

Zpravodaj Československého sdružení uživatelů TeXu

Peter Wilson

Mohlo by to fungovat. I - Porovnávání řetězců

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 20 (2010), No. 4, 297–301

Persistent URL: <http://dml.cz/dmlcz/150150>

Terms of use:

© Československé sdružení uživatelů TeXu, 2010

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

Článek se zabývá srovnáváním řetězců na úrovni Plain \TeX u. Jedná se o první překlad do češtiny ze série *Glisterings* z časopisu TUGboat.

Klíčová slova: \TeX , Plain \TeX , srovnávání řetězců.

doi: 10.5300/2010-4/297

Not all that tempts your wand'ring eyes
And heedless hearts, is lawful prize;
Nor all, that glisters, gold.

Ode to a Favourite Cat
THOMAS GRAY

Předmluva

Po mnoho let Jeremy Gibbons editoval velmi úspěšný sloupek v časopisech *\TeX and TUG News* a *TUGboat* nazvaný *Hey—It works!* [4]. Odtud jsem se naučil hodně, ale zřejmě ne dost natolik, abych mohl pozměnit definice, když jsem potřeboval naprogramovat něco podobného. Na druhou stranu jsem se sám naučil, že nejrychlejším způsobem, jak na diskusní skupině `comp.text.tex` dostat správnou odpověď, je napsat nesprávnou odpověď. Abych nepošpinil Jeremyho dobrou pověst, byla moje první myšlenka změnit název seriálu na *Hey—It might work*, ale po delším přemýšlení jsem zvolil název *Glisterings*. Název by měl naznačovat, že mezi zrníčky zlata mohou být i zrníčka písku.¹

Opravy, poznámky a návrhy na změny budou vždy vítány.

V nedávné době padlo na `comp.text.tex` několik otázek týkajících se porovnávání dvou slov nebo řetězců. Byl jsem zklamaný, že jsem na jednu z otázek dal nesprávnou odpověď. Pokusím se to nyní napravit.

If you can meet with triumph and disaster
And treat those two imposters just the same . . .

If —
RUDYARD KIPLING

Z anglického originálu *Glisterings* [1] přeložil Jan Šustek.

¹Český název seriálu byl inspirován původní myšlenkou Petera Wilsona [pozn. překl.]

1. Testování nepovinného argumentu

Pokud definujeme nové makro, které má nepovinný argument, potřebujeme nějakým způsobem otestovat, zda je makro voláno s tímto nepovinným argumentem, zejména v případě, že argument má být ignorován, když není přítomen. Jedna možnost je použít makro \LaTeX `\@empty` jako implicitní hodnotu nepovinného argumentu

```
1 \newcommand{\moje}[2][\@empty]{%
2 % jestliže #1=\@empty nedělej nic
3 % jinak dělej něco
4 }
```

Můj první nápad, jak provést uvedený test, byl použít primitiv \TeX `\ifx`, který by provedl porovnání makra `\@empty` a skutečné hodnoty argumentu, podobně jako na následujících řádcích.

```
5 \newcommand{\testujneparg}[1][\@empty]{%
6   \ifx #1\@empty
7     Nepoužit nepovinný argument #1%
8   \else
9     Použit nepovinný argument #1%
10  \fi}
```

Toto makro ale bude dávat nesprávné výsledky:

```
11 \testujneparg                               Nepoužit nepovinný argument ()
12 \testujneparg[abcd]                         Použit nepovinný argument (abcd)
13 \testujneparg[aabb]                         bbNepoužit nepovinný argument (aabb)
```

Zdvořile mi bylo vysvětleno,² že `\ifx` testuje následující dva tokeny a že v \TeX je token buď řídicí sekvence (například `\@empty`) nebo jeden znak (například „a“). Na řádce 13 primitiv `\ifx` testuje tokeny „a“ a „a“, zjistí, že jsou stejné, a proto dostaneme nesprávný výsledek. Pokud na řádce 6 změním pořadí testovaných tokenů, dostaneme lepší výsledek.

```
14 \ifx\@empty#1
```

V tuto chvíli dostaneme

```
15 \testujneparg                               Nepoužit nepovinný argument ()
16 \testujneparg[\@empty]                     Nepoužit nepovinný argument ()
17 \testujneparg[]                             Použit nepovinný argument ()
18 \testujneparg[aabb]                         Použit nepovinný argument (aabb)
```

²Mezi jinými to byli Donald Arseneau, Michael Downes a Stephan Lemke.

2. Porovnávání řetězců

Obecnější problém podobného charakteru je testovat, zda dvě slova nebo dva řetězce jsou stejné. Zde také můžeme použít primitiv `\ifx`. Pokud použijeme `\ifx` k porovnání dvou tokenů typu řídicí sekvence, výsledek je pravda, pokud jsou řídicí sekvence definovány stejně a tokeny vzniklé po jejich první expanzi jsou stejné. Proto budeme definovat dvě makra, která se expandují na řetězce, a tato makra porovnáme.

```
19 \newif\ifstejne
20 \newcommand{\srovnejretezretez}[2]{%
21   \stejnefalse
22   \begingroup
23     \def\1{#1}\def\2{#2}%
24     \ifx\1\2\endgroup \stejnetrue
25     \else \endgroup
26     \fi}
```

Dva argumenty makra `\srovnejretezretez` jsou řetězce, které se mají porovnávat. Makro `\ifstejne` bude mít hodnotu pravda, pokud jsou shodné všechny znaky jednotlivých řetězců. Jestliže argumenty jsou makra, porovnávají se názvy těchto maker a ne jejich definice. Jestliže řetězce obsahují nějaké mezery, je každá skupina mezer považována za jedinou mezeru. Proto například `\srovnejretezretez{}{_}` definuje `\ifstejne` jako nepravda, ale `\srovnejretezretez{_}{_}` jako pravda.

```
27 \newcommand{\SrovnejRetezRetez}[2]{%
28   \lowercase{\srovnejretezretez{#1}{#2}}}
```

Makro `\SrovnejRetezRetez` při provádění testu nerozlišuje velikost znaků. Například nastaví `\ifstejne` na hodnotu pravda pro každý pár (abc, abc), (abc, Abc), (abc, aBc) a podobně. Makro používá primitiv `\lowercase`, který převede velká písmena na malá, a v době porovnávání makrem `\srovnejretezretez` už budou všechna písmena malá. Porovnávání nebude fungovat, pokud argumenty budou obsahovat makra, jejichž názvy se liší velikostí písmen, protože názvy maker primitiv `\lowercase` nemění.

Makra `\srovnejretezretez` a `\SrovnejRetezRetez` sice porovnávají řetězce přesně, jak jsme chtěli, ale ukážeme si ještě nějaká rozšíření.

Makro `\srovnejretezretez` využívá primitivu `\def`, který není expandovatelný, proto není například možné použít makro `\srovnejretezretez` uvnitř `\edef`. Victor Eijkhout [3, sekce 13.8.7] a David Kastrop [5] navrhli řešení tohoto problému. Jiné možné rozšíření je testovat, zda se makro expanduje na daný řetězec. Davidovo expandovatelné makro řeší i tento problém. Michael Downes [2] ukazuje trochu jinou metodu využívající `\expandafter`.

K porovnávání makra a řetězce můžeme použít makro `\srovnejretezretez`, ke kterému přidáme `\expandafter`.

```
29 \newcommand{\srovnejmakroretez}[2]{%
30   \expandafter\srovnejretezretez\expandafter{#1}{#2}}
```

První argument makra `\srovnejmakroretez` je buď řetězec, nebo makro, které se expanduje na řetězec. Druhý argument je řetězec.

I zde můžeme udělat test, který nerozlišuje velikost znaků.

```
31 \newcommand{\SrovnejMakroRetez}[2]{%
32   \lowercase{\srovnejmakroretez{#1}{#2}}}
```

Expandovatelné makro `\srovnejznakyznaky` je založeno na Viktorově kódu. Makro používá rekurzi k postupnému porovnávání dvojic znaků a k tomu potřebuje dvě pomocná makra.

```
33 \catcode'\^^G=11 % ^^G bude písmeno
34 \newcommand{\srovnejznakyznaky}[2]{%
35   \PokudZnaky#1^^G\Jsou#2^^G\Stejne}
```

Makro `\srovnejznakyznaky` přidá na konce argumentů značku, která ukončuje řetězce. Viktor používal znak `$`, což ale znamenalo, že testované řetězce nemohly znak `$` obsahovat. Rozhodl jsem se zvolit znak `^^G` (znak ASCII BEL zapsaný v \TeX ové notaci, který je za normálních okolností v \TeX u neplatný). Primitiv `\catcode` nejdříve nastaví znak `^^G`, aby se choval jako písmeno, a po provedení testu jej nastaví zpět. Následující makro, proložené komentářem, dělá většinu práce.

```
36 \def\PokudZnaky#1#2\Jsou#3#4\Stejne{%
37   \if#1^^G\if#3^^G\stejnetrue
38     \else\stejnefalse\fi
```

Makro vezme dvě dvojice argumentů oddělené tokeny `\Jsou` a `\Stejne`. První dvojice odpovídá prvnímu porovnávanému řetězci a druhá druhému řetězci. Konkrétněji `#1` je první znak prvního řetězce a `#2` obsahuje zbývající znaky prvního řetězce, včetně znaku `^^G`. Podobně pro argumenty `#3` a `#4`. Pokud jsme došli na konce obou řetězců, pak jsou řetězce stejné. Pokud jsme ale došli pouze na konec prvního řetězce, pak se řetězce liší. Pokud jsme nedošli na konec prvního řetězce, je třeba testovat dále.

```
39   \else\if#1#3\PokudZbytek#2\JeStejny#4%
40     \else\stejnefalse\fi\fi}
```

Pokud jsou odpovídající si znaky stejné, je třeba porovnat zbytky řetězců. Pokud znaky nejsou stejné, jsme hotovi.

Poslední z maker má tři argumenty, které jsou odděleny tokeny `\JeStejný`, `\else` a `\fi\fi`. První dva argumenty jsou řetězce, které se porovnávají, zatímco třetí argument se ignoruje.

```
41 \def\PokudZbytek#1\JeStejný#2\else#3\fi\fi{%
42   \fi\fi \PokudZnaky#1\Jsou#2\Stejně}
43 \catcode'\^G=15 % ^G bude neplatný znak
```

Toto makro jednoduše zavolá makro `\PokudZnaky` k porovnání zbyteků řetězců.

Makro `\srovnejznakyznaky` může být použito jako základ testů nerozlišujících velikost písmen, podobně jako u předchozích maker. Makro je expandovatelné. To znamená, že se porovnávání provádí už na úrovni expand processoru. Makro `\srovnejretezretez` k porovnávání potřebuje hlavní procesor. Další rozdíl je, že makro `\srovnejznakyznaky` ignoruje všechny mezery. Například `\srovnejznakyznaky{ab}{a b}` považuje argumenty za identické, avšak `\srovnejretezretez{ab}{a b}` je považuje za různé.

Seznam použité a doporučené literatury

- [1] Wilson, Peter. Glisterings. In *TUGboat*, Volume 22, Number 4, pp. 339–341, prosinec 2001. ISSN 0896-3207. URL: <http://tug.org/TUGboat/Articles/tb22-4/tb72wilson.pdf>
- [2] Downes, Michael J. *Re: catcodes for jobname macro — stupid question*. Post to `comp.text.tex` newsgroup, April 25, 2001.
- [3] Eijkhout, Victor. *TEX by Topic, A TEXnician's Reference*. Addison-Wesley, 1991. Available at <http://www.eijkhout.net/tbt/>. ISBN 0-201-56882-9.
- [4] Gibbons, Jeremy. Hey — It Works!. In *TEX and TUG NEWS*, Volume 2, Number 2, pp. 7–11, 1993.
- [5] Kastrop, David. *Completely expansible string comparison*. Post to `comp.text.tex` newsgroup, September 3, 2002.

Summary: It might work. I – String comparisons

The article deals with string comparisons at Plain `TEX` level. It's a translation of Peter Wilson's article, which first appeared in *TUGboat*, 22(4), 339–341, 2001.

Key words: `TEX`, Plain `TEX`, string comparisons.

Peter Wilson, herries.press@earthlink.net
18912 8th Ave. SW, Normandy Park,
WA 98166, United States of America