packages prepared for tables and for the modification of individual characteristics of tables. Chosen packages that can be contribute most for typesetting and analysis of their compatibility is made. Almost all chosen packages work together and extend the possibilities of typesetting. Examples of typesetting with and without these packages show the potential of table typesetting.

T_EX je znám zejména v akademickém světě a používán při psaní odborných publikací. Mnoho lidí ví, že pomocí T_EXu lze vytvořit kvalitní sazbu. V dnešní době, kdy převládají programy s grafickým uživatelským rozhraním, v němž jsou důležité informace před uživatelem skryty, je však obtížné získat návod, jak připravit pomocí T_EXu soubor pro osvit či digitální tisk. V přednášce budou předvedeny postupy, jak lze v sazečské praxi využít L^AT_EX. Bude představeno několik nových makrobálíčků, které usnadňují přípravu letáků, pozvánek i sazbu knih včetně obálek a přebalů.

1. Úvod

Sazečská práce L^AT_EXisty má dva aspekty. První aspekt je typografický. Pravidla typografie však nejsou závislá na použitém nástroji. S nástrojem buď zacházet umíme, nebo neumíme. Neexistuje L^AT_EXová typografie, máme jen typografii jako takovou. Pokud výsledek není kvalitní, nebývá to chybou nástroje. Druhý aspekt je ryze technický. Mít dobrý nástroj nestačí. Máme-li dosáhnout kvalitního výsledku, musíme s tím nástrojem umět pracovat.

Typografická pravidla jsou skoro stejně stará jako písmo samo. Učebnice typografie tedy najdeme celkem snadno. V tomto článku se proto zaměříme výhradně na vysvětlení technických postupů, a to jen těch, kdy řešíme jistou problematiku pomocí L^AT_EXu. Takové návody jsou totiž různě roztrženy po internetu i jiných zdrojích. Navíc mají úzkou návaznost na technická zařízení, jež se rychle vyvíjejí, a návody tudíž zastarávají.

Uživatel komerčních sázecích systémů je v poněkud jiné situaci. V grafickém uživatelském rozhraní zvolí, pro jaké výstupní zařízení se má sazba připravit. Program nastaví správné parametry, které sazeč vůbec nemusí znát. Nemusí vůbec rozumět principu, jak dané zařízení pracuje. T_EX žádné takové rozhraní nemá. Uživatel musí vědět, jak má výstupní ovladač instruovat, aby výsledek jeho práce byl použitelný. Tato zdánlivá nevýhoda se změní ve výhodu v okamžiku, kdy se objeví nové zařízení. Uživatel komerčního programu, který takové zařízení v nabídce nemá, si musí koupit novou verzi pracovního nástroje, zatímco T_EXista jen ve svém postupu změní několik parametrů.

Konkrétní metody použití L^AT_EXu ve spojení se současnými výstupními zařízeními (osvitovými jednotkami, tiskovými stroji CTP) také nutně zastarávají. Zaměříme se tedy spíše na logiku použití L^AT_EXu, abyste metody, vysvětlené v tomto článku, mohli v budoucnu modifikovat a použít s modernější technikou.

2. Rozvržení stránky

$\text{\TeX}$ tradičně počítá souřadnice od počátku vzdáleného (1 in, 1 in) od levého horního rohu papíru. $\text{\LaTeX}$ ista však může být v roli pouhého sazeče majícího za úkol vytvořit sazbu podle dodaného grafického návrhu. Grafik jen zřídka přemýšlí v $\text{\TeX}$ ových pojmech. Téměř vždy měří rozměry tiskového zrcadla od okraje papíru. Když pak po několikáté píšete v preambuli svého dokumentu

```
\addtolength{\topmargin}{-1in}  
\addtolength{\oddsidemargin}{-1in}  
\addtolength{\evensidemargin}{-1in}
```

usoudíte, že nadešel čas k tomu, abyste si vytvořili vlastní balíček maker, který přizpůsobí zadávání rozměrů sazby přemýšlení grafika. Při analýze problému pokročíte ještě dále. Grafik někdy určí velikost obou okrajů, ale $\text{\LaTeX}$ očekává zadání levého okraje a šířky textu. Je tedy na uživateli, aby údaje od grafika převedl na hodnoty vyžadované $\text{\LaTeX}$ em. Totéž platí pro horní a dolní okraj. Protože rozměry sazby nastavujeme v každém dokumentu, je volba pohodlného balíčku maker nejdůležitější úlohou každého sazeče.

2.1. Proč ne geometry?

Odpověď na otázku, položenou v názvu této sekce, je do značné míry subjektivní. Na balíčku **geometry** není vůbec nic špatného. Osobně jej odmítám v podstatě z důvodů historických.

Vlastní makra pro nastavení rozměrů sazby jsem začal vyvíjet hned na počátku své $\text{\LaTeX}$ ové éry, tj. v době, kdy balíček **geometry** ještě neexistoval. Makra jsem vyvíjel podle svých vlastních potřeb, aby se mi s nimi dobře pracovalo. Logika použití balíčku **geometry** je velmi podobná tomu, co v mé dílně vzniklo postupným vývojem během 17 let. Přestože jsou možnosti balíčku **geometry** velké, chybí mi v něm vlastnosti, které pro svoji práci potřebuji. Balíček je dosti rozsáhlý a budování jeho nadstavby s rizikem, že se budoucí verze mohou lišit v implementaci některých maker, je pracnější než pokračování vývoje vlastního balíčku. Výsledkem vývoje, během něhož byl několikrát odeslán soubor do míst, kam chodí mrtvé bajty, a napsán znovu od začátku, je balíček **zwpagelayout**, jehož použití se zdá být zcela univerzálním.

V okamžiku psaní tohoto článku je balíček již odladěn a rutinně používán, ale není zveřejněn, protože není dokončena dokumentace. Hlavní vlastnosti balíčku si popíšeme v následujících sekcích. Zveřejnění balíčku na CTAN bude oznámeno na webu autora [13].

2.2. Rozměry papíru

Běžné distribuce T_EXu dovolují nastavení defaultních rozměrů papíru na A4, nebo letter. V typografické praxi si však s těmito formáty nevystačíme. Několik voleb najdeme ve standardních L^AT_EXových třídách, pravděpodobně bychom našli též existující balíčky. Náš balíček by tím však byl neúplný. Rozměry standardních formátů nechceme nosit v hlavě. Pomocí parametrů tedy můžeme zvolit libovolný formát z řad A, B, nebo C, ale i některé formáty americké. Můžeme tedy psát:

```
\usepackage[b5]{zwpagelayout}
```

Orientaci papíru můžeme ovlivnit použitím parametru PORTRAIT, nebo LANDSCAPE. Standardně se předpokládá orientace na výšku.

Občas potřebujeme tisknout na papír jiných rozměrů. Parametr PAPERSIZE nám umožní zadat libovolnou velikost. Při tvorbě vizitky standardní velikosti tedy použijeme:

```
\usepackage[papersize={9cm,5cm}]{zwpagelayout}
```

Tím ještě možnosti definice velikosti nekončí, ale více si ukážeme v sekci 2.5, kde budeme vytvářet obálku a přebal knihy.

Příkazy pro nastavení velikosti papíru uloží hodnoty do registrů `\paperheight` a `\paperwidth`. Tyto hodnoty jsou v okamžiku zpracování `\begin{document}` použity k nastavení rozměrů tiskového média podle toho, jaký výstupní ovladač je detekován. Balíček umí nastavit papír pro pdfL^AT_EX, dvips a (x)dvi_{pdfm}(x). Nastavení rozměrů pro výstupní ovladač je odloženo, protože některé balíčky, např. `hyperref`, nastavují rozměry podle svého uvážení. Zmíněné ovladače se bez výjimky řídí posledním příkazem, takže `zwpagelayout` vyhrává.

2.3. Ořezové značky

Ořezové značky potřebujeme zejména v případě, kdy tiskneme na papír nestandardních rozměrů. Starší balíček [14] byl závislý na postscriptu. Přitom ořezové značky lze tvořit prostým primitivem `\vrule`, tedy nezávisle na výstupním ovladači. Umístění ořezových značek odvodíme z hodnot registrů `\paperwidth` a `\paperheight`. Je tedy přirozené, že makra pro ořezové značky nadefinujeme v balíčku `zwpagelayout`.

Pro tvorbu ořezových značek jsou nejdůležitější čtyři parametry. Parametr CROPMARKS zapne tisk ořezových značek. Programově je to realizováno tak, že se zvětší hodnoty `\paperwidth` a `\paperheight` a celá sazba se použije nastavením kladných hodnot do registrů `\hoffset` a `\voffset`. Běžný uživatel o těchto registrech neví a vzdělanější uživatel si musí pamatovat, že jsou pro něj tyto registry tabu.

Při vizuální kontrole se hodí, když máme okolo stránky rámeček. Ten zapneme parametrem CROPFRAME. Ve finální verzi jej samozřejmě vypneme.

Standardně se očekává jednobarevný tisk, takže ořezové značky jsou tištěny černě. Barvu zapneme parametrem `COLOR`. I zde máme mnoho dalších možností, o nichž se zmíníme později.

Kromě ořezových značek potřebujeme též stránky označit vhodným identifikačním textem. Takový text nastavíme v parametru `CROPTITLE`. Pokud text obsahuje čárku, musíme jej vložit do složených závorek. Při zpracování parametrů ignoruje příkaz `\usepackage` mezery, proto je musíme vložit použitím kontrolního symbolu `\_`.

Příklady použití uvidíte v následujících sekcích tohoto článku. Proto jsme úvod k ořezovým značkám předřadili.

2.4. Rozměry tiskového zrcadla

Nastavení levého a pravého okraje při oboustranné sazbě je ve standardním $\text{\LaTeX}$ u poněkud těžkopádné. Uživatel musí vypočítat správné hodnoty `\oddsidemargin`, `\evensidemargin` a `\textwidth`. Balíček `zwpagelayout` nastavení usnadňuje. Předpokládá se, že nastavujeme rozměry textu na liché straně, tj. té, která je v otevřené knize vpravo. Okraje na protilehlé straně budou dopočítány symetricky. Rozměry nastavíme pomocí parametrů `LEFTMARGIN`, `RIGHTMARGIN` a `TEXTWIDTH`. Měříme od kraje papíru. Protože jsme však již rozměr papíru zadali, uvedeme pouze dvě z výše uvedených hodnot. Zadáme-li pouze jeden rozměr, předpokládá se, že levý i pravý okraj jsou stejné.

Horní a dolní okraj nastavíme pomocí parametrů `TOPMARGIN` a `BOTMARGIN`. Parametry `HEADHEIGHT`, `HEADSEP` a `FOOTSKIP` slouží k nastavení rozměrů odpovídajících parametrů $\text{\LaTeX}$ u. Parametrem `TEXTHEIGHT` nastavíme výšku textu včetně záhlaví a zápatí. Opět stačí zadat dva z trojice rozměrů, třetí rozměr se dopočítá z rozměru papíru. Hlavní odlišnost však spočívá v tom, že při sazbě knih většinou dodržujeme řádkový rejstřík. Hodnotu rozměru `\textheight` proto musíme vypočítat jako násobek `\baselineskip` zvětšený o `\topskip`. Při sazbě akcidenčních tiskovin však takovou korekci obvykle nechceme. Zakážeme ji parametrem `STRICTHEIGHT`. Bez tohoto parametru algoritmus implementovaný v balíčku `zwpagelayout` zmenší hodnotu `\textheight`. Aby zůstal zachován součet všech rozměrů určených uživatelem, zvětší se o rozdíl hodnota registru `\footskip`. Pokud má kniha živé záhlaví a nemá zápatí, dali bychom přednost úpravě velikosti `\headsep`. Dosáhneme toho parametrem `ADJUSTHEADSEP`.

Oba způsoby zarovnání jsou ukázány na obrázku 1. Ukázka vlevo byla vytvořena příkazem:

```
\usepackage[c8,margins=6mm,headheight=4mm,headsep=4mm,
  cropmarks,cropframe,croptitle=Adjust\footskip]{zwpagelayout}
```

Ukázku vpravo jsme získali pomocí:

Obrázek 1: Vyrovnání řádkového rejstříku úpravou \footskip (vlevo) a \headsep (vpravo)

```
\usepackage[c8,margins=6mm,headheight=4mm,headsep=4mm,
adjustheadsep,
cropmarks,cropframe,croptitle=Adjust\ headsep]{zwpagelayout}
```

Všimněte si použití parametru `MARGINS`. Ten slouží k nastavení všech okrajů na stejnou hodnotu.

2.5. Obálka a přebal

V sekci 2.4 jsme uvedli, že chybějící hodnoty rozměrů tiskového zrcadla lze dopočítat z velikosti papíru. Balíček `zwpagelayout` však disponuje i opačnou funkcí: výpočtem velikosti papíru z rozměrů tiskového zrcadla a okrajů. Lze dopočítat oba rozměry, nebo jen jeden z nich. Tuto možnost oceníme zejména při tvorbě knižních obálek a přebalů.

V sazečské praxi se často octneme v situaci, kdy kniha ještě není hotova, není vybrán papír, takže nelze určit tloušťku hřbetu. Přesto by zákazník chtěl vidět návrh obálky a přebalu. Tloušťku hřbetu lze pouze přibližně odhadnout. Jaká je tedy celková šířka přebalu? Pokud si ji vypočteme sami, můžeme se dopustit chyby. Automatický výpočet nás tohoto rizika zbaví. Přebal, který vidíte na obrázku 2, jsme vytvořili následujícím kódem:

Výšku přebalu známe, proto jsme ji mohli zadat, vynechali jsme pouze šířku. Povšimněte si též použití parametru `BOTMARGIN` bez hodnoty. Spodní okraj pak převezme velikost z parametru `TOPMARGIN`. Parametr `SPINE` určuje tloušťku hřbetu. Tím se též mění význam parametru `TEXTWIDTH`, jímž nyní určujeme šířku přední, resp. zadní strany. Celková šířka (bez prostoru pro ořezové značky) tedy bude 148 mm. Pro kontrolu využijeme též rámeček, jenž vyznačí nejen obrysy přebalu, ale i chlopně a hřbet. Hodnoty uvedené v parametrech jsou uschovány v makrech, která využijeme pro nastavení rozměrů jednotlivých boxů. Když se pak dozvíme skutečnou tloušťku hřbetu, změníme pouze hodnotu parametru `SPINE`. Celá sazba přebalu se pak přepočte automagicky.

Při tvorbě přebalu jsme si též představili balíček `ean13isbn` [12] pro tisk čárového kódu a ISBN ve formátu platném od 1. ledna 2007.

3. Barevný tisk

$\text{\TeX}$ nemá žádné nativní nástroje pro práci s barvou. Práce s barvou je závislá na schopnostech výstupního ovladače, jemuž $\text{\TeX}$ odesílá povely primitivem `\special`, pdf $\text{\TeX}$ lze ovládat pomocí primitivu `\pdfliteral`, což bylo přehledně popsáno v článku F. Chvály [5]. Balíček `color` obsahuje makra pro změnu barvy písma, jež expandují na příslušné povely pro daný výstupní ovladač. V dokumentaci najdete popis příkazů, ale nedočtete se nic o tom, jak připravit soubor pro osvit.

Obrázek 2: Ukázka přebalu

Možností použití barev v tiskovinách je několik. V dalších sekcích si probereme hlavní možnosti a ukážeme jejich realizaci v L^AT_EXu. Použití barvy je však poměrně komplikovaná tematika, zde bude vysvětlena pouze její část.

3.1. Jednobarevný ofsetový tisk

Jednobarevný ofsetový tisk je z hlediska T_EXu zcela jednoduchý. Od sazeče vlastně nic speciálního není očekáváno. Sazba je připravena černě, jako by se o barevnou sazbu nejednalo. Vše se odehraje až při tisku. Místo černé barvy tiskař nalije do stroje barvu jinou.

3.2. Plnobarevný tisk

Balíčky pro práci s barvou nabízejí použití dvou různých barevných prostorů, RGB a CMYK. Z tabulky v článku F. Chvály [5] by se mohlo zdát, že na volbě barevného prostoru nezáleží. Opak je však pravdou. Tiskové stroje pracují výhradně v prostoru CMYK. Navíc teoretické poučky o míchání barev platí jen přibližně. Skutečné procesní (tiskové) barvy se od ideálních, spektrálně čistých barev, na nichž jsou teoretické poučky založeny, značně liší. Směs stejných množství procesních barev tedy nedává šedé, ale hnědé tóny. Černá se přidává ze dvou důvodů, jednak proto, abychom skutečně mohli tisknout šedé tóny, a jednak proto, že jedna barva schne rychleji než směs tří tiskových barev. Situace je však ještě složitější, protože existuje více prostorů RGB a více prostorů CMYK. Zejména rozdíl mezi evropskými prostory Euroscale a americkými SWOP je značný. Pro přepočty mezi barevnými prostory neplatí žádný jednoduchý vzoreček, ale používají se ICC profily [6]. Problematika však přesahuje rámec L^AT_EXového řešení. Srozumitelné vysvětlení s odkazy na další zdroje najdete v seriálu *Grafika v UNIXu* [2].

Pamatujme si tedy, že sazbu barevného dokumentu musíme připravit v prostoru CMYK, a to v tom správném, který odpovídá použitým tiskovým strojům. Nezbyvá tedy než navštívit tiskárnu, kde se bude tiskovina vyrábět, zjistit, jaké stroje budou použity, a případně z tiskárny získat ICC profil, nebo vytištěný barevný vzorník.

Plná barevnost je realizována tiskem čtyřmi procesními barvami, azurovou (CYAN), purpurovou (MAGENTA), žlutou (YELLOW) a černou (BLACK). Barevný dokument musíme tedy rozdělit na čtyři barevné pláty. Než však začnete studovat, jak se barevné separace v praxi dělají, poraďte se s pracovníky DTP studia, kde se bude soubor zpracovávat. Separace jsou totiž v současné době vyžadovány jen zřídka. Osvětové jednotky očekávají na vstupu jediný soubor, jaký byste tiskli doma na vlastní inkoustové či laserové tiskárně. Může to být postscript, nyní stále častěji PDF. Algoritmy pro vytvoření separací obsahuje software osvětové jednotky. Digitální tiskové stroje (CTP) pracují, laicky řečeno, v podstatě stejně jako domácí barevné tiskárny, jsou jen větší a rychlejší. Ani v tomto případě proto separace dělat nebudeme.

Postscript [10] obsahuje řadu operátorů pro převod mezi barevnými prostory. Očekávali bychom tedy, že barvu při zadání barvy v jakémkoliv prostoru osvitka vše zařídí pomocí operátorů `/setblackgeneration`, `/setundercolorremoval` a jiných. Jak jsme si již řekli, barevných prostorů je mnoho. DTP studia nechtějí riskovat, že barvy převedou z jiného RGB do jiného CMYK. Barevné konverze jsou proto záměrně vypnuty a je na sazeči, aby sám převedl barvy do správného barevného prostoru, z nichž osvitová jednotka jen triviálním postupem vytvoří barevné separace.

3.3. Dvoubarevný a třibarevný tisk přímými barvami

V barevném prostoru CMYK lze realizovat jen omezený počet barev. Sytější barvy lze sice fyzicky namíchat, ale v prostoru CMYK je vytisknout nemůžeme. Pokud nepotřebujeme plnou barevnost, můžeme tisknout přímými barvami. Bývá to výtvarně zajímavější i levnější.

Pro tisk musíme opět připravit separace. Vytváření vlastního software, nebo definice přímých barev tak, aby tomu běžné programy rozuměly, může být namáhavé. Lze však simulovat přímé barvy pomocí barev procesních. Ukažme si to na jednoduchém dokumentu, který vidíte na obrázku 3. Zde máme černý a červený tisk na žlutém pozadí¹. Červenou barvu přiřadíme purpurové separaci (MAGENTA). Azurová (CYAN) zůstane nepoužita.

Balíček `zwpagelayout` při barevném tisku předpokládá prostor CMYK, ale barvy lze předefinovat. V parametru `COLORS` uvedeme jména barev a jejich hodnoty v barevném prostoru použitém pro osvit, v našem případě to bude defaultní CMYK. Zde je celý zdrojový kód:

V DTP studiu pak řekneme, že azurový plát je prázdný a nemá se svítit. Aby skutečně zůstal prázdný, předefinovali jsme parametrem `CROPCOLOR` barvu ořezových značek.

Osvitové jednotky mívají zapnutý přetisk černé. Na žlutém plátu proto vidíte bílý nápis v místě, kde bude červený text. V místě černého textu však bílý nápis nenajdeme. Před přiřazením přímých barev procesním barvám tedy musíme trochu přemýšlet.

3.4. Barevné separace pomocí maker

Polotónový tisk vyžaduje použití rastru. Pokud navíc tiskneme barevně, musíme též měnit úhel rastru. Postará se o to software osvitové jednotky. Použití rastru však snižuje čitelnost písma. Pokud se přímé barvy nepřekrývají, bývá výhodnější, když rastr použit není, nebo má úhel 45°, což poskytuje nejlepší vizuální vjem.

¹V nebarevně vytištěném Zpravodaji si musíte barvy domyslet.

Obrázek 3: Ukázka tisku přímými barvami; neseparovaný dokument s kontrolním rámečkem (nahore) a žlutý plát (dole)

Pak ovšem nelze využít standardní software, ale separace si musíme udělat sami. Jednou z možností je řešení na úrovni maker. Tak jsme udělali žlutý plát na obrázku 3. Zde je kompletní kód:

Všimněte si, že jsme změnili barevný prostor a předefinovali všechny barvy. Dokument pak zůstal stejný. Obecný postup, jak dělat separace, nejspíš neexistuje. Stačí však trochu invence k vytvoření vlastních maker.

3.5. Barevné separace pomocí PostScriptu

V úvodu jsem si řekl, že se budeme zabývat pouze L^AT_EXovými řešeními. Zde uděláme výjimku. Způsob míchání barev může být komplikovaný a programování vlastních separací pomocí makrojazyka T_EXu zbytečně pracné. Pokud při zpracování dokumentu využíváme ovladač **dvips**, nabízí se ještě jedna zajímavá možnost: separaci barev provést balíčkem Aurora [3].

4. Příprava ke korekturám

Korektury jsou nezbytnou součástí práce. Korektury nikdy neděláme na obrazovce, vždy ve vytištěné verzi. A neděláme je sami, navíc stejně potřebujeme imprimatur od autora či nakladatele. Formát knihy je často A5 či menší, zatímco naše stolní tiskárna používá formát A4. Abychom šetřili naše lesy, raději poskládáme na papír dvojice stránek tak, jak budou v otevřené knize, tj. liché stránky vpravo. První papír tedy bude obsahovat pouze stranu 1. Úlohu lze řešit mnoha různými způsoby. My si předvedeme jednoduché makro, které poskládá již hotový PDF.

První, co musí naše makro udělat, je zjištění počtu stran. Vyřešíme to pomocným makrem se dvěma parametry. Prvním parametrem je jméno PDF souboru, druhým parametrem je jméno sekvence, v níž bude počet stran uložen jako makro.

```
\def\getpdfnumpages#1#2{\pdfximage{#1}%  
  \edef#2{\the\pdfximagepages}}
```

Další pomocné makro vytiskne zvolenou stránku v rámečku.

```
\def\ZW@pdfbox#1#2{\fbox{\includegraphics[page=#1]{#2}}}
```

Vlastní definici uvedeme bez vysvětlení. Všimněte si, že jména maker ob sahují zavináče. Musíte proto vložit definici do stylového souboru, nebo mezi `\makeatletter` a `\makeatother`.

```
\def\DvaVedleSebe#1{\getpdfnumpages{#1}\ZW@pdfnumpages  
  \fboxrule 1pt \fboxsep -\fboxrule  
  \typeout{#1, number of pages = \ZW@pdfnumpages}}
```

```

\ZW@pdfpgnum2
\ vbox to \textheight{\hsize\textwidth
\ hbox to \textwidth{\hfill \ZW@pdfbox{1}{#1}}\vss}
\loop
  \ZW@tempnum\ZW@pdfpgnum
  \advance\ZW@tempnum1
  \ifnum\ZW@pdfpgnum<\ZW@pdfnumpages
    \ vbox to \textheight{\hsize\textwidth
    \ hbox to \textwidth{\hss
      \ZW@pdfbox{\number\ZW@pdfpgnum}{#1}%
      \ZW@pdfbox{\number\ZW@tempnum}{#1}\hss}\vss}
  \else
    \ vbox to \textheight{\hsize\textwidth
    \ hbox to \textwidth{%
      \ZW@pdfbox{\number\ZW@pdfpgnum}{#1}\hfill}\vss}
  \fi
  \ifnum\ZW@tempnum<\ZW@pdfnumpages
    \advance\ZW@pdfpgnum2
  \repeat}

```

Makro potřebuje pro svoji práci dva čítače:

```

\newcount\ZW@pdfpgnum
\newcount\ZW@tempnum

```

5. Čárové kódy

V sekci 2.5 jsme si předvedli čárový kód s ISBN. Není to zdaleka jediné použití čárového kódu. Jinou aplikaci představili Sojka a Hrad [4]. Čárových kódů však existuje více druhů [1] a vzhledem k jejich důležitosti byly vytvořeny komerční i volně šiřitelné fonty. Sazba čárových kódů pomocí fontů ovšem naráží na různá úskalí. Čárové kódy skupiny 2/5 a Code39 mají velmi variabilní rozměry. Liší se nejen poměr výšky a šířky, ale i poměr mezi slabou a tlustou čarou. Je dost pravděpodobné, že konkrétní snímač nebude schopen přečíst čárový kód vytisknutý fontem, který byl původně vytvořen pro jinou aplikaci.

Elegantní řešení nabízí balíček `ean13.tex` [7], který vytvořil Petr Olšák a popsal jej česky ve Zpravodaji [8] i anglicky v TUGboatu [9]. Čárový kód je vytvořen pomocí maker primitivem `\vrule`, žádný font není nutný. Makra lze parametrizovat. Čárový kód můžeme tedy snadno přizpůsobit požadovaným rozměrům.

Ze stejné myšlenky vychází balíček `makebarcode` [11]. V něm je definováno obecné makro, jež generuje čárový kód pomocí tabulky a několika parametrů.

Můžeme jej využít např. pro identifikaci záznamovým médií HP, jak je ukázáno na obrázku 4.

Obrázek 4: Čárový kód CLNU01L1 pro čistící kazetu HP

6. Závěr

Při čtení tohoto článku jste se mohli přesvědčit, že nasazení $\text{\LaTeX}$ v profesionální praxi, jakkoliv se zdá být složitým, je zvládnutelné. Vyžaduje dosti velké znalosti, při jejichž získávání nám pracovníci DTP studií, zvyklí na klikací programy, pomohou jen zřídka. Později však budeme omezeni jen hranicemi své invence. Chcete-li také proniknout do světa profesionální sazby kterýmkoliv $\text{\TeX}$ ovým formátem, řiďte se arabským příslovím z obrázku 5, který byl vysázen $\text{\XeTeX}$.

الشروع نصف العمل.

Obrázek 5: Arabské přísloví: „Začátek je polovina práce.“

Reference

- [1] Benadiková A., Mada S., Weinlich S.: *Čárové kódy, automatická identifikace*. Grada 1994, 272 pp., ISBN 80-85623-66-8.
- [2] Brabec S.: *Seriál Grafika v UNIXu*. <http://www.root.cz/serialy/grafika-v-unixu/>
- [3] Freeman T. G.: *Aurora: Colour Separation with PostScript Devices*. <http://mirror.ctan.org/support/aurora/>
- [4] Hrad M., Sojka P.: *Automatizace sazby a skenování formulářů*. Zpravodaj Československého sdružení uživatelů $\text{\TeX}$, **12** (3–4), 123–139 (2002).
- [5] Chvála F.: *O možnostech pdf $\text{\TeX}$* . Zpravodaj Československého sdružení uživatelů $\text{\TeX}$, **15** (1), 2–85 (2005).
- [6] International Color Consortium. <http://www.color.org>
- [7] Olšák P.: <http://mirror.ctan.org/macros/generic/ean.zip>
- [8] Olšák P.: *Čárové kódy EAN v $\text{\TeX}$* . Zpravodaj Československého sdružení uživatelů $\text{\TeX}$, **6** (2), 86–96 (1996).

- [9] Olšák P.: *The EAN barcodes by T_EX*. TUGboat 15 (December 1994), No. 4, pp. 459–464. <http://www.tug.org/TUGboat/Articles/tb15-4/tb45olsa.pdf>
- [10] *PostScript language reference manual / Adobe Systems Incorporated. — 3rd ed.* Addison-Wesley, 1999. ISBN 0-201-37922-8. <http://www.adobe.com/products/postscript/pdfs/PLRM.pdf>
- [11] Wagner Z.: *Čárové kódy 2/5 a Code39*. <http://mirror.ctan.org/macros/latex/contrib/makebarcode.zip>
- [12] Wagner Z.: *Čárový kód EAN13 s ISBN*. <http://mirror.ctan.org/macros/latex/contrib/ean13isbn.zip>
- [13] Wagner Z.: *Makra pro L^AT_EX*. <http://icebearsoft.euweb.cz/tex/>
- [14] Wagner Z.: *Ořezové značky*. <http://icebearsoft.euweb.cz/tex/#crop>

Summary: L^AT_EX in the Typographer's Profession

T_EX is known mainly in the academic world and is used for writing technical publications. Many people are aware of the possibility of creating high-quality typesetting with T_EX. However, these days when programs with graphical user interface hiding important information prevail, it is difficult to find instructions on how to prepare with T_EX a file for a phototypesetter or a digital printer. The lecture will demonstrate the methods of using L^AT_EX in practice. A few macro-packages that prepare leaflets and invitation cards will be discussed. Also the typesetting of books including their covers.

Zdeněk Wagner
<http://icebearsoft.euweb.cz>

Sbohem T_EXperience 2008!

Bud' vítána T_EXperience 2009!

Vážené T_EX kolegyně, vážení T_EX kolegové,
loučíme se se sborníkem a zveme, zveme, zveme...

Kdo to organizuje? ČSTUG a ÚSKM FaME UTB ve Zlíně.

Kdy to bude? Čtvrtek večer až neděle, 21. – 24. května 2009.

Kde to bude? Bungalovy a chaty na <<http://www.kralovec.cz/>>

Pro koho T_EXperience 2009 bude? Pro vás! Pro každého od vás!

Jsou známy detaily? Budou včas otištěny ve Zpravodaji.

Kde najdu přihlášku? Bude na <<http://www.cstug.cz/>>

Srdnatě voláme po přednáškách a příspěvcích!

Lákají vás Zdeněk Wagner a Pavel Stříž!

Už jen 127 spuštění TEXu...

Super, to už bude !!

Zpravodaj Československého sdružení uživatelů T_EXu
ISSN 1211-6661 (tištěná verze), ISSN 1213-8185 (online verze)

Vydalo: Československé sdružení uživatelů T_EXu
vlastním nákladem jako interní publikaci
Obálka: Antonín Strejc
Na obálce použito logo T_EXperience 2008,
jehož autory jsou
Igor Novozámský, Zdeněk Wagner a Pavel Stříž
Počet výtisků: 600
Uzávěrka: 14. 9. 2008
Odpovědný redaktor: Zdeněk Wagner
Redakční rada: Pavel Stříž, David Catto a výbor CSTUGu
Tisk a distribuce: KONVOJ, spol. s r. o., Berkova 22, 612 00 Brno,
tel. +420 549 240 233
Adresa: ČSTUG, c/o FEL ČVUT, Technická 2, 166 27 Praha 6
Tel: +420 224 353 611
Fax: +420 233 332 938
Email: 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

gacstug@cstug.cz
grantová agentura ČSTUGu

secretary@cstug.cz, orders@cstug.cz
korespondence administrativní síle sdružení, objednávky CD-ROM

cstug-members@cstug.cz
korespondence členům sdružení

cstug-faq@cstug.cz
řešené otázky s odpověďmi navrhované k zařazení do dokumentu ČSFAQ

bookorders@cstug.cz
objednávky tištěné T_EXové literatury na dobírku

ftp server sdružení:
<ftp://ftp.cstug.cz>

www server sdružení:
<http://www.cstug.cz>