

Zpravodaj Československého sdružení uživatelů TeXu

Z. Hlávka

Velkovýroba tabulek pomocí AWK

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 18 (2008), No. 1-2, 93–95

Persistent URL: <http://dml.cz/dmlcz/149934>

Terms of use:

© Československé sdružení uživatelů TeXu, 2008

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>

1. Úvod

Zkratka AWK znamená Aho, Weinberger, Kernighan a označuje poněkud svérázný programovací jazyk navržený pro zpracovávání textových souborů.

První verze tohoto programovacího jazyka vznikla již v roce 1977. V roce 1985 byl tento programovací jazyk dále rozšířen [?] a tato nová implementace byla vydána pod volnou licencí v roce 1996. Další informace lze nalézt na stránce Briana Kernighana: <http://cm.bell-labs.com/cm/cs/awkbook/>. Za pozornost stojí i kniha [?].

GNU awk (nebo gawk) je jiná implementace jazyka AWK, která je (téměř jistě) obsažená v každé distribuci Linuxu. Další distribuce existují pod názvy xgawk, mawk nebo TAWK (verze pro DOS a Windows) i GNU AWK (OS/2).

2. Příklad

Princip jazyka AWK si ukážeme na jednoduchém příkladě. Předpokládejme, že výsledky rozsáhlé simulační studie jsou uloženy v následujícím značně nepřehledném textovém souboru `priklad.txt`:

```
TABULKA.AA "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L"
"1" 1 2.0599209764691 1.1747333245979 1.03814385890582 2.0599209746917
1.10206315043 27 1.00304826425488 1.16483656626828 1.04785974714491
1.0215094731292 1.1656626828 1.03386899720460 1.00215433687505
"2" 1 1.2111642320962 1.1785509411592 1.09217031887314 1.21116423209622
1.168397750859 1.0225654118313 1.1664016471655 1.05402221156164
1.00649949971005 1.1664064716555 1.02731120933558 1.00184132881134
"3" 1 1.3220358323917 1.0628344035499 1.01922692920646 1.32203958323917
1.052451742835 1.00663451863578 1.0622641328610 1.0229410009790
1.01245982055420 1.0622661328608 1.02234509043763 1.00537251503339
"4" 1 3.0406725862321 1.12150481270856 1.0758055731613 3.04367252319
1.1156698237625 58 1.00211380906327 1.2487169003298 1.102381764515
1.01161646944822 1.24871679003298 1.10077794969210 1.00206064501333
```

Programovací jazyk AWK umožňuje tento textový výstup jednoduše přeformátovat do podoby L^AT_EXové tabulky. Následující postup funguje se standardní Linuxovou implementací `gawk`.

Nejprve vytvoříme soubor `tabulka.awk`, který obsahuje všechny potřebné instrukce:

```

BEGIN{
    getline;
    print "\\begin{tabular*}%";
    print "{\\textwidth}{1|*{" ,NF-1,"}{@{\\extracolsep\\fill}r}}";
    row=1
}

{
    printf "%.0f",row;
    for (i=2;i<=NF/2;i++) {
        printf "%%.1f",$(i+NF/2);
        printf "&{\\scriptsize (%.3f)}", $i;
    }
    print "\\\\";
    row=row+1
}

END{
    print "\\end{tabular*}";
}

```

Úvodní část programu, za příkazem `BEGIN`, obsahuje příkazy, které se provedou pouze na počátku. Zde vytiskneme hlavičku naší tabulky. Proměnná `NF` zde vytiskne počet záznamů na prvním řádku vstupního textového souboru (musíme odečíst 1, protože výsledná tabulka bude mít o jeden sloupeček méně). Vytvoříme proměnnou `row`, která bude obsahovat číslo řádku. Příkaz `getline` „přeskočí“ nezajímavý první řádek.

Hlavní část programu nalezneme v dalších složených závorkách. Tyto příkazy se vykonají postupně pro každý řádek vstupního textového souboru. Nejdříve vytiskneme číslo řádku a pak postupně tiskneme čísla z druhé poloviny textového souboru (velkým písmem a s přesností na jedno desetinné místo) a za ně vždy příslušná čísla z první poloviny textového souboru (menším písmem, v závorce a s přesností na tři desetinná místa). Vše doplňujeme potřebnými oddělovači a na konec každého řádku nezapomeneme vytisknout dvě obrácená lomítka.

Ve třetí části programu, za příkazem `END`, již vytiskneme `\end{tabular*}` a ukončíme tím tabulku.

Z příkazové řádky spustíme náš program:

```
gawk -f tabulka.awk < priklad.txt > priklad.tex
```

Získáme pěkně zformátovanou L^AT_EXovou tabulku v souboru `priklad.tex`, který obratem vložíme do našeho příspěvku pomocí `\input{priklad.tex}` příkazu:

```

\begin{table}
\input{priklad.tex}
\caption{Pěkně zformátovaná tabulka.}
\end{table}

```

1	1.2	(1.000)	1.0	(2.060)	1.0	(1.175)	1.2	(1.038)	1.0	(2.060)	1.0	(1.130)
2	1.2	(1.000)	1.1	(1.211)	1.0	(1.179)	1.2	(1.092)	1.0	(1.211)	1.0	(1.162)
3	1.1	(1.000)	1.0	(1.322)	1.0	(1.063)	1.1	(1.019)	1.0	(1.322)	1.0	(1.058)
4	1.2	(1.000)	1.1	(3.040)	1.0	(1.122)	1.2	(1.018)	1.1	(3.040)	1.0	(1.110)

Tabulka 1: Pěkně zformátovaná tabulka.

Výsledek našeho snažení se vzápětí objeví v pěkně zformátované tabulce 1.

Hlavní výhoda tohoto postupu je fakt, že pro změnu ve formátování tabulky 1 stačí tuto změnu provést pouze na jednom místě v `tabulka.awk`. Naprosto triviální je například změna počtu desetinných míst pro všechna čísla v tabulce 1.

Celý postup lze samozřejmě ještě více automatizovat a zjednodušit. Například, program `gawk` můžeme zavolat přímo z našeho statistického software už při ukládání výsledků. Pokud jsme nedočkaví, tak můžeme tabulku zároveň i přeložit \LaTeX em do PDF nebo PS. Lze si představit i jiné použití v kombinaci s příkazem `Sweave()` ve statistickém programovacím prostředí `R` dostupném na: <http://www.r-project.org/>. Pro RedHat systémy najdete RPM na serveru <http://fedoraproject.org/wiki/EPEL>.

3. Závěr

Je zřejmé, že `AWK` můžeme využít i k jiným, náročnějším úkolům. Díky své rychlosti se hodí k provádění jednoduchých operací na rozsáhlých datových souborech, k automatizaci psaní opakujících se počítačových programů, nebo k získávání potřebných informací z jasně zformátovaných rozsáhlých \LaTeX ových souborů. Jako příklad si lze představit automatické odesílání emailů autorům příspěvků v tomto sborníku, vždy s automaticky vyplněnou správnou emailovou adresou, automaticky vyplněným správným jménem autora a automaticky vyplněným správným názvem příspěvku.

Fantazii se zkrátka žádné meze nekladou.

Summary: Large-scale Production of Tables in AWK

This short article demonstrates the capabilities of `AWK` when producing \LaTeX tables and formatting them in large quantities.

Key words: `AWK`, Large-scale Table Production, Table Formatting.

Zdeněk Hlávka, KPMS MFF UK Praha
hlavka@karlin.mff.cuni.cz