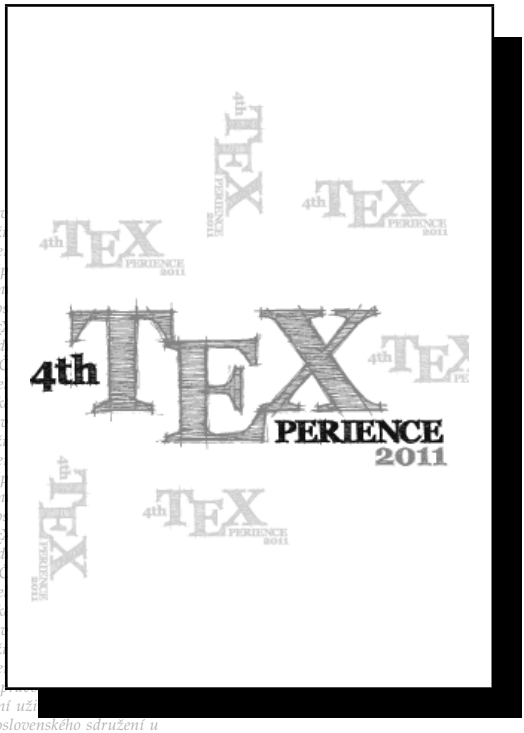
CSTUG



ZPRAVODAJ

Československého sdružení uživatelů TEXu

2

2012

ISSN 1211-6661 Tištěná verze

ISSN 1213-8185 Online verze

Ročník 22

OBSAH

Zpravodaj Československého sdružení uživatelů TEXu 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/`.

Zpravodaj je zařazen do Seznamu recenzovaných neimpaktovaných periodik vydávaných v České republice, viz `http://www.vyzkum.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, CD, či DVD na adresu:

Zdeněk Wagner
Vinohradská 114
130 00 Praha 3

`zpravodaj@cstug.cz`

Nezapomeňte přiložit všechny soubory, které dokument načítá (s výjimkou standardních součástí TEX Live), zejména v případě, kdy vás nelze kontaktovat e-mailem.

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

Úvodník ke konferenčnímu číslu

Tomáš Hála

Pořádání konferencí patří mezi tradiční aktivity Československého sdružení uživatelů TEXu.

Připomeňme semináře o Linuxu a TEXu (SLT) v letech 2001–2004 pořádané společně s CZLUG, zájmovým sdružením uživatelů Linuxu, obsahující jak společné plenární přednášky, tak příspěvky linuxové sekce a TEXové sekce.

Na přerušenou tradici jsme navázali díky obětavosti Pavla Stříže a jeho rodiny vlastní samostatnou konferencí. Pod názvem TEXperience se první ročník uskutečnil v roce 2008 v Jestřabí, druhý ročník proběhl 21. až 24. května 2009 v Královci.

Pro oživení konferenčního ducha, k udržení stávajících a navázání nových kontaktů, ale také i z organizačních důvodů, jsme v roce 2010 uskutečnili částečně společnou konferenci ConTEXt Meeting & TEXperience. Týdenní setkání v bývalém mlýně v Brejlově na břehu řeky Sázavy připravili především Jano Kula a Pavel Stříž. Z této akce obdrželi členové sdružení trojčíslo věnované příspěvkům předneseným v angličtině, ať již zahraničními účastníky, nebo našimi členy.

Obdobný model jsme použili i v loňském roce: členové České statistické společnosti a CSTUG spojili síly a pod vedením Miloše Brejchy a Pavla Stříže uspořádali společnou dvojkonferenci. Konala se od čtvrtka 29. září do neděle 2. října 2011 v příjemném prostředí hotelu Belvedér v Železné Rudě na Klatovsku. Zdařilé logo této konference reprodukujeme na obálce, autorem je Ondřej Krybus.

Nyní se připravuje další pokračování TEXperience, tentokrát péčí Jana Šustka a jeho spolupracovnic i spolupracovníků z Ostravské univerzity. Tradiční setkání proběhne v srpnu v Morávce na Frýdeckomístecku.

Všem organizátorům patří velký dík.

A co najdete v tomto čísle?

První dvě publikované práce byly představeny už v Brejlově 2010, byly však předneseny v češtině, a proto jsme je nemohli zařadit do již zmíněného trojčísla.

Následují materiály z TEXové sekce společné dvoukonference TEXperience '11 a STAKAN '11: trojici vybraných recenzovaných článků předcházejí abstrakty příspěvků, k nimž redakce bohužel nedostala plná znění. Ne vše je vždy zcela růžové. Rezervované místo proto doplňujeme jedním dalším textem.

V závěru čísla pak naleznete první část zprávy o výpravě do Bassenge-Boirs v Belgii na pátý ConTEXt Meeting na podzim 2011. Členové našeho sdružení

Foto: Jaroslav Hajtmar

jsou totiž také docela aktivní v účasti na akcích pořádaných spřátelenými zahraničními organizacemi, ať se již jedná o EuroTEX, BachoTEX, nebo již zmíněný ConTEXt Meeting.

Konference TEXperience, které pořádáme, jsou určeny všem uživatelů a příznivcům TEXu a příbuzného programového vybavení. Zúčastňuje se jich však jen menší část našich členů.

Závěrem mi proto dovolte vyslovit přání: Přijeďte příště také i Vy.

Tomáš Hála
jménem redakční rady

VyTEXčení – dávkové zpracování vysvědčení TEXem

Miroslav Olšák

VyTEXčení je TEXové makro (spolu s grafickou aplikací), které zpracuje data ze školy online a vytvoří pdf vysvědčení. Pomocí TEXu je možné ještě dělat ve vysvědčení dodatečné úpravy. Je to práce dělaná na objednávku Buďánce, což je škola, která měla specifické požadavky na tisk vysvědčení. Konkrétně potřebovala především sloučit češtinu a literaturu, mít známky v procentech/bodech, hodnotit studenty slovním komentářem a počítat celkový průměr. Z toho důvodu jí nevyhovovalo původní řešení přes Office.

Původně si správce nechal vygenerovat ze školy online (internetová aplikace) data, která otevřel ve Wordu (nebo něčem podobném). Tam bylo možné do dat ještě trochu zasahovat a následně je vytisknout. Použití školy online mělo zůstat stejné, jen ten Word měl být nahrazen TEXem. Správce přitom neměl s TEXem žádné zkušenosti.

1. XML

Vstupem pro TEXové makro je XML vygenerované školou online. Nesnažil jsem se tento formát parsovat a hledat párové a nepárové tagy. Pouze procházím jednotlivé tagy (zbytek ignoruji) a zkoumám, zda k nim mám přiřazené makro. Tato makra jsou buď jsou bez parametru nebo si uloží všechen text až do příslušného uzavíracího tagu.

Předpokládám, že data budou zhruba ve formátu:

```
<soubor>
  <bla></bla>
  <student>
    <jmeno>Jan</jmeno>
    <prijmeni>Novák</prijmeni>
    ...
    <predmet>
      <nazev>Krasopis</nazev>
      <body>42</body>
      <komentar>S tvou prací jsem spokojen, ale máš na víc.
      </komentar>
      ...
```

doi: 10.5300/2012-2/67

```
13      </predmet>
14      <predmet>
15        <nazev>Kreslení</nazev>
16        <body>78</body>
17        <komentar>Nepodceňuj domácí přípravu.</komentar>
18        ...
19      </predmet>
20      ...
21    </student>
22    <student>
23      <jmeno>Rudolf</jmeno>
24      <prijmeni>Mára</prijmeni>
25      ...
26      <predmet>
27        <nazev>Krasopis</nazev>
28        <body>81</body>
29        <komentar>Tvé písmo je jedno z nejúhlednějších
30        ve třídě, jen tak dál.</komentar>
31        ...
32      </predmet>
33      <predmet>
34        <nazev>Kreslení</nazev>
35        <body>100</body>
36        <komentar>Všechna tvá díla byla nápaditá
37        a propracovaná, nepolevuj.</komentar>
38        ...
39      </predmet>
40      ...
41    </student>
42    ...
43  </soubor>
```

Takže tagy, kterých si všímám jsou:

- otevírací tag studenta (inicializace)
- uzavírací tag studenta (vytisknutí studenta)
- otevírací tak předmětu (vynulování dat o předmětu)
- uzavírací tag předmětu (zpracování, zaboxování předmětu)
- otevírací tag údaje o studentovi (např. jméno) – načítám data až po uzavírací tag
- otevírací tag údaje o předmětu (např. body) – načítám data až po uzavírací tag

– uzavírací tag souboru (konec zpracovávání)

2. Hlavní soubor

Pro zTEXování vysvědčení je potřeba založit hlavní TEXový soubor, na který je TEX poštván. Jeho nejjednodušší verze vypadá následovně:

```
\input vytexceni     % načtu TeXová makra
\printxml oktava.xml % zpracuji xml
\bye
```

3. UTF8

Data ze školy online jsou v kódování pro TEX nestandardním – UTF8. TEX umí pochopit některé UTF8 znaky pomocí encTEXu. V něm už jsou předpřipravené české znaky (za předpokladu správného vygenerování formátu). Učitelé jsou však mazaní a do slovního komentáře dávají znaky všelijaké. Proto mám implementované varování v případě, že TEX narazí na UTF8 znak, který nezná. Tento znak pak je možné doimplementovat v souboru utfplus. Už tam mám uvozovky, různě dlouhé pomlčky a automatickou proměnu pomlčky ze spojovníku obklopeného mezerami. A ještě, když už používám ten encTEX, jsem použil (předpřipravené) automatické vlnkování. UTF8 kódování je třeba používat i v hlavním souboru.

4. Příkazy a podmínky

Poté, co jsou načteny údaje o předmětu a TEX narazí na uzavírací tag předmětu, pustí se na předmět zpracovávací makro. Toto makro definuje uživatel v hlavním souboru pomocí předpřipravených maker. Příklad, jak vytvořit takové makro:

```
\input vytexceni     % načtu TeXová makra

\ucitel Oto Oulík; Antonín Kejmar:
  \student Jan Novák:
    \setprocenta{50} +
    \nazev Kreslení:
      \smaz
\nazev Krasopis:
  \setucitel{Jaromír Kozina}
```

```
56
57 \printxml oktava.xml % zpracuji xml
58 \bye
```

Následek tohoto kódu bude, že předmětům studenta Jana Nováka, které učí Oto Oulík nebo Antonín Kejmar bude přiřazeno bodové ohodnocení 50 a předmět Kreslení s touto vlastností bude smazán. Dále předmětu Krasopis bude přiřazen Jaromír Kozina.

Makra, která ono upravovací makro vytváří, se dělí na příkazy a podmínky. Příkaz provede s předmětem jednoduchou změnu údaje, podmínka na základě aktuálních údajů zúží množinu předmětů, na které se příkaz vykoná.

Příkaz implicitně tuto množinu opět rozšíří na všechny předměty, výjimkou je, když za příkazem následuje znak plus. Příkazu se typicky dává parametr v obvyklých TEXovských svorkatých závorkách, výjimkou je příkaz `\smaz`, který parametr nemá.

Podmínka si parametr přečte po dvojtečku. Parametr podmínky je možné rozdělit středníkem, pak je podmínka splněna, pokud vyhovuje alespoň jeden úsek. Na začátku každého úseku může být `\not`, což podmínku zneguje.

5. Průměrování, slučování

Další požadavek byl, aby bylo možné sloučit dva předměty do jednoho, příčemž body budou zprůměrovány. Dále se má počítat celkový průměr všech předmětů. To vše řeším komplexním příkazem `\prumer`. Tento příkaz si uloží údaje o předmětu. Po zpracování všech předmětů založí nový předmět.

Aby tento příkaz zapadal do ostatních, má také jeden parametr, který v nejjednodušší podobě určuje název výsledného předmětu. Následující příklad tedy na konec vysvědčení přidá předmět bez učitele a komentáře s názvem Průměr a zprůměrovaným počtem bodů.

```
59 \input vytexceni     % načtu TeXová makra
60
61 \prumer{Průměr}
62
63 \printxml oktava.xml % zpracuji xml
64 \bye
```

Pokud se příkaz `\prumer` použije vícekrát, bude založeno více nových předmětů. Na onen založený předmět se také pustí fronta podmínek a příkazů, ale až od místa, kde byl příslušný příkaz k zprůměrování. Jednoduché sloučení dvou předmětů by mohlo vypadat následovně:

```
\input vytexceni     % načtu TeXová makra

\prumer{Průměr}
\nazev Český jazyk; Literatura: \prumer{Český jazyk a literatura}

\printxml oktava.xml % zpracuji xml
\bye
```

Toto sloučení má ještě několik nevýhod. Původní předměty zůstaly na svých místech a sloučený předmět je až na konci. Dále v sloučeném předmětu chybí učitel a další textová pole. K tomu slouží klíčová slova uvnitř toho parametru, která nastavují, co se s textovými poli bude dít. Každé klíčové slovo si sebere jako parametr text mezi jím a dalším klíčovým slovem.

Klíčová slova mohou být:

- `\`⟨textové pole⟩. Například `\zkratka Lit` nastaví zkratka na Lit. Parametr příkazu průměr funguje, jako by na začátku bylo klíčové slovo `\nazev`.
- `\spoj`⟨textové pole⟩. Například `\koment \smallskip` pospojuje všechny komentáře, přičemž mezi ně vrazí `\smallskip`
- `\sluc`⟨textové pole⟩. Například `\slucucitel{,␣}` rozdělí jednotlivá textová pole oddělovačem `,␣`, pokud se někteří učitelé opakují, tak je použije jen jednou a pak je opět stejným oddělovaačem spojí. Tedy pokud je učitel u jednoho předmětu „Učitel1, Učitel2“ a u druhého předmětu „Učitel3, Učitel2“, výsledný učitel bude „Učitel1, Učitel2, Učitel3“.
- Údaje o umístění průměru: `\nahore` – umístí průměr nad první zprůměrovaný předmět, `\dole` – umístí průměr pod poslední zprůměrovany předmět, `\nejvyse` – umístí průměr jako první předmět, `\nejnize` (default) – umístí průměr jako poslední předmět.
- `\nehodnot` – implicitně je při počítání průměru předmět s hodnocením nehodnocen(a) přeskakován. Toto klíčové slovo zařídí, že jakmile je jeden z průměrovaných předmětů nehodnocen, bude nehodnocen i průměr.

Takže sloučení předmětů se dá napsat:

```
\prumer{Spojené predmety \spojzkratka + \slucucitel{, }
        \spojkoment\smallskip\nahore\nehodnot} + \smaz
```

Takto vypadala původní koncepce upravování vysvědčení. Jenže po jejím předvedení jsem byl požádán, ať k tomu udělám grafické rozhraní.

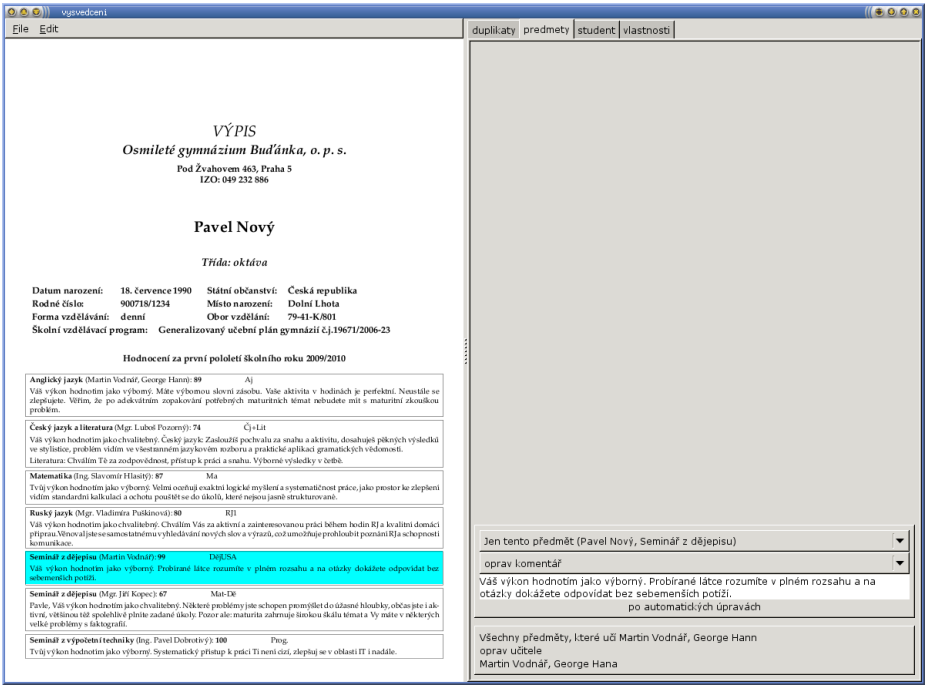
6. Komunikace s GUI aplikací

TEX tedy údaje, která načetl z vysvědčení, zapisuje do dalšího souboru s rozšířením `.dat` a GUI aplikace si tato data přečte. Dalším výstupem z TEXu je PDF, které v sobě obsahuje značky. Konkrétně se jedná o odkazy na url, ale mé GUI tyto odkazy pochopí jako identifikaci předmětu. Když tedy uživatel na předmět klikne, GUI najde ve vedlejším souboru název, učitele, ... a nabídne možnosti úpravy.

Tyto úpravy ovšem nezahrnují průměrování/slučování, takže by se dalo říci, že tyto operace GUI neumí. Ovšem umí zapnout defaultní slučování Češtiny a Literatury spolu s počítáním celkového průměru.

Další předměty, které GUI umí slučovat, jsou předměty se stejným názvem, jelikož takové předměty často (ale ne vždy) mají být sloučeny. TEX tedy GUI aplikaci předá takovéto duplikáty a GUI nabídne zaškrtávací políčka.

Po/během provádění úprav je možné vysvědčení klávesovou zkratkou přeTEXovat a podívat se, jak změny dopadly. GUI přitom změny zanáší do hlavního souboru a využívá tak schopností makra.



VÝPIS

Osmileté gymnázium Buďánka, o. p. s.

Pod Žvahovem 463, Praha 5
IZO: 049 232 886

Pavel Nový

Třída: oktáva

Datum narození:	**18. července 1990**	**Státní občanství:**	**Česká republika**
Rodné číslo:	**900718/1234**	**Místo narození:**	**Dolní Lhota**
Forma vzdělávání:	**denní**	**Obor vzdělání:**	**79-41-K/801**

Školní vzdělávací program: **Generalizovaný učební plán gymnázií č.j.19671/2006-23**

Hodnocení za první pololetí školního roku 2009/2010

Anglický jazyk (Martin Vodnář, George Hann): **89** Aj

Váš výkon hodnotím jako výborný. Máte výbornou slovní zásobu. Vaše aktivita v hodinách je perfektní. Neustále se zlepšujete. Věřím, že po adekvátním zopakování potřebných maturitních témat nebudete mít s maturitní zkouškou problém.

Český jazyk a literatura (Mgr. Luboš Pozorný): **74** Čj+Lit

Váš výkon hodnotím jako chvalitebný. Český jazyk: Zasloužíš pochvalu za snahu a aktivitu, dosahuješ pěkných výsledků ve stylistice, problém vidím ve všestranném jazykovém rozboru a praktické aplikaci gramatických vědomostí.

Literatura: Chválím Tě za zodpovědnost, přístup k práci a snahu. Výborné výsledky v četbě.

Matematika (Ing. Slavomír Hlasitý): **87** Ma

Tvůj výkon hodnotím jako výborný. Velmi oceňuji exaktní logické myšlení a systematičnost práce, jako prostor ke zlepšení vidím standardní kalkulaci a ochotu pouštět se do úkolů, které nejsou jasně strukturované.

Ruský jazyk (Mgr. Vladimíra Puškinová): **80** RJ1

Váš výkon hodnotím jako chvalitebný. Chválím Vás za aktivní a zainteresovanou práci během hodin RJ a kvalitní domácí přípravu. Věnoval jste se samostatnému vyhledávání nových slov a výrazů, což umožňuje prohloubit poznání RJ a schopnosti komunikace.

Seminář z dějepisu (Martin Vodnář): **99** DějUSA

Váš výkon hodnotím jako výborný. Probírané látce rozumíte v plném rozsahu a na otázky dokážete odpovídat bez sebemenších potíží.

Seminář z dějepisu (Mgr. Jiří Kopec): **67** Mat-Dě

Pavle, Váš výkon hodnotím jako chvalitebný. Některé problémy jste schopen promýšlet do úžasné hloubky, občas jste i aktivní, většinou též spolehlivě plníte zadané úkoly. Pozor ale: maturita zahrnuje širokou škálu témat a Vy máte v některých velké problémy s faktografií.

Seminář z výpočetní techniky (Ing. Pavel Dobrotivý): **100** Prog.

Tvůj výkon hodnotím jako výborný. Systematický přístup k práci Ti není cizí, zlepšuj se v oblasti IT i nadále.

Osmileté gymnázium Buďánka, o. p. s.
Pod Žvahovem 463, Praha 5

79-41-K/801 gymnázium – všeobecné

—

Generalizovaný učební plán gymnázií č.j.19671/2006-23

2009/2010 **denní**

oktáva

049 232 886

Pavel Nový
18. července 1990 **900718/1234**
Dolní Lhota **2**
Česká republika

1.

velmi dobré

Anglický jazyk Hodnocení: **89** Vyučující: Martin Vodnář, George Hann
Váš výkon hodnotím jako výborný. Máte výbornou slovní zásobu. Vaše aktivita v hodinách je perfektní. Neustále se zlepšujete. Věřím, že po adekvátním zopakování potřebných maturitních témat nebudete mít s maturitní zkouškou problém.

Český jazyk a literatura Hodnocení: **74** Vyučující: Mgr. Luboš Pozorný
Váš výkon hodnotím jako chvalitebný. Český jazyk: Zasloužíš pochvalu za snahu a aktivitu, dosahuješ pěkných výsledků ve stylistice, problém vidím ve všestranném jazykovém rozboru a praktické aplikaci gramatických vědomostí.

Literatura: Chválím Tě za zodpovědnost, přístup k práci a snahu. Výborné výsledky v četbě.

Matematika Hodnocení: **87** Vyučující: Ing. Slavomír Hlasitý
Tvůj výkon hodnotím jako výborný. Velmi oceňuji exaktní logické myšlení a systematičnost práce, jako prostor ke zlepšení vidím standardní kalkulaci a ochotu pouštět se do úkolů, které nejsou jasně strukturované.

Ruský jazyk Hodnocení: **80** Vyučující: Mgr. Vladimíra Puškinová
Váš výkon hodnotím jako chvalitebný. Chválím Vás za aktivní a zainteresovanou práci během hodin RJ a kvalitní domácí přípravu. Věnoval jste se samostatnému vyhledávání nových slov a výrazů, což umožňuje prohloubit poznání RJ a schopnosti komunikace.

Seminář z dějepisu Hodnocení: **99** Vyučující: Martin Vodnář
Váš výkon hodnotím jako výborný. Probírané látce rozumíte v plném rozsahu a na otázky dokážete odpovídat bez sebemenších potíží.

Seminář z dějepisu Hodnocení: **67** Vyučující: Mgr. Jiří Kopec
Pavle, Váš výkon hodnotím jako chvalitebný. Některé problémy jste schopen promýšlet do úžasné hloubky, občas jste i aktivní, většinou též spolehlivě plníte zadané úkoly. Pozor ale: maturita zahrnuje širokou škálu témat a Vy máte v některých velké problémy s faktografií.

Seminář z výpočetní techniky Hodnocení: **100** Vyučující: Ing. Pavel Dobrotivý
Tvůj výkon hodnotím jako výborný. Systematický přístup k práci Ti není cizí, zlepšuj se v oblasti IT i nadále.

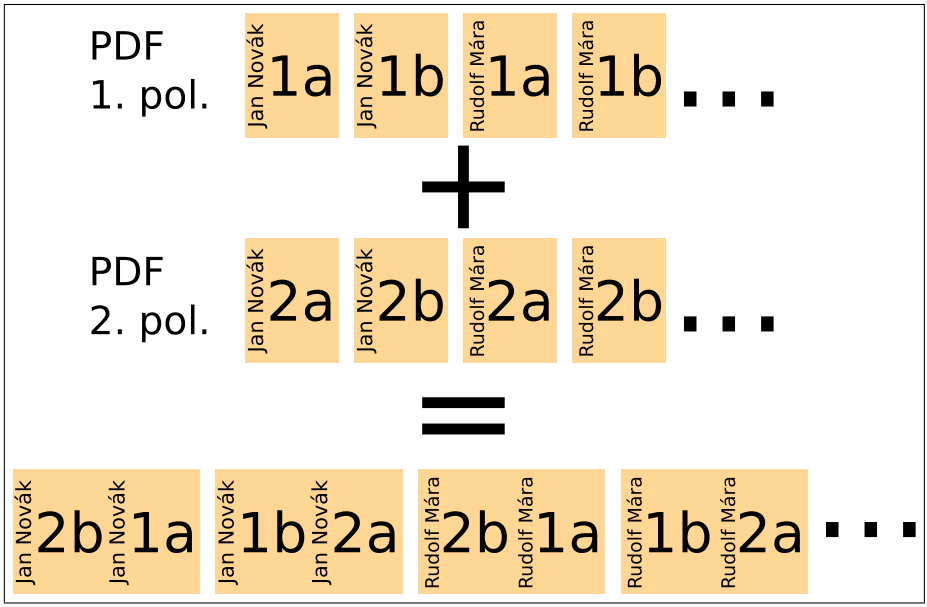
Seminář ze základů spol. věd Hodnocení: **88** Vyučující: Jene Okama
Tvůj výkon hodnotím jako výborný. Na semináři je radost s Tebou pracovat. Máš bystrý úsudek, dokážeš udržet linii problému i správně určit jádro myšlenky. Tvůj projev je kultivovaný, dobře strukturovaný a duchaplný. Na středoškolskou úroveň jsou Tvé schopnosti výjimečné. Jsem přesvědčen, že je v budoucnu dokážeš mimořádně zdokonalit.

Seminář ze zeměpisu Hodnocení: **52** Vyučující: Marcel Skočdopole
Tvůj výkon hodnotím jako dobrý. Oceňuji Tvůj zájem, sporadickou aktivitu v hodinách. Znalosti máš ovšem neuspořádané. Jako úkol do budoucna vidím větší aktivitu v hodinách a lepší systematičnost přípravy.

7. Oficiální formulář

Vysvědčení má na Buďánce dva typy (druhy designu). Vysvědčení může být tzv. Buďánkovské a oficiální. Defaultní je Buďánkovský design, přičemž oficiální je možné zapnout definováním řídící sekvence `\ofidesign`. Ten je připravený pro tisk na předpřipravený formulář, takže na rozdíl od Buďánkovského vysvědčení nemá popisky a makro se musí s údaji strefit do míst podle formuláře.

Oficiální formulář má ovšem formát A3 přeložený tak, že vznikají 4 A4 stránky. Dvě stránky obsahují údaje z prvního pololetí a dvě z druhého. Makro, které čte XML, vygeneruje vždy jen jedno pololetí, takže je potřeba výsledná PDF z obou pololetí spojit podle následujícího schématu:



Spojení probíhá pomocí pdfTEXového makra, které čte PDF soubory z jednotlivých pololetí. Výsledné spojené PDF obsahuje stránky uspořádány tak, že je možné je přímo tisknout na duplexové A3 tiskárně.

Při spojování makro počítá s tím, že studenti nemusí být ve vstupních PDF souborech ve stejném pořadí. Proto makro pracuje s pracovními dat soubory, kde jsou informace o pořadí studentů uloženy.

Snadnou úpravou tohoto makra je navíc možné dělat se stránkami ještě jisté posuny pro doladění pozice na formuláři.

Miroslav Olšák
`mirek@olsak.net`

ScanCSV – Lua knihovna pro zpracování CSV souborů ConTEXtem a LuaLATEXem

Jaroslav Hajtmar

Tento článek popisuje možnosti použití jazyka Lua pro vytvoření knihovny Lua funkcí, které mohou zajímavým způsobem zpřístupnit ConTEXtu, LuaTEXu a LuaLATEXu textové databázové údaje, uložené v CSV souborech. Prioritou při tvorbě popisované luaknihovny bylo, aby mohla být používána i uživateli bez sebemenších znalostí jazyka Lua. Kdo zvažuje, že si „něco začne“ s Lua, má příležitost zjistit, jak Lua funguje. Kdo chce zůstat „ryzím TEXistou“, má možnost používat popisovanou knihovnu formou „blackboxu“, tj. do zdrojového textu ConTEXtu (PlainTEXu, LATEXu) zapsat několik řádků Lua kódu, zkompilovat zdroják odpovídajícím formátem a koukat, jak to celé krásně funguje. Pro vážnější zájemce jsem připravil ke stažení řadu pokročilých ukázek, demonstrujících zajímavé možnosti praktického použití knihovny. Ačkoliv je Lua knihovna primárně určena pro použití v ConTEXtu, přichystal jsem řadu ukázek použití v LATEXu, který bude zřejmě čtenáři tohoto článku preferován.

1. Úvod

V tomto příspěvku se pokusím popsat svůj začátečnický pokus o vytvoření luaknihovny, s jejíž pomocí lze snadným způsobem zpřístupnit ConTEXtu, LuaTEXu (Plainu) a LuaLATEXu textové databázové údaje, uložené v CSV souborech.

ConTEXtem budu v celém příspěvku mínit verzi MKIV, která je postavena na jazyce Lua (narozdíl od původní verze MKII). Aplikace uvedených Lua funkcí umožňuje snadné vytváření dokumentů, obsahujících hromadná data z jednoduchých CSV databází, která jsou vhodným způsobem naformátována. Použití je velmi pestré, např. tisk hromadných dopisů, různých formulářů, vysvědčení, pozvánek, průkazek, vizitek, kartiček atd.

Při programování knihovny byl kladen velký důraz na to, aby funkce byly široce použitelné nejen v ConTEXtu (pro který byly primárně napsány), ale i v LuaTEXu a LuaLATEXu. Hlavním cílem bylo také to, aby uživatel (Plainista, LATEXista či ConTEXtista) těchto Lua funkcí nemusel mít vůbec žádné znalosti jazyka Lua, a přitom mohl velmi dobře prakticky tyto funkce používat. Ve skutečnosti se ve zdrojovém TEXovém kódu objevuje volání Lua kódu jen na začátku dokumentu, samotné zpřístupnění CSV dat je realizováno prostřednictvím TEXových maker.

 doi: 10.5300/2012-2/76

2. CSV formát

Většina čtenářů jistě zná formát CSV (comma-separated values, hodnoty oddělené čárkami). Jde se o velmi jednoduchý souborový formát určený pro výměnu tabulkových dat. Obecně CSV soubor sestává z řádků (záznamů), v nichž jsou jednotlivé položky (pole) odděleny oddělovačem (většinou čárkou nebo středníkem). Hodnoty takto oddělených položek v řádku mohou být navíc vymezeny dalšími znaky (standardně jsou uzavírány do uvozovek). To umožňuje, aby text položek obsahoval i oddělovač polí (tedy např. čárku či středník). Standardně CSV formát umožňuje vymezovačem (např. uvozovkami) vymezit pouze ty položky, které obsahují oddělovač položek, navíc umožňuje i to, aby jednotlivé položky obsahovaly i znak vymezovače (tj. uvozovky) atd. Jednoduchý algoritmus, který jsem navrhnul pro rozparsování položek ovšem předpokládá, že pokud již začneme ve svých CSV souborech používat vymezovač, musí jím být vymezeny úplně všechny položky. Navíc nesmí být znak pro vymezovač obsažen uvnitř žádné ze sloupcových položek. Parsovací funkci `ParseCSVdata` lze samozřejmě zobecnit.

Za hojné používání CSV souborů se středníkem (byť pod názvem CSV) vděčí tento formát Microsoft Excelu v české verzi Microsoft Windows. Standardní oddělovač (středník) lze změnit (v místním a jazykovém nastavení).

Ukázky CSV tabulek na něž se budu následně v textu i zdrojovém kódu ukázkového dokumentu odvolávat:

Příklad 1 (zaci.csv) *CSV tabulka bez záhlaví. Položky oddělené středníkem, vymezovač není:*

```
1;Petr;Novák;19.5.1989;m;Nymburk;U Brány 7
2;Jan;Novotný;5.7.1991;m;Praha;Uhlířská 178
...
```

Příklad 2 (studenti.csv) *CSV tabulka se záhlavím. Položky oddělené středníkem, vymezovač není:*

```
Prijmeni;Jmeno;DatumNarozeni;Pohlavi;Mesto;PSC;Ulice
Novák;Jan;14.10.1997;m;Zbečno;27024;Farní 21
Pospíšilová;Hana;4.1.1996;ž;Zábřeh;78901;Studénky 420
...
```

Příklad 3 (zamestnanci.csv) *CSV tabulka se záhlavím. Oddělovačem položek je tentokrát čárka, vymezovačem jsou uvozovky, uvnitř nichž se může vyskytovat separátor polí (čárka):*

```
"PorCis","Prijmeni","Jmeno","Pohlavi","DatumNarozeni","Bydliste"
"1","Májková","Barbora","ž","2.4.1995","Mírov 96, 789 53 Mírov"
"2","Nová","Zuzana","ž","13.1.1995","U Dráhy 8, 789 01 Zábřeh"
...
```

3. Použití CSV tabulek v TEXu

V r. 2005 jsem náhodou objevil makro `scanbase.tex` Petra Olšáka, které mne velmi zaujalo svou praktičností. Petr Olšák jej následně zobecnil a zmodifikoval tak, aby se dalo použít i pro zpracování CSV souborů nejen v Plainu, ale i v LATEXu (modifikaci v r. 2008 navrhl Jaromir Kuben) a ConTEXtu MKII. Pro svoji práci v Plainu jsem používal makro `scancsv.tex` hezkou řadu let, makro `scancsv-context.tex` občas použiju v ConTEXtu MKII i dnes.

Původní makro Petra Olšáka mi bylo tak zásadní inspirací pro tvorbu mých Lua funkcí, že jsem ve své Lua aplikaci ponechal i stejné názvy hlavních TEXových maker. Důležitým důvodem ovšem bylo také to, že jsem chtěl, aby se mi snáze předělávaly vlastní starší aplikace, založené na použití původního Olšákova makra. V původním zdrojovém textu pak stačilo udělat jen pár drobných úprav a zkompilovat jej ConTEXtem MKIV. Jistou nepříjemností při používání CSV souborů v ConTEXtu (LuaTEXu, LuaLATEXu) je, že pro správné fungování aplikací je požadováno UTF-8 kódování, zatímco CSV tabulky vytvořené v Microsoft Excelu jsou defaultně kódované CP 1250. Překódování naštěstí snadno zvládne každý solidní textový editor.

4. Knihovna `scancsv.lua`

4.1. Princip fungování luaknihovny

Lua knihovna `scancsv.lua` je tvořena sérií funkcí, které zpřístupňují data v CSV tabulkách (kódovaných UTF-8). Ačkoliv se budu neustále odvolávat na použití knihovny `scancsv.lua` v ConTEXtu MKIV, musím poznamenat, že funkce byly navrženy vesměs tak, aby fungovaly i v LuaTEXu či LuaLATEXu (bylo testováno pouze na jednoduchých příkladech).

Princip fungování knihovny vychází z již zmiňovaného TEXového algoritmu Petra Olšáka. Postupně jsou data obsažená v řádcích CSV tabulky prostřednictvím programových Lua cyklů rozparsována a zapracována TEXových maker. Makra jsou v průběhu zpracování automaticky nadefinována (vygenerována) a následně mohou být použita ve zdrojovém textu ConTEXtu (LuaLATEXu či LuaTEXu). První verze knihovny vyžadovala jisté znalosti Lua a určitý způsob umísťování Lua kódu do zdrojového textu. Vzhledem k tomu, že hlavní prioritou bylo, aby mohl knihovnu používat kterýkoliv TEXista, aniž by měl sebemenší znalosti Lua, doznala knihovna nakonec takových změn, že ve zdrojovém dokumentu je umístěn Lua kód pouze na začátku dokumentu [1]. Kód slouží k načtení luaknihovny, což bude předvedeno v ukázkových příkladech pro ConTEXt a LuaLATEX.

[1] Výjimkou může být volání knihovní funkce `filelineaction(csvfile,fromrow,torow)` v LuaLATEXu, která zajistí, že se načte určitý rozsah řádků CSV tabulky (`fromrow` – `torow`). Pro zpracování celého CSV souboru je i v LuaLATEX u připraveno samostatné TEXové makro.

Vzhled výsledného dokumentu (tiskové sestavy) je předurčen libovolně nadefinovaným TEXovým makrem `\lineaction`. V tomto makru mohou být samozřejmě použita všechna makra automaticky vygenerovaná funkcemi knihovny. Sestava může mít libovolný vzhled, nejčastěji tabulkový nebo formulářový. Po ukončení makra `\lineaction` se v cyklu automaticky načte další řádek CSV tabulky, znovu se naplní všechna vygenerovaná makra texty sloupcových položek z tohoto řádku a znovu se spustí `\lineaction`. To se opakuje tak dlouho, dokud není zpracována celá tabulka nebo její část (určená dvojicí parametrů `fromrow` a `torow`). Knihovna makro `\lineaction` implicitně nadefinuje, takže pokud uživatel vlastní definicí toto makro nevytvoří, použije se implicitní nastavení, které udělá to, že se vypíší všechny řádky CSV tabulky v jakémsi „zhuštěném“ formátu.

Já většinou nadefinuji makro `\printaction` (ve kterém se používají automaticky vygenerovaná makra), které určuje vzhled tiskové sestavy a toto makro následně vhodným způsobem zakomponuji do makra `\lineaction`. Volitelně si může uživatel nadefinovat také makra `\blinehook` (háčkové makro, které se vždy provede automaticky před provedením makra `\lineaction` tj. před zpracováním každého řádku CSV tabulky) a `\elinehook` (provede se vždy po `\lineaction`) a dále makra `\bfilehook` a `\efilehook`, což jsou makra, která se provedou před začátkem zpracování a na konci zpracování celého CSV souboru. Pomocí těchto „háčků“ (které jsou implicitně knihovnou nastaveny na `\relax`) lze nastavovat různá záhlaví a zápatí skupin, odstránkování atd. Rád bych upozornil na to, že nic nebrání zpracování i několika různých CSV souborů v jednom dokumentu atd.

4.2. TEXová makra pro zpřístupnění sloupcových dat CSV tabulky

Nejdůležitějšími makry pro použití jsou makra, která zpřístupňují sloupcová data v CSV tabulce. Názvy těchto maker jsou odvozeny od názvů sloupců v excelovské tabulce. Data ve sloupcích A, B, C, atd. jsou k dispozici v makerch s názvy `\cA`, `\cB`, `\cC`, atd.

K vytváření názvů sloupců je potřebná funkce `ar2xls(number)` (Arabic to XLS). Ta převádí pořadové číslo sloupce na název excelovského sloupce. Nepředpokládá se, že by někdo potřeboval více než 703 sloupců (tj. sloupce nad rozsah A – ZZ). Funkci `ar2xls` by bylo nutné v tom případě upravit.

Pokud by někomu místo excelovského formátu názvů sloupců spíše vyhovovaly názvy, založené na římských číslech, tj. `\cI`, `\cII`, `\cIII`, `\cIV`, atd., lze to snadno zařídit nastavením Lua proměnné `UserColumnNumbering` na jinou hodnotu než `XLS` (defaultní hodnota). V tomto případě se použije pro vytváření názvů sloupců funkce `ar2rom(number)`. Počet sloupců v CSV tabulce není nijak limitován (některé verze Excelu však mohou např. obsahovat maximálně 256 sloupců). Pro CSV tabulky s větším počtem sloupců může být ovšem používání excelovského či římského označování sloupců poněkud nepřehledné. Knihovna proto vygene-

ruje i TEXová makra se stejnými názvy, jako jsou názvy polí v prvním řádku CSV tabulky (záhlaví). Uživatel tak nemusí pracně „odpočítávat“ a identifikovat jednotlivé sloupce, postačí mu znát název sloupce a makro tohoto jména mu zpřístupní potřebná data. Je na místě dodat, že tato makra se vygenerují dokonce i v případě, že první řádek je „datový“ (tj. není tzv. záhlavím), přičemž tato volba je nastavena defaultně. Názvy maker v tomto případě budou nejspíš nesrozumitelné, neboť „nevhodné“ názvy sloupců jsou (z hlediska striktních požadavků na názvy TEXových maker) vždy ošetřovány funkcí `TMN(string)` (TEXMacro Name). Vstupním parametrem této funkce je textový řetězec ze záhlaví daného sloupce CSV tabulky. V tomto řetězci se nahradí všechny diakritické znaky české abecedy (pro jiné jazyky je nutná modifikace makra) znaky bez diakritiky, arabské číslice se nahradí římskými, nealfabetické znaky se se nahradí zástupným znakem `"x"`. Nakonec se zkrátí délka makra na 20 znaků (lze zvýšit). Výstupem funkce je název makra, který může být pro uživatele nesrozumitelný, ale TEXu by neměl vadit. Pokud tedy zamýšlíte využívat toho, že knihovna vygeneruje „pěkné“ názvy maker ze záhlaví CSV tabulky, měli byste se při vytváření záhlaví vyvarovat používání všech znaků, které se nemohou vyskytovat v názvech TEXových maker. Nutno dodat, že nastavením Lua proměnné `CSVHeader` na hodnotu `true` docílíme toho, že knihovna první řádek CSV tabulky nepovažuje za datový řádek. Data z tohoto řádku pak knihovna nezpracovává (nevypisuje je). Tento řádek se pak ani nezapočítává do počtu zpracovávaných řádků tabulky. Defaultní hodnota Lua proměnné `CSVHeader` je ovšem `false`, což nebrání knihovně vytvořit z prvního řádku názvy sloupcových maker. Všechny řádky jsou ovšem brány automaticky jako datové a zpracovávají se.

Pokud použijeme ke zpracování CSV soubor, uvedený v příkladu 2, budou sloupcová data zpřístupněna jednak v makrech `\cA`, `\cB`, `\cC`, ..., `\cH`, ale také v makrech `\Prijmeni`, `\Jmeno`, `\DatumNarozeni` atd., zatímco pokud použijeme ke zpracování CSV soubor, uvedený v příkladu 1, budou sloupcová data zpřístupněna v makrech `\I=\cA` (číslice 1 je nahrazena římskou číslicí), `\Petr=\cB`, `\Novak=\cC` (písmeno „á“ nahrazeno písmenem „a“), podivně vyhlížející makro `\IIXxVxIIXVIIIIIX=\cD` vzniklo nahrazením arabských cifer a teček. [2]

Všechna výše uvedená TEXová makra vznikají v průběhu cyklu zcela automaticky bez jakéhokoliv přispění uživatele, uživatel je dokonce nemusí ani v ConTEXtu definovat. Tím nastává paradoxní situace, neboť se ve zdrojovém ConTEXtovém textu se mohou objevovat např. makra `\Prijmeni`, `\Jmeno`, `\DatumNarozeni` atd., která ale nejsou nikde výše ve zdrojáku nadefinovaná.

Pokud má někdo strach, že si automaticky generované názvy TEXových maker nezapamatuje, mohu jej uklidnit. Pomocí Lua funkce `CSVReport()` resp. pomocí

[2] Sloupcové údaje jsou v tomto režimu k dispozici i prostřednictvím globálních Lua proměnných CSV.Prijmeni, CSV.Jmeno, CSV.DatumNarozeni atd. Jejich použití však již vyžaduje jisté znalosti Lua.

TEXového makra `\csvreport` si může uživatel kdykoliv ve zdrojovém textu ConTEXtu či LuaLATEXu nechat vypsat nejen řadu užitečných informací o otevřené CSV tabulce, ale mj. i názvy všech vygenerovaných maker.

Vzhledem k tomu, že jsem předpokládal, že většina CSV tabulek nebude obsahovat záhlaví s názvy sloupců, knihovna „předpokládá“, že záhlaví u tabulek nejsou. Implicitní hodnoda Lua proměnné `CSVHeader` je v knihovně určena hodnotou proměnné `UserCSVHeader` (defaultně `false`), což lze v knihovně vyeditovat. Pokud by měl někdo zájem používat snadno zapamatovatelná makra (označující sloupcové položky) a přitom nemít záhlaví CSV tabulek, může to zařídit tak, že si na začátku definice makra `\lineaction` (resp. `\printaction`) pomocí `\let` vytvoří potřebná jména, a jim přiřadí hodnoty standartních excelovských (či římských) názvů maker (např. `\let\PorCislo\cA`). Způsob nastavení vlastních názvů bude později předveden v ukázkovém příkladu.

4.3. TEXová makra pro získání „systémových“ informací

Knihovna vygeneruje automaticky i další TEXová makra, která může uživatel využít pro tvorbu tiskových sestav nebo pro vypsání důležitých informací o zpracovávané CSV tabulce:

`\csvfilename` : Jméno aktuálně otevřeného CSV souboru.

`\numcols` : Počet sloupců CSV tabulky.

`\numrows` : Počet aktuálně zpracovaných (vypsaných) řádků. Může se lišit od celkového počtu řádků tabulky (pokud uživatel vypisuje pouze určitý rozsah řádků).

`\numline` : Pořadové číslo aktuálně načteného řádku. Vhodné pro použití v tiskových sestavách.

`\csvreport` : Reportní informace o otevřeném CSV souboru (mj. názvy všech upotřebitelných TEXových maker atd.).

`\printline` : Aktuální řádek CSV tabulky ve „zhuštěném“ tvaru.

`\printall` : CSV tabulka ve „zhuštěném“ tvaru.

4.4. Modifikace funkčnosti knihovny

Uživatel si může knihovnu do značné míry přizpůsobit svým potřebám. Prostřednictvím několika globálních proměnných lze nastavit řadu defaultních vlastností. Proměnné mohou být přenastaveny buď přímo ve vlastním souboru knihovny, což umožní uživateli „jednou provždy“ provést nastavení vyhovující jeho standartním potřebám (např. oddělovačem položek v používaných CSV tabulkách nebude středník ale čárka). Nastavení hodnot příslušných proměnných lze provést po načtení knihovny i ve zdrojovém TEXovém souboru.

Přímo v knihovně lze nastavit nastavit defaultní hodnoty těchto proměnných:

`UserCSVSeparator` : Používaný separátor polí. Defaultní je středník.

`UserCSVLeftDelimiter` : Pole mohou být vymezena vymezovačem. To umožní používat znak pro separátor polí i uvnitř textu pole. Defaultní je prázdný řetězec `UserCSVLeftDelimiter=''`.

`UserCSVRightDelimiter` : Lze nastavit i pravý vymezovač. Defaultní hodnota je stejná jako v předchozím případě.

`UserCSVHeader` : Načtený CSV soubor je defaultně posuzován jako bez hlavičky (údaje v hlavičce jsou brány jako názvy sloupcových maker). Defaultně je `UserCSVHeader=false`.

`UserColumnNumbering` : Něco jiného než XLS nebo nedefinovaná hodnota (např. zakomentováním řádku) nastaví římské číslování. Defaultní je `XLS`.

Ve zdrojovém textu lze „přebít“ defaultní hodnoty výše uvedených proměnných (v případě, že mimořádně zpracováváme jiný typ CSV souboru než je běžné) a změnit tak chování knihovny prostřednictvím nastavení těchto proměnných:

`File2Scan` : Jméno používaného CSV souboru. Proměnná se může nastavit manuálně nebo se nastaví automaticky při volání funkce, jejímž parametrem je název souboru. Její nastavení umožňuje volat funkce `OpenCSVFile()`, `CSVReport()`, `lineaction()` a `filelineaction()` bez parametrů. Nastavit proměnnou lze i makrem (např. `\setfiletoscan{zaci.csv}`).

`Sep` : Oddělovač sloupců v CSV souboru. Defaultní hodnota separátoru je v proměnné `UserCSVSeparator`. Některé znaky bohužel nelze použít jako separátor. Použít lze rovněž makro (např. `\setsep{,}`).

`Ld` : Left delimiter. Sloupcové položky mohou být ohraničeny jinak zleva a jinak zprava. Defaultní hodnota je v proměnné `UserCSVLeftDelimiter`. Některé znaky nelze použít. K dispozici je makro (např. `\setld{"}`).

`Rd` : Right delimiter. Defaultní hodnota je v proměnné `UserCSVRightDelimiter`. Nejčastěji bývají jednotlivé položky ohraničeny znakem `"` (tj. vloženy do uvozovek). Makrem např. `\setrd{"}`.

`CSVHeader` : Hodnota `true`, znamená, že 1. řádek obsahuje záhlaví se jmény sloupců (tj. není datový). Defaultní hodnota `false` – všechny řádky jsou uvažovány jako datové. Lze užít TEXové makro `\setheader`.

Další možnosti nastavení zvídavý čtenář vyčte přímo z knihovního souboru `scancsv.lua`. Ten obsahuje vysvětlující komentáře, z nichž lze vyčíst princip fungování.

4.5. Praktické ukázky užití knihovny

Po dohodě s redakcí jsme ustoupili od zveřejnění zdrojového textu knihovny. Zdrojový text obsahující množství komentářů si bude moci čtenář spolu s řadou funkčních příkladů stáhnout z úložiště a následně prostudovat, vyzkoušet a otestovat. V tuto chvíli pojďme ukázat pouze výpis krátkého programu. Z něj snad bude patrné, jak knihovna funguje.

V následujícím zdrojovém textu budeme předpokládat, že používáme při práci UTF-8 kódované CSV soubory `zaci.csv`, `studenti.csv` a `zamestnanci.csv` z příkladů 1, 2 a 3. Zdrojový text příkladu je pro pochopení fungování dostatečně okomentován. Ukázka výstupu následujícího příkladu je na obr. 1 a obr. 2. Na nich jsou předvedeny poslední dvě stránky výsledného PDF souboru.

```
% Ukázka použití. Kompilace: lualatex examples-latex.tex
\documentclass{article}
\usepackage{luatextra}
\usepackage[utf8]{luainputenc}
\parindent0pt

\begin{document}
\pagestyle{empty}

% Načtení knihovny
\directlua{dofile("../scancsv-general/scancsv.lua")}

% Volitelné definice "háčků" (defaultně jsou háčky "zarelaxovány")
% Hooks před a po zpracování celé tabulky:
\def\bfilehook{Seznam účastníků zájezdu {\bf\csvfilename}:
  \par\bigskip}
\def\efilehook{\par Sestavou bylo vypsáno {\bf\numrows} účastníků.
\bigskip}

% Hooks před a po zpracování řádkové informace:
\def\blinehook{Účastník č. \numline :\par} % akce před výpisem ř.
\def\elinehook{\par\smallskip\hrulefill\medskip\par} % po výpisu ř.

% Definice pomocných vyhodnocujících maker použitých v sestavách
\def\priponaa{\if m\pohl\else a\fi}
\def\priponai{\if m\pohl\else í\fi}

% Definice tiskové sestavy:
\def\printaction{
pan\priponai\ \Jmeno\ \Prijmeni\par
narozen\priponaa\ \DatumNarozeni\par
bytem \bydliste
}

% Pro zpracování nyní stačí definice makra \lineaction např. takto:
% \def\lineaction{\printaction}
```

```
% Vzhledem k ukázkovému zpracování více CSV souborů příklad
% poněkud zobecníme (byť za cenu menší přehlednosti)

% Vlastní tiskové makro pro soubor zaci.csv si připravíme např. takto:
% Uvědomme si, že tento CSV soubor je "bez hlavičky" tj. není známa
% informace o názvech sloupců

\def\lineaction{
  % Kvůli přehlednosti si můžeme nastavit potřebná makra
\let\Jmeno\cB % ve 2. sloupci tj. excelovsky sloupec B
\let\Prijmeni\cC
\let\DatumNarozeni\cD
\let\pohl\cE% pro fungování pomocných maker \priponaa a \priponai
\def\bydliste{\cG, \cF} % můžeme i definovat nová makra
% Cílem těchto "tanečků" je docílit toho, aby byla všechna makra
% používaná v tiskové sestavě \printaction nadefinována dříve, než
\printaction  % zavoláme.
}

% Zpracování sestavy pro tabulku zaci.csv zařídí Lua tento kód:
%\directlua{filelineaction("zaci.csv")}

\filelineaction{zaci.csv} % nebo příslušné TeXové makro

% Závěrečná ukázka použití některých "systémových" maker:

Poslední zpracovaný řádek tabulky \csvfilename:\par\smallskip

\printline\bigskip

CSV tabulka \csvfilename\ ve "zhuštěném tvaru":\par\smallskip

\printall\pagebreak

Výpis reportních informací z CSV souboru \csvfilename:\par\bigskip

\csvreport

% Ukázka toho, jak naráz zpracovat více různých CSV souborů:

% Tiskové makro pro soubor studenti.csv si připravíme jinak.
```

```
% Tabulka má záhlaví s názvy sloupců. Lze se tak odvolávat na názvy
% sloupcových maker, případně si vytvořit další potřebná makra

\def\lineaction{
\let\pohl\Pohlavi% fungování pomocných maker \priponaa a \priponai
\def\bydliste{\Ulice, \PSC\ \Mesto} % bydliště trochu jinak
\printaction % standartní tisková sestava (lze užít i jinou)
}

% Vlastní zpracování sestavy pro tabulku studenti.csv
\setheader % nastaví CSVHeader=true - příznak existence záhlaví
% \filelineaction{studenti.csv} % V LaTeXu zpracuje jen celý soubor
% nebo pomocí Lua i souvislé skupiny řádků
% filelineaction("studenti.csv",3,5) -- vypíše jen od ř.3 do ř.5

\directlua{filelineaction("studenti.csv",3,5)}
% \directlua{filelineaction("studenti.csv",8,17)} % další řádky

Poslední zpracovaný řádek tabulky \csvfilename:\par\smallskip

\printline\bigskip

CSV tabulka \csvfilename\ ve "zhuštěném tvaru":\par\smallskip

\printall\pagebreak

Výpis reportních informací z CSV souboru \csvfilename:\par\bigskip

\csvreport\pagebreak

% Zveřejněná ukázka výstupu programu bude provedena od tohoto místa:

% Ukázka toho, že lze také zpracovat i CSV soubor s jiným formátem:
% Příprava tiskového makra pro soubor zamestnanci.csv
% Tabulka má také záhlaví s názvy sloupců, takže postačí jen:
\def\lineaction{
\let\pohl\Pohlavi% Pro pomocná makra
\def\bydliste{\Bydliste} % jinak - dle CSV tabulky zamestnanci.csv
\printaction % standartní tisková sestava (lze ale použít i jinou)
}

% Vlastní zpracování sestavy pro tabulku zamestnanci.csv
```

```
% Předem nastavíme potřebné proměnné modifikující chování knihovny
\setheader % tj. soubor má hlavičku s názvy polí
\setsep{,} % oddělovač polí
\setld{"} % levý vymezovač
\setrd{"} % pravý vymezovač
\filelineaction{zamestnanci.csv}

Poslední zpracovaný řádek tabulky \csvfilename:\par\smallskip

\printline\bigskip

CSV tabulka \csvfilename\ ve "zhuštěném tvaru":\par\smallskip

\printall\pagebreak

Výpis reportních informací z CSV souboru \csvfilename:\par\bigskip

\csvreport

\end{document}
```

5. Netriviální použití luaknihovny

Jednoduchá ukázka nemůže plně vystihnout možnosti popisované luaknihovny, proto jsem pro zájemce připravil i netriviální ukázky jejího praktického použití. Na několika příkladech si můžete systém vyzkoušet a nechat se inspirovat možnostmi. Na mém datovém úložišti jsou uloženy dokumenty, které byly vytvořeny pro účely administrativní agendy ConTEXtmeetingu 2010 a TEXperience 2010. Jedná se o seznamy účastníků, identifikačních kartiček, časových rozvrhů a programů konferencí (viz ukázka na obr. 3 a obr. 4). Připravena je i ukázka současného zpracování tabulek `ucitele.csv` a `zaci.csv` (s fiktivními údaji), které jsou během zpracování propojeny přes název třídy pomocí relace `1:N`. Výsledkem je dokument se seznamy tříd s jejich třídními učiteli. Všechny ukázky jsou připraveny v mém oblíbeném ConTEXtu, stejně tak jako ukázka tisku maturitních protokolů (s fiktivními údaji), které byly na našem gymnáziu několik let prakticky využívány. Aby nepřišli zkrátka ani zájemci z řad LATEXistů, přiložil jsem dvě ukázky pro tisk obálek a faktur. Modifikací zdrojových souborů a jejich následnou kompilací můžete proniknout do možností luaknihovny. Jednotlivé zdroje lze kompilovat aktuálními verzemi LuaLATEXu a ConTEXtu.

Všechny dokumenty potřebné pro vyzkoušení fungování knihovny naleznete na adrese `http://cstugbull2012.hajtmar.com`.

Seznam účastníků zájezdu **zamestnanci.csv**:

Účastník č. 1:
paní Barbora Májková
narozena 2.4.1995
bytem Mírov 96, 789 53 Mírov

Účastník č. 2:
paní Zuzana Nová
narozena 13.1.1995
bytem U Dráhy 8, 789 01 Zábřeh

Účastník č. 3:
pan Petr Dyk
narozen 10.4.1995
bytem Tůmova 207, 789 01 Zábřeh

Účastník č. 4:
paní Anežka Ferová
narozena 1.1.1995
bytem Revoluční 56, 787 01 Šumperk

Účastník č. 5:
pan Jan Neděla
narozen 2.4.1995
bytem Bratrušovská 31, 787 01 Šumperk

Sestavou bylo vypsáno **5** účastníků.

Poslední zpracovaný řádek tabulky zamestnanci.csv:

6,Neděla,Jan,m,2.4.1995,Bratrušovská 31, 787 01 Šumperk,

CSV tabulka zamestnanci.csv ve "zhuštěném tvaru":

1, Májková, Barbora, ž, 2.4.1995, Mírov 96, 789 53 Mírov,
2, Nová, Zuzana, ž, 13.1.1995, U Dráhy 8, 789 01 Zábřeh,
3, Dyk, Petr, m, 10.4.1995, Tůmova 207, 789 01 Zábřeh,
4, Ferová, Anežka, ž, 1.1.1995, Revoluční 56, 787 01 Šumperk,
6, Neděla, Jan, m, 2.4.1995, Bratrušovská 31, 787 01 Šumperk,

Obr. 1 Ukázka výstupu vygenerovaného triviálním ukázkovým příkladem. Zdrojový text příkladu je na předchozích stránkách.

Výpis reportních informací z CSV souboru zamestnanci.csv:

Informace o používaném CSV souboru
Vstupní CSV soubor: **zamestnanci.csv**
Vymezovače a oddělovače sloupců viz. Lua proměnné `Sep`, `Ld` a `Rd`
Nastavení vymezovačů a oddělovačů sloupců: `"pole1","pole2","pole3",` …
Počet sloupců v tabulce: **6**
Počet řádků v tabulce : **5**
Makra zpřístupňující řádková data v tabulce:
\cA=\PorCis, **\cB=\Prijmeni**, **\cC=\Jmeno**, **\cD=\Pohlavi**, **\cE=\DatumNarozeni**, **\cF=\Bydliste**,
Další předdefinovaná makra:
\csvfilename – otevřený soubor s CSV tabulkou (**zamestnanci.csv**)
\numcols – počet sloupců tabulky (**6**)
\numrows – počet aktuálně zpracovaných (vypsaných) řádků (**5**)
\numline – číslo aktuálně načteného řádku (pro použití v tiskových sestavách)
\csvreport – vypíše se report o otevřeném souboru
\printline – vypíše aktuální řádek CSV tabulky ve zhuštěném tvaru
\printall – vypíše CSV tabulku ve zhuštěném tvaru
\setfiletoscan{*filename*} – nastaví jméno zpracovávaného CSV souboru
\opencsvfile{*filename*} – otevření CSV souboru
\openheadercsvfile{*filename*} – otevření CSV souboru s hlavičkou
\setheader – nastavení příznaku existence hlavičky
\resetheader – nastavení příznaku neexistence hlavičky
\readrow – načtení řádku (+ zůstane na něm)
\nextrow – načtení řádku (+ skok na další)
\setsep{*separator*} – nastavení oddělovače polí na příslušný znak
\resetsep – nastavení oddělovače polí na defaultní hodnotu
\setld{*delimiter*} – nastavení levého vymezovacího znaku
\resetld – nastavení levého vymezovače na defaultní hodnotu
\setrd{*delimiter*} – nastavení pravého vymezovacího znaku
\resetrd – nastavení pravého vymezovače na defaultní hodnotu
\blinehook – nastavení háčku (vykonaného před načtením řádku)
\elinehook – nastavení háčku (vykonaného po zpracování řádku)
\bfilehook – nastavení háčku (vykonaného před načtením CSV souboru)
\efilehook – nastavení háčku (vykonaného po zpracování CSV souboru)

Obr. 2 Výpis reportních informací spolu se seznamem maker, které má uživatel knihovny `scancsv.lua` k dispozici.

Preliminary list of participants of ConTEXt meeting and TEXperience
Brejlov 13. – 19. 9. 2010

(to date 6. 9. 2010)

Id	Name	Nationality	Monday – Thursday	Thursday p.m.	Friday	Saturday	Sunday	E-mail
1	Bašić Nino		1	1	1	1	1	nino.basic@gmail.com
2	Braslau Alan		1	1	1	1	1	alan.braslau@cea.fr
3	Brejcha Miloš		1	1	1	1	1	brejcha@rydyvatelskyservis.cz
4	Březina David		0	1	1	1	1	brezina@davi.cz
5	Egger Willi		1	1	1	1	1	w.egger@hoede.nl
6	Giedryś Anna		0	0	0	0	0	brezina@davi.cz
7	Goddijn Frans		1	1	1	1	1	frans@goddijn.com
8	Gundlach Patrick		1	1	1	1	0	gundlach@spesdata.de
9	Gurawage Michael		1	1	1	1	1	guravage@literatesolutions.com
10	Habrovanská Pavlína		0	1	1	1	1	jan.viktorin@gmail.com
11	Hagen Hans		1	1	1	1	1	pragma@wxs.nl
12	Hajtmar Jaroslav		1	1	1	1	1	hajtmar@yyzs.cz
13	Hála Tomáš		0	1	1	1	1	thala@pef.mendelu.cz
14	Haltiwanger John		1	1	1	1	1	john.haltivanger@gmail.com
15	Hoekwater Taco		1	1	1	1	1	taco@bittext.nl
16	Homolka Lubor		1	1	1	1	1	homolkalubor@gmail.com
17	Horák Karel		0	1	1	1	1	horakk@math.cas.cz
18	König Harald		1	1	1	1	0	koenig@linux.de
19	Kula Jano		1	1	1	1	1	jano.kula@tiscali.cz
20	Miklavec Mojca		1	1	1	1	1	mojca.miklavec.lists@gmail.com
21	Novotný Lukáš		0	1	1	1	1	l.novotny@t-email.cz
22	Olšák Mirek		0	0	0	0	0	mirek@olsak.net
23	Olšák Petr		0	1	1	1	1	petr@olsak.net
24	Pavneet Arora		1	1	1	1	1	pavneet_arora@yaroc.com
25	Píška Karel		1	1	1	1	1	piska@fzu.cz
26	Procházka Lukáš		1	1	1	0	0	LPr@pontex.cz
27	Přichystal Jan		0	1	1	1	1	jprich@pef.mendelu.cz
28	Ramm Henning Hraban		1	1	1	1	1	hraban@ramsch.org
29	Reutenauer Arthur		1	1	1	1	1	arthur.reutenauer@normalesup.org
30	Rybička Jiří		0	1	1	1	1	rybicka@mendelu.cz
31	Scarso Luigi		1	1	1	1	1	luigi.scarso@gmail.com
32	Schuster Wolfgang		1	1	1	1	1	wolfgang.schuster@gmail.com
33	Strzelczyk Piotr		1	1	1	1	0	piotr@eps.gda.pl
34	Stříž Martin		1	1	1	1	1	martin@striz.cz
35	Stříž Milan		1	1	1	1	1	mstriz@tor.cz
36	Stříž Pavel		1	1	1	1	1	striz@fame.utb.cz
37	Střížová Yvona		1	1	1	1	1	yvonastrizova@seznam.cz
38	Štěpnička Jan		0	1	1	1	1	Jan.Stepnicka@osu.cz
39	Štěpničková Iva		0	1	1	1	1	Jan.Stepnicka@osu.cz
40	Stourač Milan		0	1	1	1	1	stourac.milan@seznam.cz
41	Šustek Jan		0	1	1	1	1	Jan.Sustek@osu.cz
42	Talandová Petra		0	1	1	1	1	PTalandova@seznam.cz
43	Tolarová Leona		0	0	0	0	0	leona.tolarova@czso.cz
44	Trušník Roman		0	1	1	1	1	trusnik@fhs.utb.cz
45	Van Deventer Eva		1	1	1	1	1	Vdeveen@unisa.ac.za
46	Vieth Ulrik		1	1	1	1	0	ulrik.vieth@arcor.de
47	Viktorin Jan		0	1	1	1	1	jan.viktorin@gmail.com
48	Voipio Mari		1	1	1	1	0	mari.voipio@xpatents.com
49	Wagner Zdeněk		0	1	1	1	1	zdenek.wagner@gmail.com
50	Wolfrum Steffen		1	1	1	1	0	context@post.verkaatz.com
–	Number of participants:	–	**31**	**47**	**47**	**46**	**40**	–

This document was created (generated) using scancsv.lua library by Jaroslav Hajtmar.
Scancsv.lua presentation to be held on Saturday, 18. 9. 2010 on 4CIH-I-37E in Brejlov.

Obr. 3 Ukázka netriviálního použití knihovny `scancsv.lua`. Využito pro administrativní agendu dvojkonference v Brejlově.

Programme of ConTEXt meeting and TEXperience
Brejlov 13. – 19. 9. 2010

Monday 13.9.

18:30	**Dinner**			90	Notice
20:00	Taco Hoekwater / Hans Hagen / Mojca Miklavec	ConTeXt installation clinic	workshop	120	Notice

Tuesday 14.9.

8:00	**Breakfast**			60	Notice
9:00		Conference opening		15	Notice
9:15	Arthur Reutenauer	Keynote talk	talk	15	Notice
9:30	Taco Hoekwater	Reference manual update	talk	15	Notice
9:45	Patrick Gundlach	Wiki context reference	talk	15	Notice
10:00	Taco Hoekwater / Hans Hagen	Other documentation news	talk	15	Notice
10:15	Peter Münster (presented by Taco)	Reference syntax in lua	talk	15	Notice
10:30	**Coffee**			30	Notice
11:00	Hans Hagen	Why structure matters	talk	30	Notice
11:30	Wolfgang Schuster	Module documentation	talk	30	Notice
12:00	Luigi Scarso	modules_mkiv	talk	30	Notice
12:30	**Lunch**			90	Notice
14:00	Mari Voipio	A Different Philosophy – Thoughts on teaching ConTeXt to non-techies	discussion	30	Notice
14:30	Mari Voipio	Observations on approaches to learning ConTeXt and writing documentation to match	discussion	60	Notice
15:30	**Coffee**			30	Notice
16:00	Wolfgang Schuster	xml interface files and internal module doku (%D, %M etc.)	talk	30	Notice
16:30	Hans Hagen	Requirements For Documentation	talk	30	Notice
17:00	Hans Hagen	My Slowly Growing Test Suite	talk	30	Notice
17:30		Documentation discussion	discussion	60	Notice
18:30	**Dinner**			90	Notice
20:00	Idris Samawi Hamid	Towards the First ConTeXt Book	talk	30	Notice
20:30	Hans Hagen	Document workflow	tutorial	30	Notice
21:00	Hans Hagen	Whatever You Want To Know	q&a	90	Notice

Wednesday 15.9.

8:00	**Breakfast**			60	Notice
9:00	Mari Voipio	Uses for ConTeXt in a standard office environment	talk	30	Notice
9:30	Hans Hagen / Mojca Miklavec	The database module, MkIV version	talk	30	Notice
10:00	Hans Hagen	XML processing news	talk+workshop	60	Notice
11:00	**Coffee**			30	Notice
11:30	Alan Braslau	Plotting data with Metafun/Metapost	talk+discussion	45	Notice
12:15	Alan Braslau	Drawing diagrams using the chart module	talk+discussion	45	Notice
13:00	**Lunch**			90	Notice
14:30	Hans Hagen	Font Goodies	talk	30	Notice
15:00	Mojca Miklavec	Some thoughts on typescripts	talk+discussion	60	Notice
16:00	**Coffee**			30	Notice
16:30	Aditya Mahajan (presented by Luigi)	Beg, borrow, and steal – running external filters in ConTeXt	talk	30	Notice
17:00	Alan Braslau	Drawing chemical structures using ppchTeX	talk	30	Notice
17:30	Wolfgang Schuster	The letter module	talk+workshop	90	Notice
19:00	**Dinner**			90	Notice
20:30	Taco Hoekwater	tlcontrib.metatex.org	talk	30	Notice
21:00	Mojca Miklavec	ConTeXt minimals, "Server edition"	talk+discussion	30	Notice
21:30	Patrick Gundlach	contextgarden.net	q&a	30	Notice

Obr. 4 Pro vygenerování výstupu byl použit zkonvertovaný nasdílený Google document s oficiálním programem dvojkonference (ConTEXt meeting a TEXperience 2010) v Brejlově.

6. `Scancsv.lua` a ConTEXtový modul `t-scancsv.lua`

Hlavní funkce luaknihovny `ParseCSVdata()` obsahuje jen jednoduchý parsovací algoritmus vycházející z formátu CSV tabulek, které používám. Algoritmus není schopen zpracovávat obecné CSV soubory, mající některé položky vymezeny vymezovačem a jiné nikoliv (když položky obsahují oddělovač polí). Stejně tak není schopen správně zpracovat CSV soubory, obsahující uvnitř vymezovačů samotné vymezovače (text pole vymezený uvozovkami obsahuje uvozovky). V ConTEXtovém modulu `t-scancsv.lua` [3], který vznikl kompletním přepracováním luaknihovny na základě inspirujících připomínek a oprav Hanse Hagena, je řada rozšíření a vylepšený parsovací algoritmus.

Skalní TEXisty jistě nepřekvapí, že zdrojový kód luaknihovny je oproti zmíněnému TEXovému zdrojáku makra p. Olšáka značně obsáhlý (a to nejen díky mým rozsáhlým komentářům). Funkčnost obou systémů je v podstatě stejná. Zmíněný ConTEXtový modul má navíc již několik zajímavých vylepšení mj. i možnost použití nepodmíněných i podmíněných cyklů, maker s proměnlivým počtem parametrů, možností relačního spojování tabulek (`1:N`) atd. Podstatným způsobem to rozšiřuje možnosti praktického použití (filtrování rozsáhlých CSV tabulek atd). Svoji původní luaknihovnu `scancsv.lua` (z r. 2010) tak mohu doporučit pro jednodušší praktické použití nebo alespoň jako studijní či inspirační materiál zájemcům o jazyk Lua a jeho používání ve vlastních TEXových dokumentech a makrech. S ConTEXtovým modulem `t-scancsv.lua` lze řešit i náročnější úlohy (viz ukázky na úložišti).

7. Závěr

Popisovaná luaknihovna i ConTEXtový modul vznikly díky laskavé pomoci členů mailové konference `ntg-context@ntg.nl`. Speciálně chci poděkovat Hansovi Hagenovi, Wolfgangu Schusterovi a Taco Hoekwaterovi. Členům mailové konference `csTeX@cs.felk.cvut.cz` chci poděkovat za rady a pomoc týkající se obecně TEXu. Speciální poděkování patří Zdeňku Wagnerovi a Petru Olšákovi. Pavlu Střížovi děkuji za testování knihovny, cenné rady a za to, že mne inspiroval luaknihovnu dopracovat do fáze, kdy ji snad může využít i někdo jiný.

Popisovaná luaknihovna by měla být považována za moji „luaprvotinu“. Nedokonalosti a neoptimálnosti kódu nechť jsou tomu přičítány. Lua jazykem jsem se začal zabývat především proto, aby se mohl oprostit zejména od častého používání jazyka Perl spolu s ConTEXtem (MKII). Pro možnosti jazyka Lua

[3] V červnu 2011 jsem kompletně původní Lua knihovnu přepracoval a vytvořil regulérní luamodul pro ConTEXt MKIV. Tento modul (`t-scancsv.lua`) obsahuje řadu zajímavých vylepšení, nové funkce a je optimalizován pro ConTEXt, v němž byl také již mnohokrát v ostrém provozu otestován.

a jeho integraci do TEXu jsem se nadchnul a všem TEXistům doporučuji Lua jazyk minimálně vzít na milost :-). Doufám, že se mezi čtenáři najdou i ti, kdo se mým příspěvkem nechají inspirovat a rozšíří řady příznivců jazyka Lua a LuaTEXu, LuaLATEXu či ConTEXtu.

Summary: ScanCSV – Lua Library for CSV file ConTEXt and LuaLATEX processing

Data stored in CSV (Comma Separated Values) files are often used in data processing. This article describes the author's `scancsv.lua` library, its origin and demonstrates practical examples of its usage in ConTEXt MKIV and LuaLATEX. Author shows how easily and quickly create print reports, letters, forms, certificates, invitations, cards, business cards, double-sided cards, tables, animations etc. using external CSV text databases.

Users of ConTEXt MKIV (but LuaLATEX and LuaTEX as well) can easily use data from external CSV tables in their own documents via the library, using the TEX macros built on the library and make this data available in an attractive and very simple and natural way.

Jaroslav Hajtmar, `hajtmar@gyza.cz`
Gymnázium Zábřeh
Nám. Osvobození 20
789 01 Zábřeh

Abstrakty vybraných příspěvků z konference TeXperience 2011

Karel Horák: Cvičení s MetaType1
Tutoriál pro zájemce o hlubší poznání vlastností a možností MetaType1.

Pavol Kráľ: O LyX 2.0
V príspevku budú stručne predstavené možnosti a vlastnosti LyXu zaujímavé z hľadiska potenciálneho používateľa systému LaTeX a príbuzných programov.

Pavol Kráľ: Programy na kolektívnu prácu v TeXu
V príspevku budú predstavené niektoré programy určené na kolektívnu prácu s textami v TeXu, napr. ScribTeX, Verbosus, MonkeyTeX a LaTeXlab.

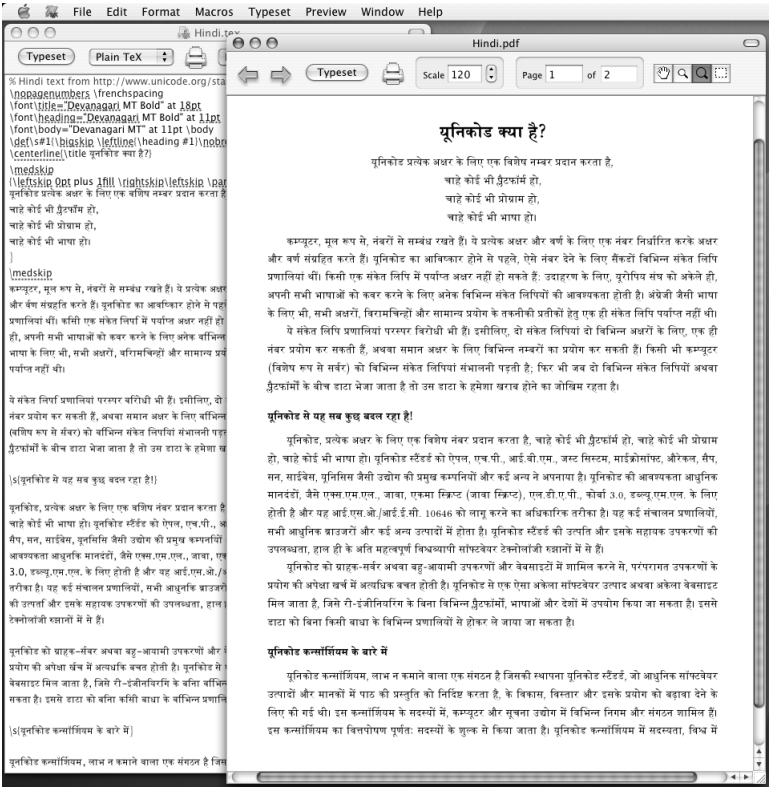
Martin Krčál:
Představení technického řešení a nové služby projektu Citace Pro
Přednáška se zaměří na představení projektu Citace Pro se zmíněním nejdůležitějších funkcí a možnosti importu dat z externích systémů.

Karel Píška: O tvorbě, testování a použití složitého OpenType fontu
The talk presents development of complex OpenType fonts. The sample fonts cover Czech and Georgian handwriting with numerous letter connections. At the beginning, we show general principles of "advanced typography" – complex metric data represented by OpenType tables (GSUB and GPOS) – and compare them with the ligature and kerning tables in METAFONT. Then we describe a history of the OpenType font production – approaches, tools and techniques. We discuss crucial problems, critical barriers, attempts and ways how to reach successful solutions.

We demonstrate several tools for font creating, testing, debugging and conversions between various source and binary formats. Among these tools are, for example, AFDKO, VOLT, FontForge, TTX, Font-TTF. We illustrate their features, advantages, disadvantages, and also cases of possible incompatibilities (or maybe bugs). Finally, we present using the OpenType fonts in the TeX world applications: XeTeX and LuaTeX (ConTeXt MkIV), the programs allowing to read and process OpenType fonts directly.

Roman Plch, Petra Šarmanová:
Matematika hrou – interaktivní výukové materiály v PDF formátu
Cílem příspěvku je představit možnosti, které poskytuje TeX tvůrcům interaktivních výukových materiálů v PDF formátu. Na příkladech výukových objektů z matematické analýzy ukážeme použití a možnosti interaktivní 3D a 2D grafiky, testů a her.

Jiří Rybička: XeLaTeX – krok správným směrem (zvaná přednáška)
Mnoho uživatelů osobních počítačů má zájem vytvořit kvalitní dokumenty, nejčastěji ze své profesní oblasti, tedy odborné texty s řadou speciálních symbolů. Donedávna byla jednou z mála skutečně použitelných variant sazba v systému LaTeX – avšak vzhledem k jeho historické koncepci vznikala řada obtížně řešitelných problémů zejména s fonty a kódováním vstupu. Příspěvek se zabývá možnostmi a některými zkušenostmi při použití systému XeLaTeX, jehož vlastnosti mohou výrazně přispět k zachování kvalitní sazby jako ve známém a rozšířeném systému LaTeX, ale s mnoha zajímavými rozšířeními: využití fontů instalovaných v rámci operačního systému, možnosti vstupu znaků v kódování UTF-8 a s tím spojené možnosti sazby v různých jazycích, možnosti matematické sazby a použití speciálních symbolů potřebných pro odborné texty. Potenciál *XeTeXu* pro cizojazyčnou sazbu ilustruje přiložený obrázek.



Pavel Stříž, Michal Mádr:
Překlad *The Not So Short Introduction to LaTeX 2ε* po 12 letech a zkušenosti s jeho zveřejněním na ctan.org

Představíme balíček lshort-czech[1] volaný např. přes texdoc lshort-czech v příchozím TeX Live 2011. Snažili jsme se doplnit mezeru v sadě překladů knihy *The Not So Short Introduction to LaTeX 2ε*. Ukážeme si přípravu balíčku před odesláním na ctan.org[2], a dílčí kroky, které následovaly. Přednáška by měla být inspirací pro ty, kteří zvažují zaslat svůj balíček, ale do pozadí zatím nevidí.

Druhá verze[3] (zdrojové kódy[4]) ke stažení a komentování, 15. června 2011. Zveřejněno na CTAN[5].

Pavel Stříž:
Matematické výpočetní prostředí Sage (Python) a SageTeXu

Tvůrci Sage si dávají za cíl vytvořit open-source alternativu komerčním systémům jako jsou Matlab, Mathematica či Maple. Platí to i pro interaktivní prostředí, které je poměrně známé u systému Mathematica. Systém je podrobně zdokumentován. Zobrazování 3D grafiky většinou zajišťuje Jmol, který je znám především chemikům a biologům. Výpočetní prostředí je složeno z dílčích open-source programů unifikovaných silným programovacím jazykem Python, který se snaží být čitelný a srozumitelný a pomocí knihoven je snadno rozšiřitelný, viz PyPi.

V přednášce si ukážeme instalaci a virtualizaci Sage, první výpočetní kroky a grafické výstupy až po užití knihoven na problematiku combinadics (combinatorial number system). Tuto oblast pravděpodobnosti (užití například v karetní hře bridž a její nemožnou knihu rozdání, viz řešení Richard Pavlicek nebo Thomas Andrews) se pokusíme společnými silami naprogramovat v Pythonu tak, abychom nepotřebovali ani externí knihovnu, v tomto případě se bude jednat o Sage-combinat, který je znám především uživatelům staršího prostředí MuPAD a jeho MuPAD-CombiNAT. Tím si zajistíme nejen možnost naše výstupy zveřejnit na The Sage Notebook, ale také jako na Sage nezávisle spustitelný zdrojový kód v Pythonu s možností si snadno vytvořit spustitelné soubory, viz cx_Freeze.

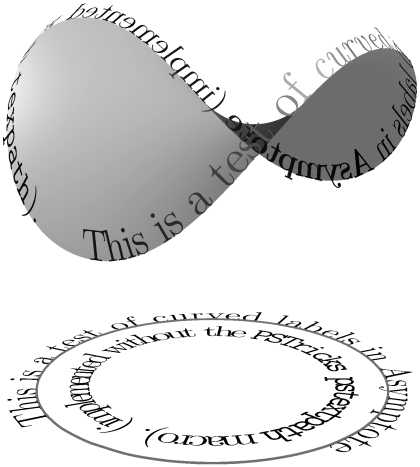
Okrajově si též během přednášky vyzkoušíme funkčnost SageTeXu, viz soubor sagetex.sty, a nahlédneme na jeho dokumentaci.

[1] http://ftp.cstug.cz/pub/tex/CTAN/info/lshort/english/lshort.pdf
[2] http://ctan.org/tex-archive/info/lshort/czech
[3] striz9.fame.utb.cz/texperience/2011/lshort-cs.pdf
[4] striz9.fame.utb.cz/texperience/2011/lshort-czech-4.27.src+build.tar.gz
[5] http://ctan.org/tex-archive/info/lshort/czech

Pavel Stříž:
Interaktivní dokument v Sage a představení programu Asymptote s doplňujícím LaTeXovým balíčkem asymptote

Vrcholem předchozí přednášky o Sage bude příprava interaktivního dokumentu v Sage na rotaci křivky zadané $y = f(x)$ kolem osy x nebo osy y. Výstupem našich snah bude možnost si vybrat osu, úhel (nula až 360 stupňů) a explicitně zadanou křivku (kvůli zjednodušení si navolíme tři učebnicové případy). Sage nám vykreslí vybranou křivku (2D graf), její rotaci (3D graf) a numericky spočte obsah (S) a objem (V) vznikajícího tělesa (zvolíme Method of Rings či Method of Cylinders).

Problém nastává tehdy, pokud chceme získat 3D model v PDF (výstupy z Jmolu nebyly ideální, dle mých testů). S touto situací nám pomůže program Asymptote, který byl inspirován Metapostem, a s LaTeXovým balíčkem asymptote. Naší snahou bude získat interaktivní 3D model bez integrálních výpočtů, což je vhodná forma na tisk a do elektronických knih. Jako prohlížeč využijeme Adobe Reader. Závěr přednášky bude tvořit ukázka TeXové sazby na plochu takového 3D objektu. A možná dojde i na popsání objektů jiných…



Pavel Stříž, Libor Sarga, Lubor Homolka:
Kolektivní práce pomocí Google dokumenty a nástrojů typu EtherPad

Poukážeme na tyto dva reprezentanty editačních nástrojů, a ukážeme si, že lze bez větších problémů editovat i TeXový zdrojový kód. To se může hodit při přípravě programů, bodů jednání, úvodníků apod.

Pavel Stříž:
O přípravě jednoho grafu na zakázku přes PGFplots a PGFplotstable

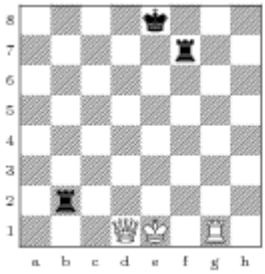
Příspěvek představuje vybrané partie dvou LaTeXových balíčků PDFplots a PGFplotstable z TeXového světa. Zdrojové kódy jsou vždy pod 1 KiB. Autor ukazuje dílčí situace, které vyústí v netradiční, programovatelný graf na konci našich snah. Bude se jednat o graf – metrovou grafovou housenku – která je autorem představena v samotném úvodu/závěru tohoto konferenčního příspěvku. A jako taková bude předána v podobě dárku příchozím.

Mezi netradiční řešené partie můžeme zařadit: stupňovité měřítko vodorovné osy, několik hlavních a vedlejších svislých os v jednom grafu, mřížka umístěná do pozadí, umístění objektů absolutně či relativně vůči souřadnicovému systému, úprava dat za běhu TeXu, změna vzhledu značek, nastavení stylu a sazba matematiky za účasti TeXu, zvětšení grafu se zachováním os, příprava podkladů pro balíček animate, jako vhodné dělení grafu (s i bez překryvu) pro vložení dílčích částí pod sebe atp.

Pavel Stříž: Sazba šachu v TeXu a práce s balíčkem animate

Budou představeny TeXové balíčky xskak a chessboard. Naznačena bude též sazba exošachu pomocí definování vlastních figur a za užití XeLaTeXu.

Bude ukázáno jednoduché i náročnější užití balíčku animate a jeho použití (nejen) na šachové diagramy.



Zdeněk Wagner:
Daňová evidence pomocí LaTeXu, XML, Tcl/Tk a Subversion

Přednáška bude zaměřena především na ukázky běhu, krátké ukázky kódů budou součástí článku.

Vedení účetní a daňové evidence bez pomoci výpočetní techniky patří již nenávratně do minulého století. V současné době klesá význam papírových daňových dokladů a dokumenty v elektronické podobě hrají stále větší roli. Na dokumenty, používané k výpočtu základu daně, jsou však kladeny větší požadavky. Forma takového dokladu musí být v souladu s odpovídajícími zákony, které se však mění. Je tedy nutno pružně reagovat na všechny změny. Na druhou stranu je daňový subjekt povinen před uplynutím skartační lhůty vytisknout opakovaně daný dokument v identické podobě. Tento příspěvek ukazuje, jak lze pomocí LaTeXu, XML, Subversion a dalších nástrojů vytvořit systém dostatečně flexibilní, ale na druhou stranu s omezujícími podmínkami na předepsaný vzhled dokumentu. Nástroje ze světa XML slouží k definici struktury a validaci dokladů i k provádění výpočtů, jakým je např. výpočet DPH. LaTeX je využit k tisku dokumentů, Tcl/Tk k tvorbě GUI. Použití Subversion umožní kromě historických pohledů na hospodaření i vazbu mezi dokladem a softwarovými nástroji platnými v době vystavení dokladu.

Navržený systém je vhodný pro drobné živnostníky. Je postaven jako stavebnice, řeší jen potřeby autora, v žádném případě se nejedná o kompletní implementaci všech účetních a daňových zákonů. V přednášce budou představeny principy činnosti a bude předvedeno použití na modelovém příkladě.

Sazba zdrojových kódů

Petra Čačková

Článek je věnován možnostem, které LaTeX nabízí pro sazbu algoritmů a zdrojových kódů programů obecně. Jsou představeny vybrané balíčky, především balíček listings, jsou nastíněny výhody (a nevýhody) vybraných řešení, nechybí ani ukázky.

Klíčová slova: LaTeX, algoritmy, zdrojové kódy, programování, doslovný výpis, listings.

1. Úvod

„V TeXu jde všechno.“ Jednou podskupinou toho „všeho“ je i příprava dokumentace, příručky k programům, materiály na školení a sbírky příkladů. Nahlédnutí pod povrch aneb „jak to funguje“, typické pro tyto dokumenty, často vyžaduje sazbu algoritmů a zdrojových kódů programů. V tomto článku jsou představeny některé možnosti jejich sazby i s ukázkami.

2. Balíčky a řešení

Základní džungle balíčků je dostupná na CTANu [1], seznam řešení odpovídající klíčovému slovu „computer code, verbatim text“ má nyní 83 položky. Ty by se dále mohly členit, například takto:

- styl „psací stroj“: balíčky alltt, alltt2,
- doslovný text: cprotect, fancyvrb, verbatim, shortvrb; listings…,
- základní algoritmy, pseudokód: algorithms, alg, algorithm2e, algorithmicx atd.,
- podle programovacích jazyků: c2cweb, cnoweb, c-pascal, cprog; javadoc atd.,
- LaTeXový kód: codedoc; example, examplep atd.
- zvýrazňování syntaxe (s Pygments): texments, verbments atd.

Je tedy z čeho vybírat a jsou dostupná jak obecná řešení, tak úzce specializované balíčky. Následuje přehled vybraných řešení a balíčků s popisem, případně s ukázkami.

2.1. Prostředí `tabbing`

Ještě před přehledem balíčků se zastavíme u standardního prostředí `tabbing` [2, s. 98]. Díky nastavení tabelačních zarážek lze dosáhnout pěkného a přehledného

doi: 10.5300/2012-2/97

výsledku. Zde však výhody končí, kód se může jevit jako složitý a nepřehledný a přizpůsobení (například zvýrazňování klíčových slov) je třeba řešit ručně.

Prostředí tabbing

```
\begin{tabbing}
\textbf{procedure} Fibo(pocet: byte);\\
// \textit{procedura pro výpis celé řady až po n. člen} \\
\textbf{var} \= clen1, clen2, pomocny: word;\\
             \> i: byte;\\
\textbf{begin} \+ \\
  clen1:=0; clen2:=1; // \textit{hodnoty prvních dvou členů} \\
  \textbf{if} \= (pocet > 0) \textbf{then begin}\+\\
    ...\-\\
  \textbf{end};\- \\
\textbf{end};\\
\end{tabbing}
```

```
procedure Fibo(pocet: byte);
// procedura pro výpis celé řady až po n. člen
var clen1, clen2, pomocny: word;
    i: byte;
begin
    clen1:=0; clen2:=1; // hodnoty prvních dvou členů
    if (pocet > 0) then begin
      ...
    end;
end;
```

2.2. Prostředí a balíček verbatim

Prostředí `verbatim` je klasikou pro doslovný výpis neproporcionálním písmem a nepodléhající formátování. Použití je jednoduché, ale zde opět výhody končí, žádné formátování nebo zvýrazňování klíčových slov. Kromě toho je prostředí nevhodné pro tvorbu vlastních maker (nesmí být parametrem jiného příkazu).

Lepší možnosti práce s doslovným textem poskytuje balíček VERBATIM [3]. Název prostředí zůstává stejný a jeho možnosti v podstatě také, navíc ale umožňuje vložení textového souboru příkazem `\verbatiminput`. Dále podporuje víceřádkové komentáře v prostředí `comment`.

Balíček verbatim

```
Víceřádkový komentář, který se nevysází:
\begin{comment}
```

```
  Program, který má na vstupu řetězec.
  Vypisuje z~něj podřetězce vždy o~jeden znak delší než předchozí.
\end{comment}

Vstup ze souboru:
\verbatiminput{string.pas}
```

Víceřádkový komentář, který se nevysází:
Vstup ze souboru:

```
var s: string; i: byte;
begin
  readln(s);
  for i:=1 to length(s) do writeln(copy(s,1,i));
end.
```

2.3. Balíček alltt

Další balíček pro doslovný výpis úseku dokumentu neproporcionálním písmem, složené závorky a zpětné lomítko si zachovávají svůj význam a zdroj tedy lze doplnit o příkazy [4]. Práce s tímto balíčkem je tedy o něco pružnější než u prostředí/balíčku VERBATIM, avšak použití pro výpis zdrojového kódu (který je u mnoha jazyků plný složených závorek) je poněkud nepraktické a vstup zdrojového textu přímo ze souboru je problematický. Existují však jiná vhodná řešení.

Balíček alltt

```
\begin{alltt}
\kw{function} RFibo(n: byte): word;
\textit{// rekurzivní funkce, vrátí hodnotu n. prvku posloupnosti}
\kw{begin}
  \kw{case} n \kw{of}
    0: RFibo:=0;
    1: RFibo:=1;
    \kw{else} RFibo:=RFibo(n−1) + RFibo(n−2);
  \kw{end};
\kw{end};
\end{alltt}
```

```
function RFibo(n: byte): word;
// rekurzivní funkce, vrátí hodnotu n. prvku posloupnosti
begin
  case n of
    0: RFibo:=0;
```

```
    1: RFibo:=1;
    else RFibo:=RFibo(n-1) + RFibo(n-2);
  end;
end;
```

2.4. Balíčky algorithmic a algorithm

Sada Algorithms [5] obsahuje balíčky ALGORITHMIC a ALGORITHM, které poskytují stejnojmenná prostředí. ALGORITHMIC slouží pro sazbu (pseudo)kódu a ALGORITHM vytváří plovoucí prostředí pro tyto algoritmy.

Balík ALGORITHMIC počítá s použitím pseudokódu a definované množiny klíčových slov. Ta jsou zavedena jako makra, dají se v případě potřeby snadno předefinovat, například do národních jazyků. V prostředí lze číslovat řádky kódu (uvedením volitelného parametru, např. `\begin{algorithmic}[5]`) a také se na ně odvolávat.

Balík ALGORITHM nabízí další možnosti – popisky algoritmů, vytvoření jejich seznamu (příkazem `\listofalgorithms`), různé způsoby orámování. Číslování algoritmů může být odvozeno od částí dokumentu (např. uvedením volitelného parametru `[chapter]` u příkazu `\usepackage{algorithm}`).

Balíčky algorithmic a algorithm

```
\begin{algorithm}
\caption{Fibonacci}\label{fibo}
\begin{algorithmic}[1]
\STATE F:
  \IF{prvek = 0} \RETURN{0}
    \ELSIF{prvek = 1} \RETURN{1}
    \ELSE \RETURN{F(prvek−1) + F(prvek−2)}
  \ENDIF
\end{algorithmic}
\end{algorithm}
```

```
1: F:
2: if prvek = 0 then
3:    return 0
4: else if prvek = 1 then
5:    return 1
6: else
7:    return F(prvek-1) + F(prvek-2)
8: end if
```

Algorithm 1: Fibonacci

2.5. Balíček c-pascal

Je přizpůsobený přímo pro oba tyto programovací jazyky [6]. Jde tedy jen o dvouúčelový nástroj, ale pokud je s tím uživatel spokojen, může se jednat o velmi elegantní řešení. V kódu se automaticky zvýrazňují klíčová slova, jsou použity různé řezy písma, výpis programu je pohodlný a přehledný. K tomu může přispět i možnost načítání kódu ze zadaného souboru (vhodné např. pro sbírku příkladů).

Balíček c-pascal

```
\BeginPascal
function RFibo(prvek: byte): word;
(* rekurzivni funkce, vrati hodnotu n. prvku posloupnosti *)
begin
  case n of
    0: RFibo:=0;
    1: RFibo:=1;
    else RFibo:=RFibo(n-1) + RFibo(n-2);
  end;
end;
\EndPascal

\InputPascal{string.pas}
```

```
function RFibo(prvek: byte): word;
(* rekurzivni funkce, vrati hodnotu n. prvku posloupnosti *)
begin
  case n of
    0: RFibo:=0;
    1: RFibo:=1;
    else RFibo:=RFibo(n-1) + RFibo(n-2);
  end;
end;

var s: string; i: byte;
begin
  readln(s);
  for i:=1 to length(s) do writeln(copy(s,1,i));
end.
```

2.6. Balíček lineno

Balíček LINENO [7] je určen pro číslování jednotlivých řádků vstupu. Lze jej použít pro zdrojové kódy programů i odstavcový text. Odstavec je rozlámán do jednotlivých řádků, k nimž jsou připojena čísla – s možností odkazovat se na ně pomocí LaTeXového mechanismu křížových odkazů. Balíček nabízí různé možnosti nastavení číslování. Namísto tradičního prostředí poskytuje příkazy pro zapnutí a vypnutí číslování.

```
function RFibo(prvek: byte): word;
// rekurzivni funkce, vrati hodnotu n. prvku posloupnosti
begin
  case n of
    0: RFibo:=0;
    1: RFibo:=1;
    else RFibo:=RFibo(n-1) + RFibo(n-2);
  end;
end;
```

Balíček lineno

```
\linenumbers
Balíček \pkg{lineno} je určen pro číslování jednotlivých řádků vstupu.
Lze jej použít pro zdrojové kódy programů... vypnutí číslování.

\resetlinenumber[1]
\begin{verbatim}
function RFibo(prvek: byte): word;
// rekurzivni funkce, vrati hodnotu n. prvku posloupnosti
 ...
\end{verbatim}
\nolinenumbers
```

2.7. Balíček fancyvrb

Balíček FANCYVRB [8] rozšiřuje možnosti výpisu doslovného textu. Nabízí prostředí `Verbatim`, které je univerzálně použitelné pro různé programovací jazyky i pro pseudokód, syntaxe se tedy nezvýrazňuje. Pokud se ale v nepovinném parametru prostředí definují znaky, které uvádějí makro a obklopují skupinu (`\begin{Verbatim}[co` lze v kódu používat i příkazy a výpis tak přizpůsobit. Kromě toho jsou k dispozici rozsáhlé možnosti přizpůsobení (způsob sazby, změna typu, řezu a stupně písma, formátování, barva, rámeček, číslování řádků).

`Verbatim` se v nové verzi balíčku dá použít i v poznámce pod čarou.

Lze také definovat znak, který bude použit místo příkazu `\verb` (např.

`\DefineShortVerb{\!}`) a například v odstavcových textech je pak možné používat kratší způsob zápisu (`!mujtext!`).

Vložit zdrojový kód přímo ze souboru je možné příkazem `\VerbatimInput`.

Balíček je navíc flexibilní, nabízí více prostředí typu verbatim – `BVerbatim` (materiál do boxu) a `LVerbatim` (LR mód). Je možné předefinovat existující prostředí i definovat vlastní.

Balíček FANCYVRB tedy poskytuje významná rozšíření pro sazbu doslovného textu s rozsáhlými možnostmi formátování a dalšího přizpůsobení. I když není určen přímo pro sazbu zdrojových kódů, i k tomuto se dá výborně využít.

Balíček fancyvrb, sazba kódu v LaTeXu

```
\begin{Verbatim}[frame=single,label=Sazba kódu v~\LaTeX u,numbers=left]
\begin{itemize}
  \item Mouka
  \item Vajíčka
  \item \textbf{Smetana}
\end{itemize}
\end{Verbatim}
```

Sazba kódu v LaTeXu

```
1 \begin{itemize}
2   \item Mouka
3   \item Vajíčka
4   \item \textbf{Smetana}
5 \end{itemize}
```

Balíček fancyvrb, sazba kódu v Pascalu

```
\DefineShortVerb{\|}

Ukázka pro ne\LaTeX ový zdrojový kód: funkce |RFibo| v~Pascalu.

\begin{Verbatim}[frame=single,label=Sazba kódu v~Pascalu,numbers=left]
function RFibo(n: byte): word;
begin
  case n of
    0: RFibo:=0;
    1: RFibo:=1;
    else RFibo:=RFibo(n-1) + RFibo(n-2);
  end;
end;
\end{Verbatim}
```

Ukázka pro neLaTeXový zdrojový kód: funkce `RFibo` v Pascalu.

Sazba kódu v Pascalu

```
function RFibo(n: byte): word;
begin
  case n of
    0: RFibo:=0;
    1: RFibo:=1;
    else RFibo:=RFibo(n-1) + RFibo(n-2);
  end;
end;
```

Balíček fancyvrb, vstup ze souboru

```
\VerbatimInput{string.pas}
```

```
var s: string; i: byte;
begin
  readln(s);
  for i:=1 to length(s) do writeln(copy(s,1,i));
end.
```

Je potřeba uvést, které znaky uvádějí makro (\\) a obklopují skupinu. Pak lze používat i příkazy.

Balíček fancyvrb, používání příkazů

```
\begin{Verbatim}[commandchars=\\\{\}]
function RFibo(prvek: byte): word;
\textcolor{seda}{begin}
  case prvek of (...) end;
\textcolor{seda}{end;}
\end{Verbatim}
```

```
function RFibo(prvek: byte): word;
begin
  case prvek of (...) end;
end;
```

2.8. Balíček listings

Balíček LISTINGS [9] je zřejmě nejoblíbenějším a v diskuzích nejdoporučovanějším balíčkem, který řeší sazbu zdrojových kódů. Seznam výhod je opravdu rozsáhlý:

- automatické zvýrazňování syntaxe pro desítky předdefinovaných jazyků,
- možnost dodefinovat další jazyky,
- bohaté možnosti nastavení sazby (formátování, číslování řádků, rámečky atd.),

- možnost vytvoření vlastního stylu formátování výpisů (`\lstdefinestyle`),
- vkládání úryvků kódu do odstavce i se zvýrazňováním syntaxe (příkaz `\lstinline`)
- vstup přímo ze souboru (`\lstinputlisting`)
- doplnění výpisu o titulek,
- vygenerování seznam algoritmů (na základě titulků).

Za nevýhody lze považovat výchozí nastavení formátování a práci s kódováním souboru.

Jednotlivé znaky ve výpisu kódu jsou ve výchozím nastavení zarovnány na jednotlivé sloupce, a to i přesto, že je použito proporcionální písmo. Text pak působí jako prostrkaný a je hůře čitelný. (Tento efekt se týká prostředí `lstlistings`, ne vkládaných souborů.) Popsané chování lze přepnout změnou nastavení z výchozího `columns=fixed` na `columns=flexible` v seznamu parametrů příkazu `\lstset`.

Druhou nevýhodou jsou problémy s kódováním národních znaků. Nesprávné zobrazení znaků může vyřešit parametr `extendedchars=true` u příkazu `\lstset`. Je třeba zavést FONTENC nebo INPUTENC, předpokládá se jednobajtové kódování. Má-li být vstup v UTF-8, je to řešitelné balíčkem LISTINGSUTF8 [10], který však vstup v UTF8 převádí „zpět“ do jednobajtového kódování.

Ukázky použití balíčku listings

Ukázka nastaveníformátování výpisu pro Pascal

```
\lstset{%
language=Pascal,
morecomment=[l]{//}, % další typ komentáře
numbers=left,        % čísla vlevo od zdrojového kódu
stepnumber=1,        % čísluje se každý řádek
numberstyle=\tiny,   % velikost čísel
basicstyle=\small,   % velikost písma kódu
frame=single,        % orámování kódu
columns=flexible     % odstraní zarovnání na sloupce
}
```

Ukázka vlastního stylu výpisu

```
% Definice:
\lstdefinestyle{moje}{numbers=left, stepnumber=1, frame=lines}
...
% Použití:
\begin{lstlisting}[title={Ukázka balíčku listings},style=moje]...
```

Ukázka balíčku listings – Pascal

```
\begin{lstlisting}[ title =Rekurzivní Fibonacci,style=moje]
function RFibo(prvek: byte): word;
(* rekurzivní funkce, vrátí hodnotu n. prvku posloupnosti *)
begin
  case n of
    0: RFibo:=0;
    1: RFibo:=1;
    else RFibo:=RFibo(n-1) + RFibo(n-2);
  end;
end;
% ukončení prostředí  lstlisting
```

Rekurzivní Fibonacci

```
function RFibo(prvek: byte): word;
(* rekurzivní funkce, vrátí hodnotu n. prvku posloupnosti *)
begin
  case n of
    0: RFibo:=0;
    1: RFibo:=1;
    else RFibo:=RFibo(n-1) + RFibo(n-2);
  end;
end;
```

Ukázka balíčku listings – vstup ze souboru

```
\ lstinputlisting [ title =Výpis řetězce (ze souboru)]{string.pas}
```

Výpis řetězce (ze souboru)

```
1 var s: string; i: byte;
2 begin
3   readln(s);
4   for i:=1 to length(s) do writeln(copy(s,1,i));
5 end.
```

Ukázka balíčku listings – LaTeX

```
% nastavení: ... language=[LaTeX]TeX, ...
\begin{lstlisting}[ title =Tabulka v~\LaTeX u,style=moje]
\begin{tabular}{|c|l|l|}\hline
\bfseries BMI & \bfseries Výsledek & \bfseries Doporučení \\ \hline
méně než 18,5 & podváha & Sněz něco! \\ \hline
```

```
18,5−−24,9    & v~normě & Dobré, pokračuj. \\ \hline
25,0−−29,9    & nadváha & Nejez brambůrky, dej si mrkev. \\ \hline
30,0 a~vyšší  & obezita  & Necpi se!  \\ \hline
\end{tabular}
% ukončení prostředí  lstlisting
```

Tabulka v LaTeXu

```
\begin{tabular}{|c|l|l|}\hline
\bfseries BMI & \bfseries Výsledek & \bfseries Doporučení \\ \hline
méně než 18,5 & podváha & Sněz něco! \\ \hline
18,5−−24,9    & v~normě & Dobré, pokračuj. \\ \hline
25,0−−29,9    & nadváha & Nejez brambůrky, dej si mrkev. \\ \hline
30,0 a~vyšší  & obezita  & Necpi se!  \\ \hline
\end{tabular}
```

Ukázka balíčku listings – C

```
\begin{lstlisting}[ title =Zjištění  dělitelů  v~C,style=moje]
#include <stdio.h>
int  i ,  cislo ;
int  main() {
   printf ("Zadej  číslo  pro  zjištění  dělitelů  \n");
   scanf (" %d",&cislo);
   for (i=1;i<cislo; i++) if( cislo %i==0) printf("%d ",i);
   getch ();
   return (0);
}
% ukončení prostředí  lstlisting
```

Zjištění dělitelů v C

```
#include <stdio.h>
int  i ,  cislo ;
int  main() {
   printf ("Zadej␣číslo␣pro␣zjištění␣dělitelů␣\n");
   scanf("%d",&cislo);
   for(i=1;i<cislo; i++) if( cislo%i==0) printf("%d␣",i);
   getch ();
   return(0);
}
```

2.9. Další možnosti

Uvedený přehled balíčků zdaleka není vyčerpávající. Obsahuje vybraná čistě LaTeXová řešení založená na stávajících připravených balíčcích. Je však možné jít ještě dál a využít například možností balíčků TEXMENTS a VERBMENTS. Oba používají zvýrazňovač syntaxe Pygments (`http://pygments.org/`). Zde už se ale nejedná jen o LaTeXové řešení, nejprve je potřeba nainstalovat Pygments, jakož i Python.

3. Shrnutí

Namísto závěru následuje tabulka, která stručně shrnuje vlastnosti vybraných „univerzálních" prostředků pro sazbu zdrojových kódů a algoritmů. Výběr konkrétního řešení však závisí především na účelu, který má být splněn.

	tabbing	**verbatim**	**lineno**	**fancyvrb**	**listings**
číslování řádků	–	–	+	+	+
autom. zvýraz. klíčových slov	–	–	–	–	+
formátování výpisu	+	–	–	+	+
popisek	–	–	–	+	+
úsek v odstavci	–	+	+	+	+
vložení souboru	–	–/+[1]	–/+[1]	+	+
plovoucí prostř.	–	–	–	–	+
seznam algoritmů	–	–	–	–	+
XeLaTeX	+	+	+	+	+[2]

Reference

[1] *Packages by keyword* [online]. 2011 [cit. 2011-11-30]. URL `http://www.ctan.org/keyword/computer_code`.

[2] Rybička, Jiří. *LaTeX pro začátečníky*. 3. vydání. Brno: Konvoj, 2003. 238 s. ISBN 80-7302-049-1.

[3] Schöpf, Rainer; Rowley, Chris. *A New Implementation of LaTeX's* ***verbatim*** *and* ***verbatim**** *Environments* [online]. 2001-03-12 [cit. 2011-11-30]. URL `http://ftp.cvut.cz/tex-archive/macros/latex/required/tools/verbatim.pdf`.

[4] Braams, Johannes. *The* ***alltt*** *environment* [online]. 1997-06-16 [cit. 2011-10-27]. URL `http://www.tug.org/texlive/devsrc/Master/texmf-dist/doc/latex/base/alltt.pdf`.

[1] je možné při použití balíčku VERBATIM

[2] potíže s kódováním národních znaků

[5] Brito, Rogério. *The `algorithms` bundle* [online]. 2009-08-24 [cit. 2011-11-11]. URL `http://mirrors.ctan.org/macros/latex/contrib/algorithms/algorithms.pdf`.
[6] Gulczynski, Michal. *PCAP — macro package for typesetting programs in Python, C and Pascal* [online]. [cit. 2011-10-20] `http://ftp.cvut.cz/tex-archive/macros/generic/c_pascal/README.eng`.
[7] Böttcher; Stephan, Lück, Uwe. *A LaTeX package to attach line numbers to paragraphs* [online]. 2005-11-02 [2011-11-15]. URL`ftp://ftp.cstug.cz/pub/tex/CTAN/macros/latex/contrib/lineno/lineno.pdf`.
[8] Van Zandt, Timothy. *The `'fancyvrb'` package: Fancy Verbatims in LaTeX* [online]. 2010-05-15 [cit. 2011-11-16]. URL `ftp://ftp.cstug.cz/pub/tex/CTAN/macros/latex/contrib/fancyvrb/fancyvrb.pdf`.
[9] Heinz, Carsten; Moses, Brooks. *The `Listings` Package* [online]. 2007-02-22 [cit. 2011-11-20]. URL `http://ftp.cstug.cz/pub/tex/CTAN/macros/latex/contrib/listings/listings.pdf`.
[10] Oberdiek, Heiko. *The `listingsutf8` package* [online]. 2007-11-11 [cit. 2011-11-30]. URL `http://ftp.cstug.cz/pub/tex/CTAN/macros/latex/contrib/oberdiek/listingsutf8.pdf`.

Summary: Source Code Typesetting

The article describes LaTeX possibilities for algorithms and source code typesetting. Chosen packages are presented, mainly the LISTINGS package Jsou představeny vybrané balíčky, především balíček LISTINGS, advantages and disadvanteges of chosen solutions are outlined, examples included.

Key words: LaTeX, algorithms, source code, programming, verbatim text, LISTINGS.

Petra Čačková
petra.cackova@mendelu.cz

Přechod na XeTeX/XeLaTeX a další vylepšení na TeXonWeb

Jan Přichystal

Abstrakt

Tento článek se věnuje popisu nových vlastností webové aplikace TeXonWeb. Jedná se především o přechod na kódování UTF8 a začlenění překladače XeTeX a XeLaTeX. V článku jsou také diskutovány vlastnosti, které jsou v současnosti ve stádiu vývoje a které budou implementovány v dohledné době. Mezi tyto funkce patří verzování dokumentů, integrace cloudových služeb a další funkce, které by měly zpříjemnit a usnadnit tvorbu TeXových dokumentů uživatelům aplikace TeXonWeb, tedy především uživatelům s TeXem začínajícím.

1. Úvod

Jak už se pomalu stává tradicí, informujeme na stránkách Zpravodaje o vývoji webové aplikace pro tvorbu TeXových dokumentů prostřednictvím webového prohlížeče, systému TeXonWeb. I v článcích z předchozích let se vždy objevovaly nové zásadní vlastnosti systému a nejinak je tomu i tentokrát. TeXonWeb prošel opět poměrně zásadním vývojem, který významně ovlivňuje jeho fungování a posouvá jej opět o kousek dál, jak v oblasti uživatelského komfortu, tak v množině nabízených funkcí.

2. Co se podařilo realizovat

Nejprve se pojďme podívat na to, co jsme slibovali v předchozích letech a co se nám podařilo implementovat. Patrně nejvýznamnější změnou, která se promítla i do dalších částí této aplikace, je přechod na kódování UTF8 jako kódování nativní. To znamená, že dokumenty, které uživatel na TeXonWeb vytváří, jsou implicitně v tomto kódování. Tento krok byl již dlouho plánován a v podstatě je vynucen současnou situací, kdy toto kódování se stává standardem a velmi usnadňuje zápis nejrůznějších znaků národních abeced bez složitých konstrukcí. Zdálo, se že se bude jednat o poměrně jednoduchý krok, protože bezproblémovost spojení TeXu, OS Linux a UTF8 byla již před tím ověřena na pracovním počítači autora tohoto článku.

Nicméně, jak už to tak bývá, při přechodu na UTF8 jsme narazili na neobvyklý problém. Ten spočíval v tom, že písmeno e s háčkem se přímo v HTML

 doi: 10.5300/2012-2/110

prvku `textarea`, který je použit pro editaci TEXového dokumentu, nesprávně zobrazovalo jako HTML entita. Tento problém se neprojevoval u žádného jiného českého znaku s akcentem a neprojevoval se ani na jiných webových stránkách, kde to autor testoval. Řešení, které se podařilo po dlouhé době nalézt, spočívalo v odstranění obezličky umístěné přímo v modulu CGI.pm. Tento modul je v programovacím jazyce Perl zodpovědný za renderování stránky. Z jakýchsi historických důvodů zde vývojáři zanechali opravu, která při použití kódování UTF8 způsobovala zmíněný problém. Bylo nutné v metodě `escapeHTML` zakomentovat řádek obsahující text `$toencode =~ s{\x9b}{&#8250;}gso;`. Dále se pak již tento problém neobjevoval.

Ve spojení s přechodem na kódování UTF8 došlo i k začlenění překladačů XeTeX a XeLaTeX, které umí toto kódování naplno využít. Jsou již standardní součástí instalační sady TEXLive a proto s jejich instalací nebyl spojen žádný problém. Rozhodli jsme se také nastavit překladač XeLaTeX jako defaultní, což vedlo k několika drobným změnám v uživatelském rozhraní. Ty však způsobily některým uživatelům těžkosti. Problém spočíval především v tom, že XeLaTeX vyžaduje poněkud jinou strukturu hlavičky dokumentu, respektive vyžaduje připojení balíčku `xltxtra`. To znamená, že původní definice pro překladač `cslatex` s novým překladačem nefungovala a mnoho uživatelů bylo překvapeno, že jim dokument najednou nejde přeložit. Další problém způsobovalo nové kódování. Původní dokumenty napsané v kódování ISO Latin 2 bylo nutné překódovat nebo použít původní překladač `cslatex` ovšem s připojeným balíčkem pro vkládaní UTF8 znaků. Na tyto skutečnosti byli samozřejmě uživatelé upozorněni, ale zřejmě ne dostatečně, neboť počet dotazů na nefunkčnost byl po nasazení změn poměrně velký. V současné chvíli je situace již stabilizovaná a noví uživatelé přijdou do styku s již upravenou verzí a tudíž žádné problémy mít nebudou.

Výrazně kupředu se posunul též správce souborů. V současné chvíli již podporuje i tvorbu adresářových struktur, komplexní operace se soubory, jako jsou kopírování, přejmenování, přesunutí, smazání a také změnu kódování. O každém souboru je ve správci zobrazena informace o jeho názvu, velikosti, datumu poslední změny a kódování. Správce souborů umožňuje také snadné stažení souboru na lokální počítač nebo naopak upload souboru na server TEXonWeb. Při návrhu vzhledu správce souborů jsme vycházeli z designu uživatelského rozhraní GNOME v OS Linux, takže použití je dle našeho názoru velmi intuitivní.

3. Na čem aktuálně pracujeme

První zajímavou vlastností bude možnost verzování dokumentů, které uživatel na TEXonWeb zpracovává. Tato funkce by měla uživatelům nabídnout podobný komfort, jako například Google Documents, tedy možnost vracet se k předchozím verzím dokumentu v případě, že uživatel například provedl změny, se kterými

není spokojen. Tato funkce je založena na programu `svn`, který se běžně v dnešní době pro verzování používá. Každý uživatel bude mít při vytvoření nového účtu definován i vlastní repozitář, ve kterém se jeho soubory budou verzovat. Nebude se verzovat celý uživatelský prostor, ale pouze soubory, které uživatel explicitně vybere ve správci souborů. V menu Soubor bude pak nabídka, pomocí které bude moci uživatel verzování ovládat. V současné chvíli je tato funkce již těsně před dokončením a lze ji očekávat na TEXonWeb velmi brzy.

Co se týče práce se soubory, vyslyšeli jsme i volání mnoha uživatelů a rozhodli se prozkoumat možnosti integrace TEXonWeb a cloudových úložišť. Myšlenka je to jasná. Proč by uživatel měl mít soubory roztahané po různých účtech na různých serverech? Není lepší mít vše pohromadě a přístupné i na všech mobilních zařízeních? Rozhodli jsme se tedy, že zkusíme propojit TEXonWeb a Google Drive, který v současnosti nabízí asi nejpropracovanější API. Úložiště bude přístupné přímo, jako běžný adresář mezi ostatními soubory na TEXonWeb. Tato funkce je prozatím ve stádiu průzkumu a její nasazení si pravděpodobně ještě vyžádá delší čas.

Modernizaci čeká také vzhled uživatelského rozhraní. Po dlouhé době, kdy se design TEXonWeb prakticky nezměnil, přišel čas udělat výraznější zásah. Rozhodli jsme se využívat možností, které nabízí HTML4 a také upravit ikonky a rozmístění některých prvků tak, aby práce s TEXonWeb byla zas o něco příjemnější a jednodušší. V souvislosti s těmito úpravami hodláme vytvořit i design pro mobilní zařízení, aby uživatelé mohli TEXovat například i na tabletech, tedy na zařízeních ovládaných dotyky prstů přímo na obrazovce.

4. Co je v plánu

Poměrně zásadní věc, kterou v LATEXu považujeme za nedotaženou, je sazba tabulek. V současnosti se používá především prostředí `tabular` s různými nadstavbami pro realizaci parciálních problémů (sazba přes více stránek, různé způsoby zarovnávání údajů ve sloupcích, atd.). Tyto nadstavby jsou dostupné prostřednictvím balíčků, které jsou spolu často nekompatibilní a nelze je použít zároveň. Tuto problematiku diskutovala například Talandová (2006) ve své diplomové práci. Rovněž vytvoření komplikovanější tabulky nebo její případná úprava je poměrně složitá a pro začátečníka často neřešitelná. Tento problém se částečně snaží řešit průvodce tvorbou tabulek, který je v současné době již na TEXonWeb dostupný. Ten však umí tabulku pouze navrhnout, nikoliv však upravovat. Vyžívá také pouze základní prostředí `tabular` a některé pokročilé záležitosti tabulkové sazby jsou v něm nerealizovatelné.

Z těchto důvodů jsme se rozhodli na základě současných zkušeností připravit nové prostředí pro sazbu tabulek, které by mělo pokrýt základní běžné požadavky na tabulkovou sazbu a zároveň připravit i vizuálního průvodce, který by

začínajícím uživatelům tuto činnost výrazně usnadnil. První neúspěšný pokus realizoval již Klang (2011) ve své diplomové práci. Výsledek však trpěl mnoha neduhy a v podstatě nebyl stoprocentně použitelný. V současné chvíli pracujeme nanovém řešení, které v případě úspěchu hodláme nasadit na TEXonWeb a dát i samostatně k dispozici.

Při výuce předmětu Zpracování textů na počítači, kde studentům představujeme základy systému TEX/LATEX, získáváme poměrně hodně zpětné vazby o tom, co potenciálně uživatelům TEXonWeb při tvorbě dokumentů chybí. Uživatelé zvyklí na práci s WYSIWYG systémy jsou často lehce ztraceni v záplavě speciálních TEXových symbolů. Problémem je množství složených závorek, jejichž absence nebo nadbytek způsobuje problémy při překladu a obtížně se tyto věci také hledají v logovém souboru. Proto jsme se rozhodli vylepšit zásadním způsobem i vlastní editor kódu tak, aby se svými vlastnostmi přibližoval tomu, co známe z běžných TEXových editorů, jako jsou TeXnicCenter, WinShell nebo Vim. Mezi základní vylepšení bychom tak rádi začlenili zvýrazňování párových závorek, našeptávač základních TEXových maker a kontrolu syntaxe, ještě před překladem. Tyto vlastnosti by snad mohly pomoci začínajícím uživatelům vyhnout se překlepům v názvech značek a zapomínání párových závorek v atributech maker a při zápisu prostředí. Tato vlastnost bude samozřejmě volitelná a zkušený uživatel ji může snadno vypnout.

Mezi drobnosti, kterými chystáme vylepšit uživatelské rozhraní, patří i funkce známá například z Google Documents, a tou je automatické ukládání ozpracovaných dokumentů. V současné chvíli si uživatel musí dokumenty ukládat sám, což může vést k tomu, že uživatel z nepozornosti poslední úpravy neuloží. TEXonWeb sice uživatele upozorňuje při opuštění stránky, že může přijít o neuložené změny, ale například pád prohlížeče uživatele o poslední úpravy může připravit.

Dále plánujeme také aktualizaci anglické verze TEXonWeb, která již dnes za tou českou o něco zaostává.

5. 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 usnad-

nit 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 řesto 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 dipozici 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`.

I když je systém TEXonWeb nejčastěji prezentován mou osobou, nejsem sám, kdo se na jeho vývoji podílí. Cenné rady a nápady dodává Jiří Rybička a významný podíl na kódu má Václav Telenský.

6. Literatura

Klang, M. *Tvorba stylu pro sazbu tabulek v systému TEX/LATEX.* (Diplomová práce.) Brno : Mendelova univerzita, 2011.

Talandová, P. *Přístupy ve zpracování tabulek v systémech DTP.* (Diplomová práce.) Brno : Mendelova zemědělská a lesnická univerzita, 2006.

Summary: Switch to XELATEX and other improvements in TEXonWeb

This article describes new features of the web application TEXonWeb. This is mainly switch to UTF8 encoding and XETEX and XELATEX integration. In the article are also discussed features being developed at this time and being prepared in the near future. These functions include for example document versioning, cloud integration and other functions which should enrich and facilitate the creation of TEX documents to TEXonWeb users, mainly TEX beginners.

Novinky v projektu `bib.sty`

Tomáš Hála

Abstrakt

Projekt `bib.sty` se zaměřuje na řešení sazby bibliografických citací z hlediska nakladatelské praxe, tj. na optimalizaci sazbu a snížení náročnosti následných korektur.

V tomto článku jsou představeny novinky v uživatelském rozhraní i v implementaci, například rozšíření o struktury pro další typy bibliografických citací, jako jsou normy, mapy a zákony, volitelně aditivní chování makropříkazu pro zápis URI v případě odkazu na více internetových zdrojů či uživatelsky snadnější nastavení fontů. Bylo navrženo řešení problému zlomu standardních čísel publikací (ISBN, ISSN, ISMN) a implementováno příkazem `\typesetorbreak`. V závěru jsou nastíněny další směry vývoje.

Klíčová slova: bibliografické citace, sazební styl, `bib.sty`

Úvod

Častým problémem v ediční činnosti je sazba a jednotná úprava bibliografických citací. Projekt `bib.sty` se zaměřuje na řešení z hlediska nakladatelské praxe, tj. optimalizuje sazbu a snižuje náročnost následných korektur. Práce uživatele či sazeče tedy spočívá pouze ve volbě druhu publikace a v označkování jednotlivých atributů citace.

Množina prvků a následně jejich pořadí jsou kontrolovány automaticky právě podle zvoleného druhu publikace. Současně probíhají i interní kontroly, například na čísla stran či počet autorů.

Až do předcházející zveřejněné verze stylu `bib.sty` (2.30) byl projekt určen a vyvíjen pro nadstavbu LaTeX, představovaná nová verze (2.46[1]) je již oproštěna od specifických prvků této nadstavby, zejména čitačů, čímž se stává plně použitelnou také v nadstavbě ConTeXt.

Výsledná typografická úprava vychází ze zavedené praxe v ČR a je předdefinována. Uživatel však může si vytvořit úpravu vlastní.

Novinky v oblasti rozhraní

Mapy, normy a zákony

K existujícímu repertoáru typů publikací byly přidány tři další: `\publM` pro mapy, `\publN` pro normy a `\publZ` pro zákony citované ve zkrácené formě. Pro mapy

[1] Dostupné na `http://www.konvoj.cz/styly/biblio/2.46/bib.sty`

doi: 10.5300/2012-2/115

je zaveden nový povinný atribut `\meritko`, ostatní atributy jsou kontrolovány shodně s monografickými publikacemi. Množinu povinných atributů pro normy tvoří název, vydání a rok; zákony a další právní předpisy potřebují název a datum citace.

Nedohledatelná primární odpovědnost
Primární odpovědnost (zjednodušeně autor) je podle úzu i podle normy povinným prvkem bibliografických citací. S rozvojem elektronického publikování však řada dokumentů je zveřejňována anonymně, případně jen pod přezdívkou a není možná žádným způsobem autora či autory dohledat. Může se jednat se o různé návody, rady, informace z webových stránek apod. V některých případech lze sice připsat primární odpovědnost určité korporaci, ale nelze tak postupovat vždy. Ani označení této skutečnosti slovem *anonym* situaci neřeší, problém se pouze přesouvá do oblasti tvorby odkazů na citace. Zejména v případech tzv. jmenných odkazů s uvedením roku se náhrada chybějícího jména autora slovem *anonym* projeví zcela nepřehlednými odkazy.

Pro odlišení jména autora zapomenutého od nedohledatelného tedy nezbylo než implementovat pomocný makropříkaz `\autorne`, který nahrazuje makropříkaz `\autor{...}`. Z podobného důvodu byl přidán makropříkaz `\autordokne` pro nedohledatelného autora nadřízeného souborného dokumentu, což lze využít například u sborníků příspěvků, u nichž není uveden editor.

Nepravidelně vycházející periodika
Chování makropříkazů `\doneD` a `\doneE` bylo upraveno i pro nepravidelně vycházející periodika, tzn. i pro případy, kdy určitému dílu je přiřazeno jak ISBN, tak ISSN.

Násobné adresy
U některých citovaných zdrojů autoři uvádějí v rámci dostupnosti více internetových adres. Atribut `\www` je automaticky sázen strojopisným řezem, a proto zápis `\www{adresa1, adresa2}` produkuje čárku nesprávně strojopisným řezem. Správný výstup získáme zápisem `\www{adresa1{\zakladnifont,} adresa2}`, jedná se však o značně nepohodlný postup a navíc v rozporu s koncepcí projektu, kdy se uživatel má starat (pokud možno) výlučně o značkování a nikoliv o vizuální prezentaci, alespoň v průběhu značkování.

Proto bylo zásadní způsobem změněno chování atributu `\www` z běžného na volitelně aditivní, což znamená, že opakovaným použitím v rámci jednoho citačního záznamu se hodnoty nepřepisují, ale řetězí do seznamu. Uživatel aktivuje tuto vlastnost nastavením `\additivetrue`.

Novinky v implementaci

Podpora pro ConTeXt
Původně byl styl `bib.sty` určen a vyvíjen pro nadstavbu LaTeX. Autor tohoto článku však začal při vlastní práci pociťovat některé nedostatky LaTeXu, a proto před několika lety vyzkoušel ConTeXt. Bohužel LaTeXová konstrukce některých prvků, např. výčtové prostředí pro sazbu seznamu citací a čitače, nedovolovaly použít tento styl. Implementace byla proto kompletně revidována: LaTeXové čitače byly nahrazeny čitači základního TeXu, výčtové prostředí a volba řezů písma se nastavuje automaticky podle zvolené nadstavby. Po zavedení nové „logické proměnné“ (`\newif\ifConTeXt`) můžeme toto rozhodnutí učinit ověřením existence `\contextversion` – pochopitelně za předpokladu, že si tento makropříkaz uživatel LaTeXu nedefinoval soukromě:

```
\ifx\contextversion\undefined\else\ConTeXttrue\fi
```

Aktuální verze (2.46, listopad 2011) je tedy použitelná jak v nadstavbě LaTeX, tak také v nadstavbě ConTeXt.

Datum citace
Byla doplněna kontrola data citace uplatněná u on-line dokumentů a u zákonů citovaných ve zkrácené formě.

Údaje o vydání
Původně byl údaj o vydání implementován jako povinný údaj. Vydavatelů ovšem často u prvního vydání tento údaj neuvádějí, v takovém případě jej citace rovněž neobsahuje. Proto byl tento atribut o vydání změněn na nepovinný a jeho absence se nekontroluje.

Standardní čísla
Původní zobrazení atributů pro ISBN, ISSN a ISMN bylo doplněno o možnost nastavení závorek, což u článků a příspěvků, pokud je tento údaj, ač nepovinný, vyžadován.

Volba fontů
Všechny fonty lze nyní uživatelsky nastavit. Přehled příkazů, které může uživatel použít, a jejich implicitní nastavení uvádí tabulka.

příkaz	význam a použití	ConTeXt	LaTeX
`\fontautor`	autor díla; autor či editor souborného díla	`\sc`	`\scshape`
`\fontautordod`	dodatky k autorům („a kol.“, „ed.“)	`\sc`	`\scshape`
`\fontautorkorp`	název korporace v místě autora	`\tf`	`\rmfamily\upshape`
`\fontcitace`	základní font pro citaci	`\rm\tf`	`\mdseries\upshape`
`\fontdostupnost`	text „Dostupné na:“	`\rm\tf`	`\mdseries\upshape`
`\fontonlinetext`	text „on-line“	`\rm\tf`	`\mdseries\upshape`
`\fontnazev`	název díla, název celku	`\it`	`\itshape`
`\fonturi`	URI	`\rm\tt`	`\mdseries\ttfamily`
`\fontmeritko`	měřítko	`\rm\tf`	`\mdseries\upshape`
`\fontISNtext`	zkratky standardních čísel	`\rm\sc`	`\rmfamily\scshape`

Řešené problémy

V průběhu implementace nastaly dva problémy zasluhující zmínku – sazba URI a standardních čísel publikací.

URL
Prvním z nich je odlišná implementace a z ní vyplývající odlišné chování makropříkazu `\url`. V LaTeXu se makropříkaz chová podle očekávání, avšak v ConTeXtu láme v prvním přípustném místě až *za* pravým okrajem zrcadla, což za všech způsobí okolností nepříjemné přetečení sazby. Problém byl oznámen hlavnímu tvůrci ConTeXtu Hansi Hagenovi, který přislíbil opravu.

Problém zlomu standardních čísel publikací
Standardní čísla publikací by měla tvořit jeden celek, což znamená, že by nemělo docházet k řádkovému zlomu ani mezi zkratkou (např. ISBN) a vlastním kódem, ani v rámci kódu, který obsahuje spojovníky, jež jsou algoritmem řádkového zlomu považovány za velmi dobrá místa zlomu.

Pokud potlačíme tyto dělicí body, např. nezlomitelnou mezerou za zkratkou či jinými další prostředky v rámci kódu), algoritmus chápe celý atribut jako jeden celek. Nenachází-li se na daném řádku dostatek místa pro tak velký element, algoritmus zlomu určí vhodné místo *před* tímto elementem a důsledkem jsou pak příliš roztažené mezery mezi jednotlivými objekty takového řádku. Nejmenším zlem v případě, kdy se element se standardním číslem nevejde na řádek, se ukázalo vyvolání „násilného“ řádkového zlomu s tím, že odstavec s bibliografickou citací nebude mít předposlední řádek zarovnán na blok.

K rozhodnutí, zda pro element máme k dispozici dostatek místa či nikoliv, je potřeba znát šířku posledního, tj. právě sázeného řádku. Olšák (1997, s. 203) připomíná, že systém TeX nedisponuje žádným jiným nástrojem umožňujícím tuto hodnotu zjistit kromě registru `\predisplaysize` z display módu, a uvádí postup zpracování této hodnoty. Tento postup se stal základem pro makropříkaz `\typesetorbreak`:

```
\def\typesetorbreak#1{\setbox0=\hbox{#1}\lastlinewidthfree=\textwidth
  \ifhmode $$ \advance\predisplaysize by -3.6667em
  \global\lastlinewidth=\predisplaysize  \predisplaysize=\maxdimen
  \abovedisplayskip=-\baselineskip       \belowdisplayskip=0pt $$\fi
  \vskip-\baselineskip                   \hskip\lastlinewidth
  \advance\lastlinewidthfree by -\lastlinewidth
  \ifdim\wd0>\lastlinewidthfree\crlf\fi#1
}
```

Nástin dalšího vývoje

V současné době se velmi často užívají tzv. čísla DOI (Digital Object Identifier), blíže viz IDF (2013). Další atribut (`\doi`) a jeho začlenění do výsledného tvaru bibliografických citací se ukazuje jako žádoucí.

Dále je plánování přidání nepovinného atributu pro tzv. podřízená odpovědnost, tedy další autory, kteří se podíleli na díle (ilustrátoři, překladatel apod.). Toto bod si však vyžádá hlubší rozbor, neboť bude potřeba rozhodnout o pořadí v případě požadavku na více různých podřízených odpovědností.

V rámci další optimalizace sazby a následných korektur bude určitě stát za úvahu podpora automatického seřazení podle prvního prvku citace v případě jmenných odkazů, případně alespoň kontrola správnosti uspořádání.

Krátce před dokončením tohoto textu vstoupila v platnost nová verze normy ČSN ISO 690:2011, nahrazující předchozí dvoudílnou normu (ČSN ISO 690:1996, ČSN ISO 690-2:2000). Jedná o normu převzatou překladem, bez národní přílohy a s řadou podrobností, které přesahují praktické potřeby v akademické sféře i v nakladatelské praxi. Určitým přínosem je podrobné vymezení různých kategorií citovatelných dokumentů, podobně jako tomu bylo před rokem 1990. Užitečné prvky této normy poslouží k rozšíření možností projektu `bib.sty`.

Reference

ČSN ISO 690 Informace a dokumentace – Pravidla pro bibliografické odkazy a citace informačních zdrojů. Praha : Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2011, 39 s.

ČSN ISO 690 Informace a dokumentace – Pravidla pro bibliografické odkazy a citace informačních zdrojů. Praha : Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 1996, 39 s.

ČSN ISO 690 Informace a dokumentace – Pravidla pro bibliografické odkazy a citace informačních zdrojů. Praha : Úřad pro technickou normalizaci, metrologii a státní zkušebnictví, 2000, 39 s.

International DOI Foundation (IDF). *The DOI® System* [on-line]. 2011. [cit. 2011-09-25]. Dostupné na: `www.doi.org`.

Olšák, Petr. *TEXbook naruby.* 1. vyd. Brno : Konvoj, 1997. 466 s. ISBN 80-85615-64-9.

Summary: New Features in Project `bib.sty`

Project `bib.sty` focuses on typesetting of references from publishers point of view, ie. on optimisation of typesetting process and on lowering of its laboriousness.

This paper presents new features in the user interface as well as in the implementation, eg. new structures for norm, maps, and laws; optionally additive behaviour of macro for URI if more links are used; user-friendly font settings. The solution of linebreaks at standard book numbers has been proposed and implemented by the command `\typesetorbreak`. Finally, new suggestions for the futher development have been outlined.

thala@mendelu.cz,
Mendelova univerzita v Brně, Provozně ekonomická fakulta,
ústav informatiky, Zemědělská 1, CZ 613 00 Brno,
KONVOJ, spol. s r. o., Bystřinova 4, 612 00 Brno, *konvoj@konvoj.cz*

Softwarová podpora korektur interpunkce

Tomáš Hála, Gita Urbanová

Abstrakt:

Článek se zabývá možnostmi automatizace korektur interpunkčních jevů. Nejprve byly srovnány možnosti existujícího programového vybavení. Dále byla stanovena míra chybovosti při psaní interpunkčních jevů (2,35 % pro ekonomické texty, 0,96 % pro texty z oboru informatika). Celkově vysoká hodnota vedla k rozhodnutí implementovat jednoduchý korektor těchto jevů. Tomu předcházelo určení množiny jevů vhodných pro automatickou korekci a množiny jevů, které lze detekovat, ale již ne jednoduše automatizovaně opravit.

Klíčová slova: korektura, interpunkční znaménka, písařské chyby, zjištění chybovosti, automatická oprava, detekce nejednoznačných jevů

Úvod

Korektura patří mezi neodmyslitelné součásti procesu vzniku kvalitní tiskoviny či elektronické prezentace. Jejím smyslem je odhalení pravopisných, typografických, případně i věcných chyb v dokumentu.

Jednou z oblastí, ve které se chybuje, je interpunkce. Chyby v psaní interpunkčních znamének mohou být způsobeny jak neznalostí autora, tak – což je častější – nepozorností při sestavování textu.

Před předáním na korektury je vždy lepší se pokusit odstranit co nejvíce případných chyb automatizovaným způsobem, aby se korektor-člověk mohl soustředit na ostatní jevy.

Proto jsme se rozhodli prozkoumat jednak možnosti automatizace korektur interpunkčních jevů v existujícím programovém vybavení, jednak určit míru chybovosti, která by posloužila k rozhodnutí, zda se vyplatí korektor implementovat.

Problémy s psaním interpunkce nebo s překlepy v interpunkci se objevují nejen v obyčejných dokumentech, ale také v pracích vědeckých, o což jsme se opřeli v další úvaze.

Programové vybavení pro korektury textu

Automatizované korekce provádí programy nazývané textové korektory. Ty se vyskytují jako součást textového editoru či procesoru, či jako samostatné programy pro korekci dokumentu.

 doi: 10.5300/2012-2/120

Textové procesory v kancelářských balících obsahují rozsáhlé možnosti a zjednodušení při pořizování textového dokumentu. Například korektor implementovaný v programu MS Word se člení do několika samostatných oddílů – automatické opravy textu, kontrola pravopisu, kontrola gramatiky a nástroj Tezaurus, který napomáhá nalezení správného slova. (Mansfield, 1998) Opravami interpunkce se přímo nezabývá, má však implicitně aktivovány některé další funkce, mezi něž patří například změna na velká písmena na začátku vět, což se ovšem může negativně promítnout v situacích, kdy tečka neoznačuje konec věty ale zkratku či iniciály. Tento repertoár obsahují všechny verze od „97“ do současnosti. Dále je zde implementována opravu počátečník uvozovek ("→„).

Kancelářský balík LibreOffice (2010) nabízí v rámci automatických oprav odstranění dvojitých mezer, náhradu spojovníku a sekvence dvou spojovníků za půlčtverčíkovou pomlčku (též v MS Word) a rovněž odstranění mezer a tabulátorů na začátcích a koncích řádků a odstavců. Bez nastavování nahrazuje sekvenci tří teček výpustkem. Dalšími jevy z oblasti interpunkce či sazby značek se nezabývá.

V kancelářských balících lze dále využít funkci automatických náhrad, ty si však uživatel musí definovat sám.

Unixové programy, sloužící často k psaní a úpravě textu (`joe`, `vim` apod.), byly v souladu se základní unixovou zásadou vývoje OS Unix (*psát programy, které budou dělat právě jednu věc, a tu budou dělat dobře*; Brandejs, 2003) konstruovány pouze pro tyto editační účely a další funkce neobsahují.

Příkladem programu ctícího tuto zásadu může být program `vlna` (Olšák, 2002, 2010). Jedná se o jednoúčelový malý program, který v dokumentu vyhledá všechny jednopísmenné předložky a spojky, za něž vloží místo obyčejné mezery mezeru nezlomitelnou.

Ze stejného důvodu jsou v OS Unix opravy řešeny samostatným programem `aspell`, korektor však je určen pro opravy překlepů, interpunkcí se stejně jako dosud zmíněné programy nezabývá.

Jediným programem, který sloužil v minulosti ke sledovanému účelu, byl program `ikor` (Rybička, [1997]). Jednalo se o interaktivní korektor písařských chyb. Korektor byl založen na lexikální analýze vybraných atomů a na stanovení jejich pořadí, které bylo podrobeno analýze vhodnosti. Písařskou chybou se zde rozumělo nesprávné pořadí interpunkce a mezerování. Zjištěné nesrovnalosti program zapisoval do změnového souboru, z něhož se ve druhé fázi čerpaly informace pro vlastní korekci textu. Program byl vytvořen pro OS DOS a není veřejně dostupný.

Za zmínku stojí také program `lacheck`, určený k předběžné kontrole dokumentů napsaných v systému LaTeX. I přesto, že jeho nejvýznamější praktický přínos spočívá v kontrole párovosti závorek všech druhů včetně kontroly párů příkazů pro sázecí prostředí, je schopen odhalit nesprávné mezerování, doporučit použití výpustků apod. Jedná se však pouze o detekci velmi malé množiny jevů. Kromě toho jeho užití pro jiné formáty je dosti omezené.

Interpunkcí se zabývá také program Grammaticon (Lingea, 2012). Jedná se o komplexnější, interaktivní řešení odhalující – kromě řady gramatických jevů – chybnou interpunkci a chybějící nebo nadbytečné čárky ve větách. Program Grammaticon však bohužel patří mezi komerční produkty.

Shrneme-li situaci ve zkoumaném programovém vybavení, musíme konstatovat, že neexistuje žádný produkt, který by se zabýval komplexně detekcí a případnou korekcí písařských chyb a současně nebyl (výlučně) interaktivní, dále aby byl dostupný a volně šiřitelný.

Hodnocení chybovosti v odborných textech

Na základě empirických zkušeností se sazbou a korekturami textu bylo zřejmé, že určité množství chyb je způsobnosti písařskými nepřesnostmi v oblasti interpunkčních znamének. Bylo však potřeba toto množství kvantifikovat, aby bylo možné rozhodnout, zda se vyplatí korektor implementovat.

Pro posouzení chybování byl použit náhodný výběr původních vědeckých prací ekonomických oborů (21 textů) a vědeckých pracích oboru informatika (16 textů). Oba obory byly hodnoceny odděleně. Ve všech případech se jednalo o rukopis, tedy ještě nekorigovaný dokument.

Každá práce byla hodnocena týmž způsobem, zpracován je pouze český nebo slovenský text. Vložené grafy, tabulky, obrázky a jejich popisky stejně jako závěrečné autorovy osobní údaje nebyly pro statistiku chybování v interpunkci uvažovány. Slovenské texty jsme z hodnocení záměrně nevyloučili, neboť pravidla slovenské interpunkce jsou téměř shodná s českými.

V každém textu byly zvlášť posouzeny jednotlivé jevy. Podrobné tabulky četností členěné podle souborů a jevů prezentovala Urbanová (2003). Zde uvádíme pouze souhrnné výsledky.

Součástí kvalitního zpracování dokumentu by měly být nástroje, které spolehlivě eliminují výskyt chyby, jak gramatické, tak případně typografické či jiné. Každá odborná práce by ve své výsledné podobě obsahovat chyby neměla.

Na výsledné hodnotě statistiky původních vědeckých prací ekonomických je však znát, že takové bezchybné úrovně zdaleka nedosahují. Kontrola chybovosti v těchto textech byla provedena pouze pro interpunkční znaménka. Podíl chybně zapsaných interpunkčních jevů dosáhl u ekonomických textů 2,35 %, což je na tak odborný materiál dosti značné, a lze soudit, že nástroj k odstranění nebo alespoň snížení procenta chyb by v takovém případě měl své opodstatnění.

Statistické zpracování souboru 16 náhodně zvolených prací v oboru informatika ukázalo, že procentní chybovost v interpunkci činila celkově 0,96 %, což je lepší hodnota než u souboru prací ekonomických. Při kontrole chybování v interpunkci byla aplikována stejná pravidla jako pro práce ekonomické.

Za chyby byla považována znaménka chybně oddělená od ostatního textu, také znaménka chybějící ve smyslu uvozovek či závorek, které se musí vysky-

	obory ekonomické			*obor informatika*		
jev	výskytů celkem	četnost chyb	chybovost [%]	výskytů celkem	četnost chyb	chybovost [%]
tečka	3 143	129	4,10	1 728	22	1,27
čárka	3 125	21	0,67	1 862	12	0,64
uvozovky	212	4	1,89	154	2	1,30
pomlčka	392	5	1,28	284	1	0,35
závorka	816	20	2,45	588	3	0,51
výpustek	12	2	16,67	11	2	18,18
dvojtečka	400	10	2,50	224	4	1,79
středník	29	0	—	18	0	—
otazník	12	0	—	19	1	5,26
vykřičník	0	—	—	0	—	—
odsuvník	0	—	—	0	—	—
celkem	8 141	191	2,35	4 888	47	0,96

Tabulka 1: Hodnocení chybování v interpunkci

tovat v páru. Znaménka, která chybí, například z důvodu neznalosti českého pravopisu nebo pouhého překlepu uvažována nejsou, neboť zkoumán byl pouze zápis interpunkce, nikoliv její správné jazykové použití.

Ve sledovaných pracích se vyskytovala i chybně psaná sousedící interpunkční znaménka. V takovém případě bylo nutné rozhodnout, kterému jevu tato chyba náleží. Například ve spojení: „zima.(chladno)“ chybí mezi tečkou a závorkou mezera. Je tato chyba způsobena chybějící mezerou za tečkou, nebo mezera chybí před závorkou? Chybné psaní bylo správně připsáno závorce, neboť tečka ve své definici obsahuje, že mezerou je zprava oddělována, jestliže se za ní nevyskytuje jiné interpunkční znaménko. Naproti tomu před otevírající závorkou se mezera vyskytuje vždy, pokud neleží na počátku řádku.

Sestavení množiny jevů vhodných pro automatickou korekci a pro detekci

Uvedená interpunkční znaménka spolu s jejich pravidly pro psaní dělíme na dvě skupiny: Znaménka, která lze řešit automatickou korekcí a znaménka, na jejichž chybný zápis reagují varovná hlášení.

Automatická korekce je vhodná pro znaménka, která neobsahují nejednoznačná a na kontextu závislá pravidla pro psaní interpunkce. Patří mezi ně tečka, před kterou se v českém jazyce nikdy nevyskytuje mezera a za níž se mezera naopak povinně vkládá, jestliže nenásleduje jiná interpunkce. Takto automatizovaně lze korektury provést také u středníku, vykřičníku, otazníku, závorek, odsuvníku a uvozovek.

Dvojtečka je automaticky řešena jen ve vztahu k písmenům abecedy, neboť u dvojtečky mezi čísly dochází k jisté nejednoznačnosti. Nevíme totiž, zda se jedná o matematické dělení, nebo číselný poměr (odhlížíme od poměru vyjádřeného písmeny, např. a:b). Stejně tak u čárky není jasné, zda se jedná o výčet čísel oddělený čárkami, nebo o desetinné číslo. Také zde korektor rozhoduje automatizovaně jen v okolí písmen české abecedy.

Jako další případ lze uvést apostrof, vyznačující v lingvistických textech palatizovanou souhlásku. Proto může tudíž stát i na začátcích slov, např. 'sloboda (Pravidlá slovenského pravopisu, 1991). To platí i pro češtinu, přestože Pravidla českého pravopisu tento případ výslovně nezmiňují.

Taktéž lomítko píšeme podle kontextu s mezerami (verše), bez mezer (např. vyjádření alternativ: ano/ne) nebo „střídavě“ /náhrada závorek/.

Pouze *detekce* je aplikována na interpunkci, která se řídí pravidly závislými na kontextu. K takovým jevům náleží výpustek, lomítko a pomlčka. Součástí detekce jsou varovná hlášení, upozorňující například na lichý počet uvozovek v souboru nebo na zápis více desetinných čárek mezi čísly.

Implementace korektoru české interpunkce

Korektor české interpunkce (Urbanová, 2003) byl sestaven v jazyce Perl s využitím regulárních výrazů, které jsou účinným nástrojem pro zpracování řetězců v textových souborech. Proto se pomocí nich velmi dobře vyhledávají interpunkční znaménka a zkoumají korektní zápisy identifikovaných jevů.

Každý jev obsluhovaný tímto korektorem interpunkce je implementován jako samostatná procedura, aby bylo možné volit potřebné jevy a případně jevy nežádoucí vypustit. Příkladem nežádoucí korekce může být například úprava u kulatých závorek, užívané jako součást chemických vzorců, u komplexních sloučenin je pak nežádoucí i korekce hranatých závorek.

Jako další případ lze uvést apostrof, vyznačující v lingvistických textech palatizovanou souhlásku. Proto může tudíž stát i na začátcích slov, např. 'sloboda (Pravidlá slovenského pravopisu, 1991). To platí i pro češtinu, přestože Pravidla českého pravopisu tento případ výslovně nezmiňují.

Taktéž lomítko píšeme podle kontextu s mezerami (verše), bez mezer (např. vyjádření alternativ: ano/ne) nebo „střídavě“ /náhrada závorek/.

Řízení korekce

Korektor realizuje opravy výše uvedených jevů a umožňuje řídit působení jejich korekce pomocí direktiv v textu. Jevy zahrnuté v korektoru jsou řízeny procedurami samostatně, aby bylo možné přidělit každému jevu vlastní direktivu.

Tento přístup byl inspirován programem `vlna` (Olšák, 1995–2010), kde vkládání nezlomitelných mezer lze vypnout direktivou `%~-` a zapnout analogicky

direktiva	význam	direktiva	význam
T	tečka	H	závorky hranaté
C	čárka	K	závorky kulaté
S	středník	Z	závorky složené
D	dvojtečka		
O	otazník	M	redukce mezer
V	vykřičník	L	lomítko
U	uvozovky		
A	apostrof	E	zahrnuje všechny opravy

Tabulka 2: Seznam direktiv pro řízení korektury interpunkce

pomocí `%~+`. V nežádoucích místech, například v matematických prostředích či v prostředí `verbatim`, se tak automatické opravy neprovádějí a text zůstane nedotčen.

Pro tento korektor jsme jako prefix zvolili `%!`, za nímž následuje jednopísmenná direktiva se znaménkem `+` nebo `-`. Seznam direktiv, obsažený též v nápovědě k programu, uvádí tabulka 2.

Požadované nastavení je také možné načíst z konfiguračního souboru.

Varovná hlášení
Snahou korektoru je také v některých případech, kde nelze u nalezených jevů jednoznačně rozhodnout, alespoň vypsat varovná hlášení. Hlášení vypisuje korektor v těchto případech:

- v souboru objevil lichý počet uvozovek;
- konfigurační soubor obsahuje neznámý symbol (pro jistotu následuje volání funkce `die`);
- v souboru se vyskytla pomlčka ohraničená mezerou jen z jedné strany;
- v souboru se vyskytují dvě nebo čtyři po sobě jdoucí tečky;
- soubor obsahuje chybně zapsané desetinné číslo, např.: 1,223,4.

Závěr

Automatizované korektury přináší uživateli mnohá zjednodušení a krácení času stráveného zpracováním textů. Většina textových editorů či procesorů obsahuje korektor pravopisu, některé i korektor gramatiky, nikde však se nenachází k dispozici nástroj k opravě písařských chyb v interpunkci. Ani nástroj automatických oprav uživateli neposlouží, neboť by uživatel si musel všechny sledované jevy či chyby sám vložit.

Sledováním četnosti různých typů chyb způsobených špatným psaním interpunkce bylo zjištěno, že písařské chyby v oblasti interpunkce nejsou zdaleka zanedbatelné. Proto byl implementován jednoduchý korektor.

Představená verze korektoru české interpunkce uživateli zcela poslouží v případě zpracování holého textu. Zpracování dokumentů značkovaných v imple-

mentacích jazyka TEX vyžaduje potlačení oprav okolo složených závorek. Do budoucna je možné uvažovat rozšíření na další formáty, např. HTML.

Při zpracování odborných dokumentů je však třeba mít na paměti, že automatizované opravy některých znamének by mohly být spíše na škodu, některé případy již byly zmíněny.

Korektor byl vytvořen primárně pro optimalizaci zpracování textů v češtině, lze jej však využít i pro zpracování textů ve slovenštině, neboť pravidla psaní interpunkčních znamének se v těchto dvou jazycích neliší.

Program se nachází na adrese `www.korektury.cz/software/punch` a je volně ke stažení.

Reference

Brandejs, Michal. *Linux : Praktický průvodce.* 2. vyd. Brno : Konvoj, 2003. 312 s. ISBN 80-7302-050-5.

Grammaticon [on-line]. [s. a.]. [cit. 2012-06-25]. Dostupné na: `http://www.lingea.cz/grammaticon.htm`.

lacheck v. 1.26 [software]. 1998.

LibreOffice 3.3 [software]. c2000, 2010.

Mansfield, Ron. *Word 97.* 1. vyd. Grada Publishing : Praha, 1998. 887 s. ISBN 80-7169-517-3.

Olšák, Petr. *vlna* v. 1.2, 1.5 [software]. 2002, 2010.

Pravidlá slovenského pravopisu. 1. vyd. Bratislava : Slovenská akadémia vied, 1991. 536 s. ISBN 80-224-0080-7.

Rybička, Jiří. *ikor* [software]. [1997].

Urbanová, Gita. *Implementace korektoru české interpunkce.* (Bakalářská práce.) Brno : Mendelova zemědělská a lesnická univerzita, 2003. 35 s.

Summary: Software Support for Punctuation Proof

This paper is focused on possibilities of automating corrections of punctuation mistypings. First, functionality of current software tools is compared. Second, the error rate of punctuation mistypings is determined (2.35% and 0.96% for economics and informatics texts, respectively). These high values are a compelling reason to implement a simple program for correcting such mistypings. Also, we identify mistypings suitable for automatic corrections as well as those that can be detected but are not easy to automatically correct.

Key words: proof, punctuation marks, mistyping, error rate, automated correction, detection of ambiguous phenomena

thala@mendelu.cz

Mendelova univerzita v Brně, Provozně ekonomická fakulta, ústav informatiky, Zemědělská 1, CZ 613 00 Brno

Zpráva z konference 5th ConTEXt meeting 2011

Tomáš Hála

Od roku 2007 se každoročně konají setkání uživatelů typografické nadstavby ConTEXt nazvaná ConTEXt meeting. Do širšího povědomí se tato setkání mohla dostat v roce 2010, když Československé sdružení uživatelů TEXu uspořádalo svoji konferenci TEXperience společně právě s jedním s těchto setkání.

Pro další ročník setkání uživatelů ConTEXtu pořadatalé vybrali malé belgické městečko Bassenge. Vzdálenost nemohla překonat touhu po setkání, a tak jsme v pondělí časně ráno společně s Lukášem Procházkou (řidič) a Janem Kulou vyrazili na celodenní jízdu na toto páté setkání.

Po ubytování v malém soukromém zámečku se účastníci v pondělí večer sešli ke krátkému neformálnímu setkání. Během něj se Haraldu Königovi, skutečnému králi Linuxu, podařilo po urputném boji oživit stářím unavený (školní) notebook autora těchto řádků, čímž byl zachráněn úterní dopolední přednáškový program.

V něm jako první vystoupil Tomáš Hála (ČR) a podělil se s přítomnými, ač dlouholetý „LaTeXista“, o svoje první zkušenosti s ConTeXtem v příspěvku nazvaném *Migration to* ConTeXt*? First experience with* ConTeXt *typesetting.* Nejprve byly srovnány nadstavby LaTeX a ConTeXt z hlediska uživatele, tedy rychlost překladu, chybová a varovná hlášení a kompletní protokol o překladu (logfile). Některá hlášení jsou mírně zavádějící, neboť odkazují – podobně jako v LaTeXu – na vnitřní, uživateli neznámé makropříkazy.

Hlavním důvodem, proč autor ConTeXt vyzkoušel, je bezesporu vyřešený problém sazby na řádkový rejstřík, čehož lze snadno dosáhnout nastavením jednoho parametru při definici layoutu. K dalšími zmíněným výhodám patří celkově jednodušší způsob nastavení všech rozměrů sazebního obrazce a dalších parametrů sazby, stejně jako snadná aktivace visící interpunkce. I koncepce příkazů – obvykle trojice `\setup`*něco*, `\start`*něco* a `\stop`*něco*, doplněná o rozlišování nastavovacích parametrů (v hranatých závorkách) od sázených parametrů (ve složených závorkách) vytváří pro uživatele příjemnější prostředí a také zlepšuje čitelnost kódu.

Celkově byla zmíněna celá řada problémů, návrhů řešení i pocitů, z nichž některé posloužily vývojovému týmu jako zpětná vazba. Mezi takové záležitosti lze zařadit například nedokončené nastavení pro slovenštinu, ovšem na druhou stranu se ukázalo, že v ConTeXtu i začátečník jen s pomocí dokumentaci dokáže nastavení přizpůsobit svým potřebám.

Thomas Schmitz (Německo) v příspěvku *Typesetting and Edited Volume with* ConTeXt popsal svoje zkušenosti s ConTeXtem při práci na kolektivně zpracovávaném sborníku pro jisté německé nakladatelství. Ukázal nejen jak zpracovat takový sborník jako ConTeXtový projekt, ale také zaměřit na proces zpracování v kolektivu uživatelů s různým stupněm počítačových dovedností.

Další blok byl věnován MetaPostu. Taco Hoekwater (Nizozemí) zaměřil své vystoupení *MetaPost 1.750: Numerical engines* na pokroky v přípravě knihovny podporující MetaPost 2. Hlavním řešeným problémem byla absence zpracování čísel s plovoucí řádovou čárkou. Dosavadní stav je totiž postaven na zlomcích 32bitového celého čísla, což způsobuje pozorovatelnou nepřesnost při výpočtech. MetaPost 2 bude pokrývat čtyři možné způsoby zpracování desetinných čísel, a to *scaled 32-bit* kvůli zpětné kompatibilitě, *IEEE floating point*, což odpovídá typu double, *MPFR* – binární zobrazení s libovolnou přesností, a *decNumber* – desítkové s libovolnou přesností. Autor zabýval souvislostmi a problémy, které tyto změny v implementaci vyvolají.

Ve svém dalším vystoupení *Using large numbers in MetaPost* prezentoval řešení implementace velkých čísel.

(dokončení příště)

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

Vydalo:	Československé sdružení uživatelů TEXu vlastním nákladem jako interní publikaci
Obálka:	Antonín Strejc
Ilustrace na obálce:	Ondřej Krybus
Počet výtisků:	400
Uzávěrka:	15. 4. 2012
Odpovědný redaktor:	Zdeněk Wagner
Redakční rada:	Zdeněk Wagner (šéfredaktor), Ján Buša, Jiří Demel, Tomáš Hála, Jaromír Kuben, Michal Růžička, Jiří Rybička, Petr Sojka, Pavel Stříž, Jan Šustek
Technický redaktor:	Tomáš Hála
Tisk a distribuce:	KONVOJ, spol. s r. o., Berkova 22, 612 00 Brno, tel. +420 541 245 548
Adresa:	CSTUG, c/o FEL ČVUT, Technická 2, 166 27 Praha 6
E-mail:	cstug@cstug.cz

Zřízené poštovní aliasy sdružení CSTUG:

`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 CSTUGu

`secretary@cstug.cz`, `orders@cstug.cz`
korespondence administrativní síle sdružení, objednávky CD a DVD

`cstug-members@cstug.cz`
korespondence členům sdružení

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

`bookorders@cstug.cz`
objednávky tištěné TEXové literatury na dobírku

ftp server sdružení:
`ftp://ftp.cstug.cz`

webový server sdružení:
`http://www.cstug.cz`

CONTENTS