

Zpravodaj Československého sdružení uživatelů TeXu

Tomáš Hála

Tabulky v ConTeXtu: přístupy, možnosti, algoritmy

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 29 (2019), No. 1-4, 24–43

Persistent URL: <http://dml.cz/dmlcz/150168>

Terms of use:

© Československé sdružení uživatelů TeXu, 2019

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

Sazba tabulek trvale patří mezi obtížnější prvky při zpracování publikací. Příspěvek přináší přehled možností sazby tabulek v systému CONTEXtu. Zabývá se srovnáním dostupných způsobů, starších i současných, konkrétně se jedná o prostředí `table`, `tabulate`, `TABLE`, `xtables`, srovnává jejich možnosti vzájemně i s „konkurenčním“ L^AT_EXem.

Tabulky mohou být i generovány z dat v jiném formátu, velmi často z formátu CSV. Proto se příspěvek dále zabývá i možnými přístupy tzv. databázového zpracování.

Dále budou předvedeny některé jednoduché algoritmy, kterými lze snadno rozšířit stávající možnosti. Algoritmy budou prezentovány v jazyce Lua, jenž je součástí systému CONTEXt MkIV.

Klíčová slova: CONTEXt, L^AT_EX, srovnání, sazba, tabulky, `table`, `tabulate`, `TABLE`, `xtables`, CSV

Úvod

Srovnáme-li nástroje sazby tabulek napříč implementacemi T_EXu, uvidíme značně odlišné přístupy. Základní plain nedisponuje specializovanými prostředky pro sazbu tabulek, což ostatně vyplývá i ze skutečnosti, že „bible“ T_EXu (Knuth, 1986) se samotnou sazbou tabulek zabývá až v kapitole 22 nazvané Alignment, a to v souvislosti s obecnou problematikou zarovnání sazebního materiálu.

Způsob zápisu v plainu – i přesto, že je ve své podstatě přímočarý a zcela logický – patrně nikdy nebude prohlášen za uživatelsky přívětivý, neboť vyžaduje poněkud hlubší znalost systému a jeho chování.

Požadavek na tvorbu jednoduššího uživatelského prostředí vyústil v 80. letech minulého století vznikem nadstavby L^AT_EX, která pro tabulkově orientovanou sazbu používá dvě základní prostředí – `tabbing` napodobující systém tabeláčnických zářezek psacího stroje, jež je vhodné pro pořadovou sazbu, a `tabular` pro běžné tabulky.

Od 90. let minulého století je k dispozici další nadstavba – CONTEXt. Jedná se, stejně jako u T_EXu samotného, o volně šiřitelný software určený pro sazbu dokumentů s požadavkem vysoké kvality výstupního produktu. Základní vlastnosti verze MkIV této nadstavby byly představeny již dříve (T. Hála, 2013 a 2015).

ConTeXt: základní tabulky – table

Nejstarším způsobem sazby tabulek je prostředí `table`. Ve srovnání s \LaTeX em má prostředí `table` nejbližší ke známému prostředí `tabular`. Namísto znaku `&`, který v \LaTeX u slouží k oddělování sloupců, zde příkazem `\NC` zahajujeme sazbu nového sloupce (`NC` je zkratka pro `new column`, nový sloupec).

Definice sloupců

K určení vlastností jednotlivých sloupců jsou k dispozici kódy podobné prostředí `tabular`, pro sloupec lze však použít i více než jeden kód. Aby nedošlo k chybné interpretaci, svislá čára – v \LaTeX u používaná k aktivaci svislých linek – zde slouží pouze k oddělení definic jednotlivých sloupců. Kromě obdobných kódů jako v \LaTeX u (`c`, `l`, `r`, `p`) lze zadat i informaci o vyrovnání sloupců (`s0`, `s1`, `s2`, `s3`), kde číslo je faktorem velikosti mezisloupcových mezer. Zde uvedené příklady se vyrovnáním sloupců v tomto prostředí nezabývají.

Ač historicky nejstarším, je prostředí `table` dosud funkčním způsobem sazby tabulek. Četnost jeho užití se však dnes již snižuje, vyspělejší prostředky (`TABLE`, `xtables`) nabízejí pohodlnější správu tabulek i s přístupem k jednotlivým buňkám. Užití prostředí `table` ilustruje následující kód s výstupem na obrázku 1.

```
\starttable[|r|p(0.6\textwidth)|]\HL
\NC Podle úpravy:
    \NC pořadově sázené, otevřené, uzavřené,
      s členícími linkami\,\dots
    \MR
\NC Podle účelu:
    \NC formuláře, bilance, knižní, časopisecké\,\dots
    \MR\HL
\NC Podle nástroje:\NC ~\FR\HL[2]
\NC \LaTeX: \NC tabbing\MR
\NC \NC tabular + halda balíčků\MR\HL
\NC \ConTeXt: \NC tabulate\MR
\NC \NC table, TABLE, xtables:\MR\HL
\NC Lua:\NC neomezené možnosti:\MR\HL[3]
\stoptable
```

Vyrovnávání řádků

Ukázka z obrázku 1 nemá správně vyrovnán první řádek, neboť všechny řádky jsou vyrovnávány příkazem `\MR`, sloužícím jako pro horní, tak pro dolní vyrovnání běžného řádku. Jiné způsoby vyvoláme příkazy `\SR`, rovněž horní i dolní, ale s využitím u oddělovacích řádků, `\FR` pouze pro horní vyrovnání, `\LR` pouze pro dolní vyrovnání a `\NR` bez vyrovnání (Otten a Hagen, 2006).

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje:	
L ^A T _E X:	tabbing tabular + halda balíčků
ConT _E Xt:	tabulate table, TABLE, xtables:
Lua:	neomezené možnosti:

Obrázek 1: Tabulka vytvořená prostředím `table`.

V ukázce provedeme dvě změny ve vyrovnání řádků (modře zvýrazněné příkazy), výsledek představuje obrázek 2.

```
\starttable[|r|p(0.6\textwidth)|]
\NC Podle úpravy:
  \NC pořadově sázené, otevřené, uzavřené,
    s členícími linkami\,\dots\FR
\NC Podle účelu:
  \NC formuláře, bilance, knižní, časopisecké\,\dots
  \SR\HL
\NC Podle nástroje:\NC\MR\HL[2]
\NC \LaTeX \NC tabbing\MR
\NC \NC tabular + halda balíčků\MR\HL
\NC \ConTeXt: \NC tabulate\MR
\NC \NC table, TABLE, xtables\MR\HL
\NC Lua:\NC neomezené možnosti\MR\HL[3]
\stoptable
```

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje	
L ^A T _E X	tabbing tabular + halda balíčků
ConT _E Xt:	tabulate table, TABLE, xtables
Lua:	neomezené možnosti

Obrázek 2: Tabulka (`table`) se změnou vyrovnání řádků.

Vyznačování sloupců, svislé linky

Pokud chceme, aby v daném místě byl sloupec ohraničen svislou linkou, použijeme namísto příkazu `\NC` příkaz `\VL` (vertical line, svislá linka). Tento způsob se ukazuje jako poněkud nepraktický, pokud sazeč nemá předem rozmyšleno, kde se budou linky nacházet a kde nikoliv.

Vodorovné linky

Předchozí ukázky obsahují dosud nezmíněný příkaz `\HL` vytvářející vodorovné linky. Jeho nepovinný parametr slouží k určení stupně tloušťky čáry. Příпустné hodnoty jsou 1, 2, 3. I přesto, že tento způsob uživatele určitým způsobem omezuje, pro běžnou práci postačuje.

ConTeXt: pořadová sazba – `tabulate`

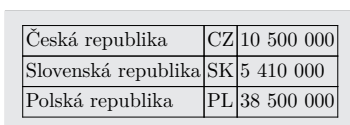
Prostředí `tabulate` pro pořadovou sazbu vychází z prostředí `table` – s tabulkou pracujeme úplně stejně (způsob definice sloupců, ohraničení sloupců atd.), máme však k dispozici širší repertoár formátovacích povelů a parametrů. Toto prostředí lze využít pro tabelaci. Podrobnější popis uvádí Otten a Hagen (2006), další inspiraci přináší například Mahajan (2007 a 2008).

ConTeXt: Natural Tables – `TABLE`

Natural tables (přirozené tabulky) vznikly o něco později a autor `CONTEXt`u Hans Hagen se inspiroval jazykem HTML. Z následujícího příkladu a jeho výstupu na obrázku 3 je patrné, jak se jednotlivé značky používají.

```
\bTABLE
\bTR\bTD Česká republika\eTD\bTD CZ \eTD\bTD 10 500 000\eTD\eTR
\bTR\bTD Slovenská republika\eTD\bTD SK \eTD\bTD 5 410 000\eTD\eTR
\bTR\bTD Polská republika\eTD\bTD PL \eTD\bTD 38 500 000\eTD\eTR
\eTABLE
```

Vidíme zde sice náhradu slov `start` a `stop` za `b` a `e` oproti běžným zvyklostem v `CONTEXt`u, ale mocný příkaz `\setupTABLE`, kterým lze elegantně nastavit řadu vlastností tabulce jako celku i jejím jednotlivým částem, je k dispozici.



Česká republika	CZ	10 500 000
Slovenská republika	SK	5 410 000
Polská republika	PL	38 500 000

Obrázek 3: Výsledný tvar tabulky s implicitním nastavením.

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami ...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje:	
L ^A T _E X:	tabbing
	tabular + halda balíčků
ConT _E Xt:	tabulate
	table, TABLE, xtables
Lua:	neomezené možnosti

`\setupTABLE[c] [2] [align=middle]`

Obrázek 4: Tabulka s centrovaným druhým sloupcem.

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami ...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje:	
L ^A T _E X:	tabbing
	tabular + halda balíčků
ConT _E Xt:	tabulate
	table, TABLE, xtables
Lua:	neomezené možnosti

`\setupTABLE[r] [odd] [offset=-2dd]`

Obrázek 5: Tabulka se záporným vnitřním okrajem.

TABLE – nastavení vlastností

Příkaz `\setupTABLE` má velmi pružnou syntaxi. Můžeme jím nastavovat vlastnosti celé tabulky, jednotlivých řádků, sloupců či buněk. Zápis je poměrně pohodlný a především se jedná o koncepční přístup při definici vlastností sazby.

Ukázky na obrázcích 4–7 postupně předvedou vybrané vlastnosti z velmi rozsáhlého repertoáru. Pro lepší pochopení uveďme, že prvním parametrem udáváme, zda definujeme sloupce (`c`) či řádky (`r`), druhým parametrem číslo sloupce nebo řádku, případně můžeme použít označení `first`, `last` a třetí parametr pak obsahuje nastavení vlastností¹. Záporná čísla slouží k počítání sloupců či řádky od konce tabulky.

¹Zde je potřeba zmínit, že hodnoty `left` a `right` u atributu `align` udávají poněkud netradičně umístění praporku, nikoliv zarovnání okraje.

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje:	
L ^A T _E X:	tabbing
	tabular + halda balíčků
ConT _E Xt:	tabulate
	table, TABLE, xtables
Lua:	neomezené možnosti

```
\setupTABLE[r][1,last][background=color,
backgroundcolor=yellow]
```

Obrázek 6: Tabulka s barevným pozadím prvního a posledního řádku.

Podle úpravy:	pořadově sázené, otevřené, uzavřené, s členícími linkami...
Podle účelu:	formuláře, bilance, knižní, časopisecké ...
Podle nástroje:	
L ^A T _E X:	tabbing
	tabular + halda balíčků
ConT _E Xt:	tabulate
	table, TABLE, xtables
Lua:	neomezené možnosti

```
\setupTABLE[1][4,6][align=left,
background=color,backgroundcolor=yellow]
```

Obrázek 7: Tabulka s barevným pozadím u vybraných buněk.

Speciálním případem je situace, kdy první parametr obsahuje číslo. Tím říkáme, že určujeme vlastnosti konkrétní buňky či buněk. První parametr pak chápeme jako sloupcovou souřadnici, druhý jako řádkovou.

Zarovnání čísel v prostředí TABLE

S požadavkem na zarovnání hodnot na desetinnou čárku (nebo tečku či jiný znak) se sazeči setkávají poměrně často. Nejedná se o zcela triviální problém, avšak prostředí TABLE nabízí relativně jednoduché řešení, jak ukazují poslední dva příkazy v kódu na obrázku 8. Zbývající příkazy zajišťují vodorovné i svislé linky. Další příklady popisuje například Egger (2003).

Málo:	Hodně
10.000 :	1,0
-10.00 :	10,000000
1000.0000:	-1000,000

```

\setupTABLE[frame=off]
\setupTABLE[column][first][leftframe=on]
\setupTABLE[column][last][rightframe=on]
\setupTABLE[row][first][topframe=on]
\setupTABLE[row][first,last][bottomframe=on]
\setupTABLE[column][1][alignmentcharacter={.},
aligncharacter=yes,align=middle]
\setupTABLE[column][2][alignmentcharacter={,},
aligncharacter=yes,align=middle]

```

Obrázek 8: Tabulka se čísly zarovnanými podle požadavku.

CONTEXt: Extreme Tables – xtables

Jedná se o nejnovější a doporučovaný nástroj, který rozšiřuje možnosti tabulkové sazby o další nástroje (Hagen, 2015). Koncept strukturních značek odpovídá logice prostředí `table`. Rozsah tohoto příspěvku bohužel neumožňuje představit zde všechny zajímavé vlastnosti.

Následující úsek kódu představuje ukázkou jednoduché tabulky vytvořené v prostředí `xtables` (převzato od Hagen, 2015).

```

\startxtable[offset=1cm]
\startxrow
\startxcell one \stopxcell
\startxcell two \stopxcell
\stopxrow
\startxrow
\startxcell alpha \stopxcell
\startxcell beta \stopxcell
\stopxrow
\stopxtable

```

Zpracování CSV

Ve všech dosud uvedených příkladech jsme předpokládali, že data jsou součástí zdrojového textu sázeného v jazyce `CONTEXt`, případně vložena příkazem `\input` z jiného souboru. Často se však lze setkat se situací, kdy máme data ve formátu `CSV` a nechce se nám tato data ručně ani automatizovaně značkovat.

Formát CSV

`CSV` (comma separated values) je formát uložení dat v souboru, ve kterém řádky představují záznamy a v rámci řádků jsou jednotlivé hodnoty jednotně odděleny čárkou jakožto dohodnutým oddělovačem. Formát `CSV` není popsán žádnou

normou, existuje k němu však specifikace (Shafranovich, 2016). Dnes se název formátu zobecňuje na všechna podobná data nezávisle na zvoleném oddělovacím znaku. Někteří autoři (Korpela, 2016; Raymond, 2016) však používají jiná označení, například obecné DSV (delimiter separated values) nebo konkrétnější TSV (tabulator separated values).

Modul database

Modul database (autor Hans Hagen, soubor `m-database`) je dostupný například v distribuci `TEXlive` nebo na stránkách `CONTEXTu` (příspěvatelé CG, 2017). Distribuovaná verze modulu `m-database.mkiv` obsahuje dataci 2010, verze souboru `m-database.lua` nese označení 1.001. Není známo, zda autor modul dále vyvíjí či nikoliv, lze jej však i přes jen velmi stručnou dokumentaci prakticky používat.

Miklaveč (2016) uvádí dosud neřešený rozpor se specifikací CSV (Shafranovich, 2016): Podle specifikace totiž může buňka, je-li správně označena uvozovkami, vést přes více řádků. Víceřádkový obsah jedné buňky však není implementován, a proto je pak výsledná tabulka chybně formátována.

Připojení modulu database, definice čtení a zobrazení tabulky

Modul database připojíme v preambuli, a to obvyklým způsobem:

```
\usemodule[database]
```

Poté uvedeme, jak se mají data správně načíst a jak má být tabulka sázena na výstupu. Obojí definujeme parametry příkazu `\defineseparatedlist`, jehož prvním parametrem pojmenujeme zapsané vlastnosti a zároveň, jak ukážeme dále, vytváříme speciální příkazy, kterými ohraničujeme oblast formátu CSV.

Běžná definice pro konverzi formátu CSV do prostředí `TABLE` může vypadat takto:

```
\defineseparatedlist [CSV]
[separator=;,
before=\bTABLE,after=\eTABLE,
first=\bTR,last=\eTR,
left=\bTD,right=\eTD]
```

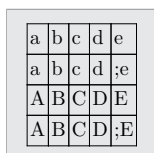
a	b	c	d	e	
a	"b"	"c"	"d"	" e"	
A	B	C	D	E	
A	"B"	"C"	"D"	" E"	

```
a;b;c;d;e
a;"b";"c";"d";";e"
A;B;C;D;E
A;"B";"C";"D";";E"
```

Obrázek 9: Data ve formátu CSV se čtyřmi řádky a pěti sloupci.

Získali jsme výstup (viz obrázek 9), z něhož je patrné, že uvedená definice nebyla dokonalá. Uvozovky, které ohraničují řetězce, nebyly vzaty v potaz. Závadu napravíme doplněním parametru `quotechar`, čímž dostaneme správný výsledek, viz obrázek 10.

```
\defineseparatedlist[CSV]
[separator=;, quotechar=",
before=\bTABLE,after=\eTABLE,
first=\bTR,last=\eTR,
left=\bTD,right=\eTD]
```



a	b	c	d	e
a	b	c	d	;
A	B	C	D	E
A	B	C	D	;

Obrázek 10: Data ve formátu CSV se čtyřmi řádky a pěti sloupci – oprava.

V definici nového prostředí `CSV` jsme uvedli příkazy známé z prostředí `TABLE`. Tím jsme nyní získali možnost tabulky pohodlně formátovat a graficky upravovat pomocí příkazu `\setupTABLE`.

Dva způsoby psaní příkazu

Jak již bylo zmíněno, příkazem `\defineseparatedlist[CSV][...]` definujeme chování prostředí pro sazbu tabulky dat ve formátu CSV, které se uplatní mezi `\startseparatedlist[CSV]` a `\stopseparatedlist`. Můžeme však také použít stručnější způsob zápisu:

```
\startCSV
a;b;c;d;e
a;"b";"c";"d";"e"
A;B;C;D;E
A;"B";"C";"D";"E"
\stopCSV
```

Konverze \LaTeX ové tabulky do \ConTeX tu

Před více než 15 lety autor tohoto příspěvku sázel v \LaTeX u. Jedním z vysázených děl byla učebnice o radioaktivitě (J. Hála, 1999), jejíž autor nyní připravuje nové vydání. Je potřeba s co nejmenší námahou tyto \LaTeX ové tabulky převést, resp. použít v \ConTeX tu.

Takto vypadal původní L^AT_EXový zdrojový text:

```

\begin{center}\setlength{\tabcolsep}{1mm}\def\arraystretch{1.2}
\let\hp=\hphantom
\small\sf\begin{tabular}{|c|l|l|c|l|} \hline
%~-
Z
& známé & & izotop & & T (s)
& reakce\\
& izotopy (A) & & & & s-nejdelším T&
& & & & & \\
& & & & & \\
101 & 248--259 & & & & \\
& & & & & \\
102 & 250--259 & & & & \\
& & & & & \\
103 & 252--262 & & & & \\
& & & & & \\
104 & 253--262 & & & & \\
& & & & & \\
105 & 255--258, 260--263 & & & & \\
& & & & & \\
106 & 258--261, 263 & & & & \\
& & & & & \\
107 & 261, 262, 264 & & & & \\
& & & & & \\
108 & 264, 265, 267, 269 & & & & \\
& & & & & \\
109 & 266, 268 & & & & \\
& & & & & \\
110 & 269, 271--273 & & & & \\
& & & & & \\
111 & 272 & & & & \\
& & & & & \\
112 & 272 & & & & \\
& & & & & \\
%~+
\end{tabular}\end{center}}{}
\endTAB

```

Na tabulku zapsanou v L^AT_EXu může určitým způsobem pohlížet také jako na data typu CSV. Pro oddělení sloupců slouží znak & a pokud fyzicky tabulku

Z	známé	izotop	T (s)	reakce
	izotopy (A)	s nejdelším T		[2.5mm]
101	248–259	²⁵⁸ Md	55 dní	²⁵⁵ Es(α , n)
102	250–259	²⁵⁵ No	185	²⁴⁴ Pu(¹⁶ O, 5n)
103	252–262	²⁵⁶ Lr	45	²⁴³ Am(¹⁸ O, 5n)
104	253–262	²⁶¹ Rf	65	²⁴⁸ Cm(¹⁸ O, 5n)
105	255–258, 260–263	²⁶² Db	34	²⁴⁹ Bk(¹⁸ O, 5n)
106	258–261, 263	²⁶³ Sg	0,9	²⁴⁹ Cf(¹⁸ O, 4n)
107	261, 262, 264	²⁶² Bh	0,0061	²⁰⁹ Bi(⁵⁴ Cr, 2n)
108	264, 265, 267, 269	²⁶⁹ Hs	19,7	produkt α přeměny ²⁷³ 110
109	266, 268	²⁶⁶ Mt	0,0034	²⁰⁹ Bi(⁵⁹ Fe, n)
110	269, 271–273	²⁶⁹ 110	0,0027	²⁰⁸ Pb(⁶² Ni, n)
111	272	²⁷² 111	0,0015	²⁰⁹ Bi(⁶⁴ Ni, n)
112	272	²⁷² 112	0,00028	²⁰⁸ Pb(⁷⁰ Zn, n)

Obrázek 11: Přehled nuklidů transfermiových prvků po vysázení `CONTEXT`em bez dalších typografických úprav.

Zpracování dat z externího souboru

Dosavadní ukázky vycházely z předpokladu, že data jsou součástí zdrojového textu. Získat data z externího souboru však lze také:

```
\processseparatedfile [TSV] [filename]
```

ScanCSV a HandleCSV

Problematikou zpracování souborů ve formátu CSV, zejména pro vlastní školské potřeby, se zabývá Jaroslav Hajtmar. Na konferencích `TEXperience` v r. 2012 a `CONTEXT Meeting` v r. 2013 prezentoval svou knihovnu `ScanCSV` (Hajtmar, 2012, soubor `t-scancsv.lua`). Dnes však tuto knihovnu sám autor již považuje za zastaralou a nahradil ji novější verzí nazvanou `HandleCSV`, která se skládá ze dvou souborů `t-handlecsv.lua` a `t-handlecsv-tools.lua`. Na jejím současném vývoji se podílí dále Pablo Rodriguez. Zatím se jedná však spíše o soukromý, méně známý projekt s domovskou stránkou (Rodriguez, 2016), z níž vede odkaz na GitHub, kde zájemce nalezne zdrojový kód a stručnou dokumentaci.

Na jednoduchém příkladu se můžeme podívat, jak lze s knihovnou `HandleCSV` pracovat. Vstupní data a požadovaná výstupní sestava jsou na obrázku 12. Data zpracujeme následujícím zdrojovým kódem.

```
\usemodule [handlecsv]
\setheader
\opencsvfile {a.csv}
```

```
"Name";"Surname";"Birthdate"
"John";"Smith";10/03/02
"Jane";"Newman";03/03/92
```

```
John Smith was born on 10/03/02.
Jane Newman was born on 03/03/92.
```

Obrázek 12: Vstupní data a požadovaná výstupní sestava.

První položka	
Druhá položka	8.45 EUR
Třetí položka	5.90 EUR
VAT 21%	3.75 EUR
Celkem	21.60 EUR

Obrázek 13: Výsledná tabulka obsahující část faktury.

```
\starttext
  \startbuffer[loop]
    \cA\ \cB\ was born on \cC.\crlf
  \stopbuffer
  \doloop{\getbuffer[loop]%
    \nextrow\ifEOF\exitloop\fi}
\stoptext
```

Tabulkový procesor – modul spreadsheet

Modul `spreadsheet` (autor Hans Hagen, soubor `m-spreadsheet`) umožňuje podle očekávání provádět jednoduché výpočty. Je dostupný v distribuci `TEXlive` i na webových stránkách `CONTEX Garden`. Modul je udržován, ale jeho další vývoj neprobíhá. Dokumentace (Hagen, 2016) je aktualizována, ale po jistou dobu nebyla pro drobnou technickou závadu zařazována do distribuce.

Příkazy `\startcell` a `\stopcell` ohraničují prostředí buňky, do níž se nepíše text k sazbě v jazyce `TEX`, ale v jazyce `Lua`, v němž jsou příslušné výpočty vyhodnocovány. Jedná se o velmi zajímavé a elegantní řešení, avšak méně pohodlné, pokud sázená tabulka obsahuje větší množství „`TEX`ového“ textu a méně výpočtů, neboť všechny texty je potřeba opatřit uvozovkami. Autor modulu sám doporučuje jeho použití jen pro menší výpočty, například faktury (Hagen, 2016). Jeho příklad v drobné úpravě uvádíme i zde, výsledek přináší obrázek 13.

```

\startspreadsheetable[test][frame=off]
\startrow
\startcell[align=flushleft,width=8cm]"První položka" \stopcell
\startcell[align=flushright,width=3cm]@ "0.2f EUR" 3.50 \stopcell
\stoprow
\startrow
\startcell[align=flushleft]"Druhá položka" \stopcell
\startcell[align=flushright]@ "0.2f EUR" 8.45 \stopcell
\stoprow
\startrow
\startcell[align=flushleft]"Třetí položka" \stopcell
\startcell[align=flushright]@ "0.2f EUR" 5.90 \stopcell
\stoprow
\startrow[topframe=on]
\startcell[align=flushleft]"VAT 21\percent" \stopcell
\startcell[align=flushright]@ "0.2f EUR" 0.21*sum(B) \stopcell
\stoprow
\startrow[topframe=on]
\startcellalign=flushleft]"\bf Celkem" \stopcell
\startcell[align=flushright]@ "0.2f EUR" sum(B) \stopcell
\stoprow
\stopspreadsheetable

```

Vlastní řešení

Nyní zde předvedeme speciální řešení zhotovené pro jednu zakázku obsahující ekonomická data, která mají být prezentována jednak tabulkou, jednak grafem.

Zákazník však zpočátku nebyl schopen přesně formulovat požadavky, nebylo jasné, jakou formou se budou data prezentovat, ani jak dlouhá časová řada má být prezentována. K tomu přistoupily i požadavky na snadnou správu, konkrétně obecnou datovou strukturu, s pružným přístupem a indexovatelnou, a také na snadnou editaci zdrojového textu, která musela být jednoduchá.

Pro vstupní data byl navržen tento formát:

```

\tabulka[] [title="Pracoviště A",
data=
RS 2517000 2515000 2386000 1954000
PU 4.57 4.62 4.65 4.05
PKL 361 491 392 268
PK 6699 5508 9475 6012
]

```

Původní úvaha spočívala v načítání dat do dvourozměrného pole s obvyklým oddělovačem řádků.³ Pokud jsou však data předána ke zpracování ve formě atributu v parametrech, jsou součástí zdrojového kódu, předzpracována *input-procesorem*, a tudíž se znak Line Feed změní na mezeru. Pro zjednodušení implementace bylo načítání doplněno o pevně danou konstantu udávající počet sloupců. Toto řešení bylo možno použít proto, že všechny zákaznickovy tabulky měly stejný počet sloupců.

Tabulku vysázíme pomocí `CONTEXTového` makra propojeného s funkcí napsanou v jazyce Lua:

```
\def\tabulka[#1][#2]{
  \ctxlua{userdata.tabulka(' #1 ', '#2')}
}
```

Pro vykreslení sloupcového grafu potřebujeme následující definice:

```
\def\grafwidth{0.5}
\def\graf#1#2{%
  \def\koeff{\csname#1koeff\endcsname}
  \startcolor[#1color]
  \vrule width\grafwidth cc height #2dd\
  \stopcolor
}
```

Následuje zdrojový kód funkce `tabulka` napsaný v jazyce Lua. Protože se jedná o starší projekt, naleznete zde generování tabulky starším způsobem, tj. pomocí prostředí `table`.

```
function userdata.tabulka(keywords, keyvals)
  keyword_options = utilities.parsers.settings_to_array(keywords)
  named_values = utilities.parsers.settings_to_hash(keyvals)
  context.pracoviste(string.unquoted(named_values["title"]))
  local roky = { 2010, 2011, 2012, 2013 }
  roku,kolikroku,delim = #roky,3,', '
  t = string.split(named_values["data"], " ")
  context("\bgroup\setupbodyfont [8dd]
    \starttable[s2|p(15cc)|s3w(4cc)c|w(4cc)c|w(4cc)c|]")
  context("\NC")
  for r=roku-2,roku do
    context("\NC")
    for i=r,#t-1,roku do -- 1., ., 11. ... údaj
      if t[i]~="x" then
        context("\graf{"..t[i-r+1].."}{\compute{"..t[i].."/
          \csname "..t[i-r+1].."koeff\endcsname}}%")
      end
    end
  end
end
```

³Řešení bylo provozováno v Linuxu, jednalo se o znak LF (Line Feed).


```

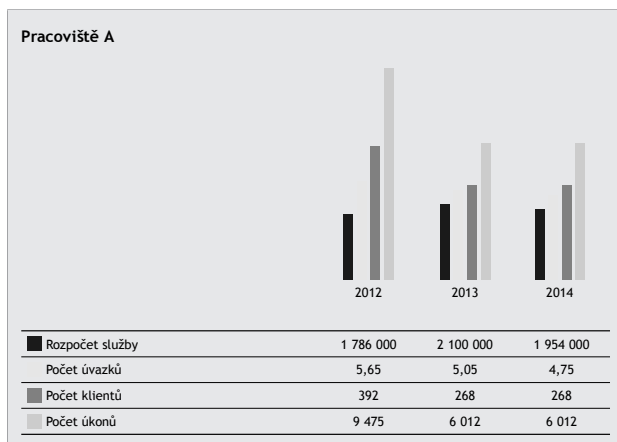
    end
end
context("\AR\NC")
for i=roku-kolikroku+1,roku do context("\NC "..roky[i]) end
context(" \FR\HL[2] ")
for i=1,#t-1 do
    s = i % (roku + 1)
    t[i]=string.strip(t[i])
    if s == 1 then
        if utf.len(t[i]) < 6 then
            context("\NC\ctverec{"..t[i].."color}\
                \csname "..t[i].."endcsname")
        else context("\NC",t[i])
        end
    end
end
if s > kolikroku or s == 0 then
    if delim==',' then
        cislo,pocet=string.gsub(t[i],"[.]",",")
        cislo,pocet=string.gsub(cislo,"(%d)(%d%d)(%d%d)d$",
            "%1\\,\\,%2\\,\\,%3")
        cislo,pocet=string.gsub(cislo,"(%d)(%d%d)d$",
            "%1\\,\\,%2")
    else
        cislo=t[i]
        cislo,pocet=string.gsub(cislo,"(%d)(%d%d)(%d%d)d$",
            "%1,%2,%3")
        cislo,pocet=string.gsub(cislo,"(%d)(%d%d)d$", "%1,%2")
    end
end
context("\NC "..cislo)
end
if s == 0 then context("\FR\HL") end
end
context("[2]\stoptable\egroup\par");
end

```

Výslednou tabulku i se sloupcovým grafem ukazuje obrázek 14.

Projekt CTX-CSV

Další specifické potřeby v oblasti zpracování dat ve formátu CSV vedly autora tohoto příspěvku k zahájení vývoje vlastních nástrojů, zejména s výstupem ve



Obrázek 14: Výsledná tabulka i se sloupcovým grafem.

formě grafů, pro což není v `CONTEXTu` zatím kompaktní a uživatelsky přívětivá podpora. Vzhledem k probíhajícímu vývoji není projekt zatím zveřejněn.

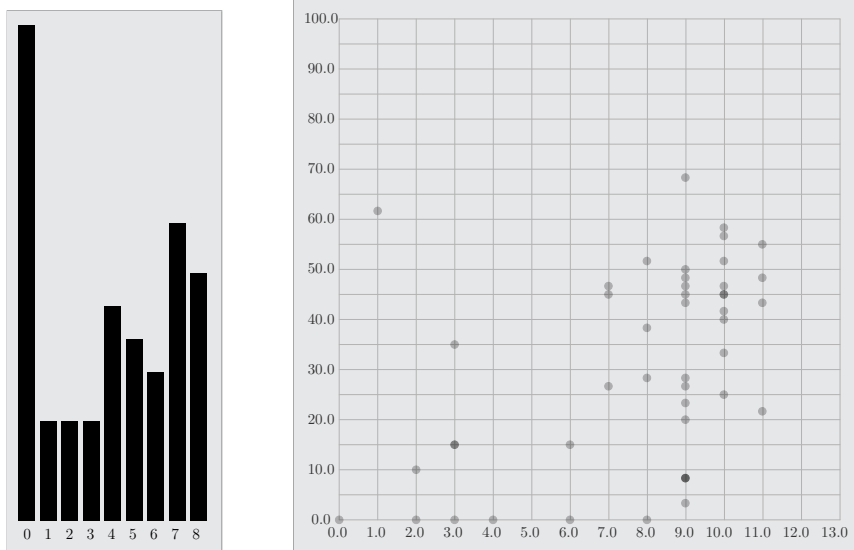
Projekt se skládá ze tří částí – souboru `csv-tools.lua`, obsahujícího vstupně výstupní operace pro zpracování souboru ve formátu CSV, souboru `csv-stat.lua`, zabývajícího se grafickou reprezentací načtených dat, a souboru `t-csv.mkiv`, který slouží jako modul. Grafy jsou vykreslovány `MetaPostem`, výpočty zajišťují úseky kódu v jazyce `Lua`.

Základní vstupně-výstupní operace se souborem ve formátu CSV provádějí následující tři makra:

```
\CSVRead[comma2dot] [data=\soubor,separator=|]
\CSVWrite[dot2comma] [output=pomout.csv,enclosechar="]
\CSVTABLE[dot2comma] []
```

Následuje ukázka části funkce v jazyce `Lua`, která reprezentuje tabulkové výpočty v načtených datech. Funkce je užita v příkazu `\CSVNewColumn`:

```
function vypocet(t,c)
  keyword_options = utilities.parsers.settings_to_array(keywords)
  named_values = utilities.parsers.settings_to_hash(keyvals)
  ...
  if t[r][c+3]>0 then t[r][c+10]=t[r][c+3] end
  if t[r][c+3]>1 then t[r][c+11]=t[r][c+3] end
  if t[r][c+3]>2 then t[r][c+12]=t[r][c+3] end
  if t[r][c+3]>3 then t[r][c+13]=t[r][c+3] end
  ...
return t end
```



Obrázek 15: Ukázka vykreslených grafů.

Příkazy sloužící k výpočtům a k vykreslení grafů mohou být například tyto:

```

\CSVNewColumn [] [table=1,column=23,expr=vypocet]
\CSVComputeColumn [] [table=1,column=23,startrow=2,meze={0,13}]
\CSVDrawDistribution [] [table=1,column=23,maxparts=12]
\CSVComputeCorrelation [] [table=1,columns={23,18},startrow=2
%,meze={0,60}]
\CSVDrawCluster [] [table=1,xdraw=19.2,columns={23,18},startrow=2
%,groupby=2]

```

Ukázku vykreslených grafů obsahuje obrázek 15.

Závěr

CONTEXT nabízí zajímavé možnosti sazby tabulek. Představené ukázky vycházejí jednak z řešení čistě „T_EXových“, tj. ze sady strukturních značek, jednak z možností, jež přináší jazyk Lua.

I přesto, že máme k dispozici vícero možností, základní vlastnosti jsou integrovány do jednoho celku, což zbavuje uživatele nutnosti připojovat různé balíčky, jako je tomu v L^AT_EXu, z nichž některé nejsou ani vzájemně kompatibilní, jak dokumentuje Talandová (2008).

V oblasti zpracování formátu CSV však různé přístupy různých autorů ukazují zatím spíše na tříštění sil než na hledání cesty pro vývoj univerzálního prostředku.

Odkazy

- EGGER, Willi, 2003. *My Way: Use of the natural table environment* [online] [cit. 2019-06-21]. Dostupné z: <http://dl.contextgarden.net/myway/NaturalTables.pdf>.
- HAGEN, Hans, 2015. *Extreme Tables: CONTEXt MkIV* [online] [cit. 2016-06-21]. Dostupné z: <http://www.pragma-ade.com/general/manuals/xtables-mkiv.pdf>.
- HAGEN, Hans. *Simple Spreadsheets: CONTEXt MkIV* [online] [cit. 2016-07-20]. Dostupné z: <http://www.pragma-ade.nl/general/manuals/spreadsheets-mkiv.pdf>.
- HAJTMAR, Jaroslav, 2012. ScanCSV – Lua knihovna pro zpracování CSV souborů CONTEXtem a LuaL^AT_EXem. *Zpravodaj ČSTUG*. Roč. 22, č. 2, s. 76–90. ISSN 1211-6661. Dostupné z DOI: 10.5300/2012-2/76.
- HÁLA, Jiří, 1999. *Radioaktivita, ionizující záření, jaderná energie*. Konvoj. ISBN 80-85615-56-8.
- HÁLA, Tomáš, 2013. L^AT_EX, nebo CONTEXt?: První zkušenosti se sazbou CONTEXtem. *Zpravodaj ČSTUG*. Roč. 23, č. 1, s. 57–64. ISSN 1211-6661. Dostupné z DOI: 10.5300/2013-1/57.
- HÁLA, Tomáš, 2015. Proč jsem zkusil CONTEXt. In: *Zborník príspevkov medzinárodnej konferencie OSSConf 2015*. Žilinská univerzita v Žiline, s. 37–40. ISBN 978-80-970457-7-7.
- KNUTH, Donald Ervin, 1986. *The T_EXbook*. Addison-Wesley. ISBN 0-201-13448-9.
- KORPELA, Jukka. *Tab Separated Values (TSV): A format for tabular data exchange* [online] [cit. 2016-05-06]. Dostupné z: <https://www.cs.tut.fi/~jkorpela/TSV.html>.
- MAHAJAN, Aditya, 2007. CONTEXt basics for users: Table macros. *TUGboat*. Roč. 28, č. 3, s. 372–374.
- MAHAJAN, Aditya, 2008. CONTEXt basics for users: Table macros II. *TUGboat*. Roč. 29, č. 1, s. 219–222.
- MIKLAVEC, Mojca. *My Way: Creating tables using CSV (comma-separated values)* [online] [cit. 2016-05-12]. Dostupné z: <http://dl.contextgarden.net/myway/csv.pdf>.
- OTTEN, Ton a HAGEN, Hans, 2006. Exkurze do CONTEXtu. *Zpravodaj ČSTUG*. Roč. 16, č. 2–4, s. 57–224. ISSN 1211-6661. Dostupné z DOI: 10.5300/2006-2-4/1.

- PŘISPĚVATELÉ CONTEXT GARDEN, 2017. CONTEXT *Garden: Modules* [online] [cit. 2017-06-27]. Dostupné z: <http://wiki.contextgarden.net/Modules>.
- RAYMOND, Eric Steven. *The Art of Unix Programming: Chapter 5, Textuality: Data File Metaformats* [online] [cit. 2016-05-06]. Dostupné z: <http://www.catb.org/~esr/writings/taoup/html/ch05s02.html>.
- RODRIGUEZ, Pablo. *HandleCSV* [online] [cit. 2016-05-06]. Dostupné z: <http://www.handlecsv.tk/>.
- SHAFRANOVICH, Y. *Common Format and MIME Type for Comma-Separated Values (CSV) Files* [online] [cit. 2016-05-06]. Dostupné z: <https://tools.ietf.org/html/rfc4180>.
- TALANDOVÁ, Petra, 2008. Možnosti tabulkové sazby. *Zpravodaj ČSTUG*. Roč. 18, č. 3, s. 151–160. ISSN 1211-6661. Dostupné z DOI: 10.5300/2008-3/151.

Summary: Tables in ConT_EXt: Ways, Possibilities, Algorithms

In the publication process, the typesetting of tables belongs to the more complicated tasks. This paper reviews old and current ways of typesetting tables in ConT_EXt (environments `table`, `tabulate`, `TABLE`, `xtables`), and compares them mutually and with the ‘rival’ L^AT_EX.

Tables can be generated from other formats such as the frequently used CSV. Therefore, the paper deals also with database processing.

Finally, some simple algorithms for easy extensions of the available repertoire are presented.

Keywords: ConT_EXt, L^AT_EX, comparison, typesetting, tables, `table`, `tabulate`, `TABLE`, `xtables`, CSV

*Mendelova univerzita, Provozně ekonomická fakulta, ústav informatiky,
Zemědělská 1, 613 00 Brno, thala@mendelu.cz*