

Zpravodaj Československého sdružení uživatelů TeXu

Jan Šustek

Zpracování dat z tabulkového editoru TeXem

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 30 (2020), No. 1-2, 57–63

Persistent URL: <http://dml.cz/dmlcz/150271>

Terms of use:

© Československé sdružení uživatelů TeXu, 2020

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>

Abstrakt: V článku je ukázán způsob, jak je možné načíst a zpracovat v T_EXu data vytvořená v tabulkovém editoru. Uvedená makra jsou podrobně popsána s cílem motivovat čtenáře nebát se v T_EXu drobně programovat. Dále je v článku ukázáno třířádkové makro na vložení celé tabulky do T_EXového dokumentu.

Klíčová slova: CSV, tabulka.

1 Úvod

Poslední dobou vyšlo několik článků, které se zabývají sazbou tabulek v T_EXu. Marek Pomp [1] se věnuje sazbě tabulek vygenerovaných statistickým softwarem R s využitím balíčků L^AT_EXu. Tomáš Hála [2] popisuje tvorbu tabulek v ConT_EXtu.

Tento článek řeší problematiku na úrovni PlainT_EXu¹ a dodržuje tři zásady OPmacu [3]:

- V jednoduchosti je síla.
- Makro není univerzální, ale je čitelné a srozumitelné.
- Uživatel si makro může snadno předefinovat k obrazu svému.

Uvedená makra jsou v článku podrobně popsána s cílem motivovat čtenáře nebát se v T_EXu naprogramovat si jednoduchá makra, a tím si usnadnit práci.

Podobný problém řeší soubor `scansv.tex` [4] od Petra Olšáka, makra jsou však obecnější a umožňují například používat jiný oddělovač položek, než je středník. Daní za toto je menší srozumitelnost a komplikovanější definice použitých maker.

2 Příklad

V sekci 3 si vytvoříme soubor `naticsvs.tex` a v něm nadefinujeme makro

```
1 \zpracuj{soubor}<soubor><činnost>
```

Funkcí makra bude načíst `<soubor>` ve formátu CSV řádek po řádku a s každým řádkem provést `<činnost>`. Jednotlivé údaje na řádku jsou uloženy do maker `\udaji`, `\udajii`, `\udajiii`, `\udajiv` atd. Počet údajů na daném řádku je uložen

¹Makra jsou použitelná i v dalších formátech a nadstavbách T_EXu.

do číselného registru `\pocetudaju`. Číslo aktuálně zpracovávaného řádku CSV souboru je uloženo do číselného registru `\cislорadkucsv`.

Představme si, že máme v tabulkovém editoru tabulku se jmény účastníků nějakého závodu a jejich výsledky. Tuto tabulku uložíme ve formátu CSV do souboru `1.csv`.

```

2 Jiří;Novák;Praha;48:29
3 Petr;Novotný;Ostrava;49:17
4 Jan;Svoboda;Brno;52:13
5 Martin;Procházka;Olomouc;53:07
6 Josef;Dvořák;Plzeň;53:43
7 Tomáš;Kučera;České Budějovice;56:59
8 Pavel;Černý;Liberec;57:23
9 Jaroslav;Veselý;Ústí nad Labem;1:02:31

```

Úkolem bude z této tabulky vysázet výsledkovou listinu v následujícím tvaru.

```

Jiří Novák (Praha) ..... 48:29
Petr Novotný (Ostrava) ..... 49:17
Jan Svoboda (Brno).....52:13
Martin Procházka (Olomouc).....53:07
Josef Dvořák (Plzeň) ..... 53:43
Tomáš Kučera (České Budějovice) ..... 56:59
Pavel Černý (Liberec) ..... 57:23
Jaroslav Veselý (Ústí nad Labem).....1:02:31

```

K tomu si připravíme makro `\vypisucastnika`, které vysází jeden řádek výsledkové listiny. Použitím tohoto makra v argumentu makra `\zpracujсoubor` pak postupně vysázíme celou výsledkovou listinu.

```

10 \input naticsv
11 \def\vypisucastnika{\hbox to\hsize{%
12   \ifnum\cislорadkucsv<4 \bf \else \rm \fi
13   \udaji\ \udajii\ (\udajiii)\dotfill\udajiv}}
14 \zpracujсoubor{1.csv}\vypisucastnika

```

3 Implementace

V této sekci je stručně popsán celý soubor `naticsv.tex`.

Nejdříve nadeklarujeme dříve uvedené číselné registry a nadefinujeme si pomocné makro, které budeme využívat při testování podmínek uvnitř cyklů.

```

15 \newcount\pocetudaju
16 \newcount\cislорadkucsv
17 \def\nic{}

```

V makru `\radek` je vždy uložen obsah celého aktuálního řádku souboru. Makro `\rozdelradek` tento obsah rozdělí na údaje oddělené středníkem. K tomu využije makro `\rozdelradekA`, které v cyklu načte vždy jeden údaj ukončený středníkem a tento údaj uloží do příslušného makra `\udajn`.

```

18 \def\rozdelradek{\pocetudaju0
19 \expandafter\rozdelradekA\radek;\konecradku}
20 \def\rozdelradekA#1;#2\konecradku{\advance\pocetudaju1
21 \expandafter\def
22 \csname udaj\romannumeral\pocetudaju\endcsname{#1}%
23 \def\tmp{#2}%
24 \ifx\tmp\nic\let\next\relax
25 \else\def\next{\rozdelradekA#2\konecradku}%
26 \fi\next}

```

Makro `\zpracujsoubor` uloží svůj argument $\langle cinnost \rangle$ pro pozdější použití do makra `\cinnost`. Dále lokálně nastaví `\endlinechar` na hodnotu -1 , aby na konci načítaných řádků nevznikaly zavlečené mezery. Makrem `\zpracujsouborA` poté spustí cyklus přes jednotlivé řádky $\langle souboru \rangle$. V cyklu je pak každý řádek $\langle souboru \rangle$ zpracován uvnitř skupiny.²

Nesmíme zapomenout, že \TeX na konec každého souboru načítaného pomocí `\read` automaticky přidává prázdný řádek.³ Takový řádek samozřejmě neobsahuje žádný středník a není na něj možné aplikovat makro `\rozdelradek`. Tato situace je ošetřena testem na řádku 35.

```

27 \newread\vstup
28 \def\zpracujsoubor#1#2{{\def\cinnost{#2}%
29 \openin\vstup#1
30 \cislорadkucsv0
31 \endlinechar-1
32 \zpracujsouborA
33 \closein\vstup}}
34 \def\zpracujsouborA{\read\vstup to\radek
35 \ifx\radek\nic\else
36 \advance\cislорadkucsv1
37 {\rozdelradek
38 \cinnost}%
39 \fi
40 \ifeof\vstup\else\expandafter\zpracujsouborA\fi}

```

²Pokud bychom skupinu nevytvořili, pak by v případě, že některý řádek obsahuje méně údajů, ve zbývajících makrech `\udajn` nesprávně zůstaly uloženy údaje z předchozích řádků.

³Knuth tento jev bez zdůvodnění popisuje v [5] na straně 217.

4 Pojmenované sloupce

V této sekci se budeme zabývat situací, kdy v prvním řádku CSV souboru jsou uložena záhlaví sloupců. My tato záhlaví využijeme k pojmenování maker obsahujících údaje, namísto obecného pojmenování `\udajn`. Ukážeme, jak je k tomuto účelu možné efektivně využít příkaz `\csname`, bez nutnosti zasahovat do souboru `nacticsv.tex`.

U našeho příkladu s výsledkovou listinou to znamená, že budeme mít následující data v tabulce uložené v souboru `2.csv`.

```
41 jmeno;prijmeni;mesto;cas
42 Jiří;Novák;Praha;48:29
43 Petr;Novotný;Ostrava;49:17
44 Jan;Svoboda;Brno;52:13
45 Martin;Procházka;Olomouc;53:07
46 Josef;Dvořák;Plzeň;53:43
47 Tomáš;Kučera;České Budějovice;56:59
48 Pavel;Černý;Liberec;57:23
49 Jaroslav;Veselý;Ústí nad Labem;1:02:31
```

To nám umožní zřehlednit makro `\vypisucastnika` a namísto řádku 13 psát řádek 52.

```
50 \def\vypisucastnika{\hbox to\hsize{%
51   \ifnum\cisloradkucsv<4 \bf \else \rm \fi
52   \jmeno\ \prijmeni\ (\mesto)\dotfill\cas}}
```

Zpracování prvního řádku CSV souboru odlišíme od zpracování zbývajících řádků. To provedeme pomocí makra `\jedenradek`, které je na začátku definováno jako makro `\prvniradek`, avšak uvnitř makra `\prvniradek` je předefinováno, aby se na dalších řádcích chovalo jako makro `\vypisucastnika`.

Uvnitř makra `\prvniradek` načteme záhlaví sloupců a nadefinujeme příslušná makra, ať se expandují na správná makra `\udajn`. Zároveň opravíme hodnotu registru `\cisloradkucsv`, aby počítal pouze datové řádky souboru a ne řádek se záhlavím. S ohledem na poznámku pod čarou 2 musí být všechna nastavení v makru `\prvniradek` provedena globálně.

```
53 \input nacticsv
54 \newcount\tmpnum
55 \def\prvniradek{%
56   \tmpnum0
57   \loop \ifnum\tmpnum<\pocetudaju \advance\tmpnum1
58     \expandafter\gdef
59     \csname
```

```

60      \csname udaj\romannumeral\tmpnum\endcsname\expandafter
61      \endcsname\expandafter
62      {\csname udaj\romannumeral\tmpnum\endcsname}%
63      \repeat
64      \global\let\jedenradek\vypisucastnika
65      \global\cisloradkucsv0
66      }
67 \let\jedenradek\prvniradek
68 \zpracuj{soubor{2.csv}\jedenradek

```

Hlavní trik je na řádcích 58–62, které stojí za to rozebrat důkladněji. Předpokládejme, že uvnitř makra `\prvniradek` v cyklu na řádku 57 máme `\tmpnum = 1`, tedy že zpracováváme první údaj prvního řádku. Díky makru `\rozdelradek` máme nadefinováno (viz řádek 22)

```

69 \def\udaji{jmeno}

```

Nyní se na řádku 58 expanduje příkaz `\expandafter`. V rámci této expanze se jedenkrát expanduje příkaz `\csname` na řádku 59. Tím dojde k úplné expanzi řádku 60:

1. Příkaz `\csname` na řádku 60 se postupně expanduje na

```

70 \csname udaji\endcsname
71 \udaji
72 jmeno

```

2. Příkaz `\expandafter` na řádku 60 se expanduje tak, že se expanduje příkaz `\expandafter` na řádku 61 a ten se expanduje tak, že se jedenkrát expanduje příkaz `\csname` na řádku 62. Tímto na řádku 62 dostáváme

```

73 {\udaji}%

```

Výsledkem expanze příkazu `\csname` z řádku 59 tak je, že z řádků 59–61 vznikl jediný token `\jmeno`. Expanzi příkazu `\expandafter` z řádku 58 tak máme hotovou a ve čtecí frontě máme tokeny

```

74 \gdef\jmeno{\udaji}%

```

Když poté, například na druhém řádku souboru, uživatel zavolá makro `\jmeno`, to se expanduje na makro `\udaji` a to se expanduje na výsledný text Jiří.

5 Přímé vysázení tabulky

Na závěr si ukážeme třířádkové řešení z [6], jak je možné v OPmacu přímo vysázet tabulku z CSV souboru.

Vytvoříme si makro `\csvtable⟨deklarace⟩⟨soubor⟩`, které z CSV souboru `⟨soubor⟩` vysází tabulku, jejíž sloupce budou zarovnány podle `⟨deklarace⟩`. Zarovnání sloupců je definováno stejně jako v OPmacovém makru `\table`. V souboru jsou jednotlivé řádky tabulky odděleny koncem řádku souboru. Jednotlivé buňky jsou odděleny středníkem.

Nechť soubor 3.csv obsahuje text

```
75 a;bc;de
76 fgh;ijk;l
77 m;nop;qrsr
```

Potom budeme chtít, aby se po použití

```
78 \csvtable{|l|r|r|}{3.csv}
```

vysázela následující tabulka.

a	bc	de
fgh	ijk	l
m	nop	qrsr

Práci makra `\csvtable` vložíme do skupiny, v níž lokálně přiřadíme znaku středník kategorií 4 (bude se chovat jako oddělovač buněk v tabulce) a znak konce řádku nadefinujeme jako aktivní, aby se expandoval na `\crl`. Pak už jen použijeme makro `\table` a v něm načteme daný soubor. Dostáváme následující definici makra `\csvtable`.

```
79 \def\csvtable#1#2{\catcode'\;=4 \edef{\^~M}{\crl}
80 \table{#1}{\crl\input#2}}}
```

Nyní si naše makro vylepšíme tak, aby umělo zpracovávat buňky obsahující středník. Obsah takové buňky je uzavřen do uvozovek.

Nechť soubor 4.csv obsahuje text

```
81 a;bc;de
82 fgh;"ij;k";l
83 m;nop;"qr;sr"
```

Potom budeme chtít, aby se po použití

```
84 \csvtable{|l|r|r|}{4.csv}
```

vysázela následující tabulka.

a	bc	de
fgh	ij;k	l
m	nop	qr;sr

Pro tyto účely stačí makro `\csvtable` rozšířit o jeden řádek. Znak uvozovky nastavíme jako aktivní, ať otevře skupinu, uvnitř které

- bude mít znak středník kategorii 12 (vysází se) a
- znak uvozovky bude aktivní a zavře skupinu.

Celá definice makra `\csvtable` pak je tato.

```

85 \def\csvtable#1#2{{\catcode'\;=4 \edef{\^M}{\crl}
86 \edef{"}{\begingroup\edef{"}{\endgroup}\catcode'\;=12 }
87 \table{#1}{\crl\input#2}}}
```

Seznam literatury

1. POMP, Marek. Tabulky v dobře dokumentovaných statistických výpočtech. *Zpravodaj ČSTUG*. 2018, roč. 25, č. 1–4, s. 22–37. ISSN 1211-6661. Dostupné z DOI: 10.5300/2018-1-4/22.
2. HÁLA, Tomáš. Tabulky v ConT_EXtu: přístupy, možnosti, algoritmy. *Zpravodaj ČSTUG*. 2019, roč. 25, č. 1–4, s. 24–43. ISSN 1211-6661. Dostupné z DOI: 10.5300/2019-1-4/24.
3. OLŠÁK, Petr. *OPmac: rozšiřující makra plain T_EXu* [online]. 2017-06-17 [cit. 2020-03-24]. Dostupné z: <http://petr.olsak.net/opmac.html>.
4. OLŠÁK, Petr. *scancsv.tex* [online]. 2005-02-26 [cit. 2020-03-24]. Dostupné z: <http://petr.olsak.net/ftp/olsak/makra/scancsv.tex>.
5. KNUTH, Donald Ervin. *Computers & Typesetting, Volume A: The T_EXbook*. Addison-Wesley, 1986.
6. KOLOUCH, Ondřej; NOVOTNÝ, Lukáš; ŠUSTEK, Jan. *T_EX for Master Level / T_EX pro magisterské studium*. Ostravská univerzita, 2016.

Summary: Processing Spreadsheet Data in T_EX

In the paper we show a way to read and process spreadsheet data in T_EX. The macros are described in detail, motivating readers to create simple macros without doubts. We also show a three-line macro for inserting a whole table into a T_EX document.

Key words: CSV, table.

*Jan Šustek, jan.sustek@osu.cz
Ostravská univerzita, Přírodovědecká fakulta, Katedra matematiky
30. dubna 22, CZ-701 03 Ostrava, Czech Republic*