

Zpravodaj Československého sdružení uživatelů TeXu

Peter Wilson

Mělo by to fungovat X

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 30 (2020), No. 3-4, 168–176

Persistent URL: <http://dml.cz/dmlcz/150289>

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

Článek přináší ukázkou využití programovacího jazyka Lua ke zjištění počtu jednotlivých znaků v daném textovém souboru. Druhá část článku se zabývá možností změny rozvržení stránky v L^AT_EXu v souvislosti se sazbou vícestránkového seznamu obrázků.

Klíčová slova: Počet slov, četnost znaků, Lua, textový soubor, rozvržení strany, prostředí `quote`, prostředí `quotation`, balíček `changepage`, třída `memoir`, seznam obrázků

His eye, which scornfully glisters like fire,
Shows his hot courage and his high desire.

Venus and Adonis
WILLIAM SHAKESPEARE

Cílem tohoto seriálu je ukázat čtenáři krátké kousky kódu, které mohou vyřešit některé z jeho problémů. Doufám, že situaci ještě více nezkomplikuji v důsledku mých chyb. Opravy, poznámky a návrhy na změny budou vždy vítány.

Words are wise men's counters, they do but reckon
with them, but they are the money of fools.

Leviathan
THOMAS HOBBS

Počítání slov

Některé druhy publikací mohou být omezeny předepsaným počtem slov. Pak vystává otázka, jakým (nejlepším) způsobem můžeme slova spočítat. Toto detailně vysvětlují Otázky a odpovědi [1]. Nicméně je také možné jednoduše spočítat slova na jedné stránce rukopisu a výsledek vynásobit počtem stran. Tuto techniku, nazývanou *casting off*, používají designéři knih a vydavatelé při zpracování knihy, viz například [3, kap. 8]. Vydavatelé ani tak nepotřebují vědět přesný počet slov v knize, ale zajímá je počet stránek knihy po vysázení.

Z anglického originálu *Glisterings* [7] přeložil Jan Šustek. Všechny díly *Glisterings* byly také sepsány do knihy, kterou je možné zakoupit prostřednictvím TUG na stránce <https://tug.org/store/glisterings/> (pozn. redakce)

Když píšete diplomovou práci, je možné, že jste limitováni předem stanoveným rozsahem počtu slov. Přesto je pravděpodobné, že nikdo nebude kontrolovat přesný počet slov, pokud nebude na první pohled vidět nesoulad zadaného rozsahu a délky vaší diplomové práce. Otázkou ale je, co se počítá jako jedno slovo. Za kolik slov se počítá tabulka nebo obrázek? Když se používají písma v různých velikostech, mají slova stejnou váhu? A co složená slova, poznámky pod čarou, rovnice, webové adresy, verše? Za kolik slov se mají počítat?

V době, kdy čtete tyto řádky, je již LuaTeX na světě a možná jste jej i použili. V době, kdy jsem tyto řádky psal, byl LuaTeX ještě ve vývoji a sám jsem jej nezkoušel. Ale předpokládal jsem, že jazyk Lua [2] bude dostupný na všech platformách, kde je TeX, a proto jsem se jej rozhodl použít pro počítání slov.

Měl jsem to štěstí, že jsem mohl pracovat s ruční sazbu, podobně jako to dělával Gutenberg v 15. století. Na rozdíl od digitální sazby, kde je k dispozici neomezený počet jednotlivých znaků, je v ruční sazbě počet znaků jasně omezen. Pokud ve fontu máme pouze 23 tiskařských liter „e“, pak lze najednou vysázet text, v němž se znak „e“ nevyskytuje více než 23krát. Je proto důležité vědět, kolik jednotlivých znaků je třeba k vysázení jedné stránky textu. Kdysi jsem chtěl vysázet báseň ze 16. století (pouze dva verše na stránku) v konkrétním fontu, ale nemohl jsem, protože bych potřeboval jednu literu „h“ navíc (text obsahoval staroanglická slova thee, thou, thine, slova s příponou -eth apod.).

Pracuji v systému Linux, pro který existují programy na zjištění počtu slov a celkového počtu znaků v daném souboru. Dá se očekávat, že podobné programy existují i pro jiné operační systémy. Já jsem však potřeboval program, který zjistí počty jednotlivých znaků (počet znaků „A“, počet znaků „a“ atd.), proto jsem si jej zkusil vytvořit v jazyce Lua.

```
1 #!/usr/local/bin/luau5.1
2 local BUFSIZE = 2^13
3 local f = io.input(arg[1]) -- otevření vstupního souboru
4 local cc = 0 -- počet všech znaků
5 local lc = 0 -- počet řádků
6 local wc = 0 -- počet slov
7 local ct = {} -- počty jednotlivých znaků
8 for i = 33,126 do -- inicializace tabulky
9   ct[i] = 0
10 end
11 local tc = 0 -- počet znaků bez znaků konce řádku
12 while true do
13   -- načtení bloku textu
14   local lines, rest = f:read(BUFSIZE, "*line")
15   if not lines then break end
16   if rest then
```

```

17     lines = lines .. rest .. "\n" end
18     cc = cc + #lines
19     -- zjištění počtu slov v bloku
20     local _, t = string.gsub(lines, "%S+", "")
21     wc = wc + t
22     -- zjištění počtu konců řádku v bloku
23     _, t = string.gsub(lines, "\n", "\n")
24     lc = lc + t
25     -- tabulka počtu jednotlivých znaků
26     local K
27     for i = 1, string.len(lines) do
28         K = string.byte(lines,i)
29         if K >= 33 and K <= 126 then
30             ct[K] = ct[K] + 1
31         end
32     end
33 end
34 -- odstraní příponu vstupního souboru a nahradí ji příponou .gwc
35 base, ext = string.match(arg[1], "(%w+)%.(%w+)")
36 ofile = base .. ".gwc"
37 tc = cc - lc
38 io.output(ofile)
39 io.write("Informace o souboru ", arg[1], "\n")
40 io.write("", "pocet radku = ", lc, "\n",
41          "pocet slov = ", wc, "\n",
42          "pocet znaku = ", tc, "\n\n")
43 io.write("Jednotlive znaky\n")
44 for k,v in pairs(ct) do
45     if v > 0 then
46         print(string.char(k),v)
47         io.write(" ", string.char(k), " ",
48                 string.format("%4d",v), "\n")
49     end
50 end
51 print("Vystup ulozen do souboru ", ofile)

```

Pokud program uložíme do souboru `gwc.lua`, pak počty znaků (například v souboru `soubor.txt`) získáme jeho spuštěním z příkazového řádku:¹

```
52 gwc.lua soubor.txt
```

Tento program jako slovo počítá posloupnost tisknutelných znaků ukončenou jednou nebo více mezerami nebo koncem řádku. Program se zajímá pouze o znaky odpovídající literám ve fontu, který jsem měl k dispozici. Naštěstí v angličtině se stačí omezit na množinu tisknutelných znaků v ASCII. V případě jiných jazyků je třeba program rozšířit.² Část kódu byla převzata z [2, s. 198], kde je také kód detailně popsán.

Change is not made without inconvenience,
even from worse to better.

A Dictionary of the English Language: Preface
SAMUEL JOHNSON

Změna rozvržení stránky

Čas od času uživatelé položí otázku „Jak mohu změnit rozvržení konkrétní stránky?“, kde pod rozvržením mají na mysli rozměry nebo umístění sazby, různá záhlaví či zápatí.

Tvar stránky

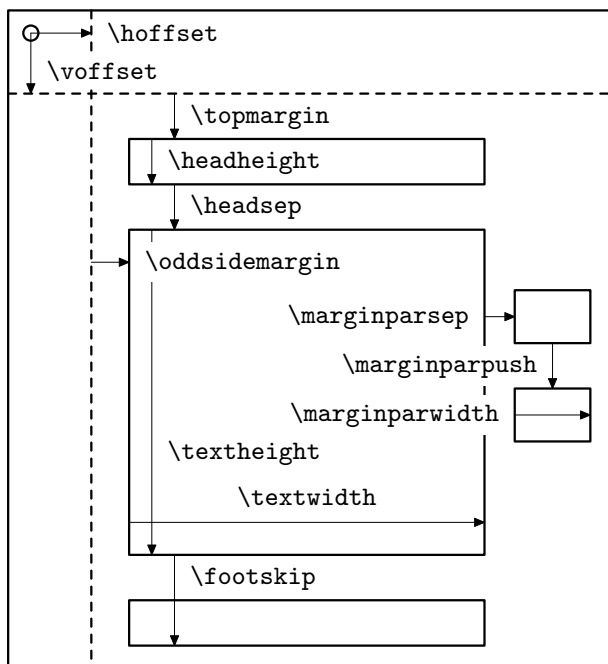
V \TeX u můžete udělat mnoho věcí, ale co nemůžete,³ je změnit šířku textu uvnitř odstavce. Například pokud by měl mít na první stránce text šířku 15 cm, zatímco na druhé stránce šířku 12 cm, pak odstavec, který začíná na první straně, bude pokračovat se šířkou 15 cm i na druhé straně. To je způsobeno tím, že \TeX nejdříve interně vysází odstavec podle aktuálně nastavené šířky textu. Poté určí, zda se odstavec vysází na aktuální stranu, nebo zda je třeba stránku zlomit. Pokud ke zlomu stránky dojde, vysází se část odstavce na první stranu a zbytek odstavce na druhou stranu, odstavec však již má řádky vysázené na původní šířku.

Na obrázku 1 jsou uvedeny parametry, které se používají při nastavení rozvržení stránky. Kroužkem je označen Knuthův bod o souřadnicích (1 in, 1 in) od levého horního okraje stránky.

¹V systému Linux musíme soubor `gwc.lua` nastavit jako spustitelný. Na řádku 1 pak musíme mít správně uvedenu cestu k interpretu jazyka Lua. Další možností je zavolat přímo interpret jazyka Lua, tj. zavolat `lua gwc.lua soubor.txt` (pozn. překl.)

²Pro soubor v češtině nebo slovenštině můžeme program jednoduše upravit, pokud soubor bude v některém osmibitovém kódování. V tom případě je třeba na řádcích 8 a 29 namísto hodnoty 126 psát hodnotu 255. Kódování výstupního souboru `soubor.gwc` bude stejně jako kódování vstupního souboru `soubor.txt`. (pozn. překl.)

³Čtenář, který nemá v \TeX u rád negativní odpovědi, si může přečíst článek [5], kde je podrobně rozebrán a vyřešen obecnější problém. Pokud se na stránce nevyskytují pružné vertikální mezery, je možné problém vyřešit vhodným použitím primitivu `\parshape`. (pozn. překl.)



Obrázek 1: \LaTeX ové parametry rozvržení liché (pravé) stránky

Ke změně výšky textového bloku na konkrétní stránce lze použít \LaTeX ového makra `\enlargethispage`. Toto makro má jeden délkový parametr, jehož hodnota se připočte k registru `\textheight`, a tato změna je platná lokálně pouze pro aktuální stranu – kladná hodnota parametru zvýší hodnotu `\textheight`, záporná ji sníží. Změna se projeví na spodní hraně textového bloku, horní hrana zůstane na stejném místě.

Prostředí `quote` a `quotation` dočasně změní velikost okrajů a šířku textu. Obecněji můžete šířku textu dočasně změnit pomocí prostředí `adjustwidth` z balíčku `changepage`. [6]

Prostředí `adjustwidth` se volá se dvěma délkovými parametry. Uvnitř prostředí jsou levý a pravý okraj zvětšeny o hodnoty příslušných parametrů. Například v tomto odstavci bylo použito

```
\begin{adjustwidth}{3cm}{1cm}
```

na začátku odstavce a podobně na konci odstavce bylo použito

```
\end{adjustwidth}
```

Aktuální stránka bude vysázena s hodnotami parametrů platnými v okamžiku, kdy je na stránku vložena první položka (například znak nebo box). Parametry, které se změni až poté, budou platné od následující stránky. Je však možné změnit parametry po vysázení jedné stránky před vložením první položky na následující stránku. Toho se využívá, když se v L^AT_EXu přepíná z jednosloupcové sazby do dvousloupcové a naopak. L^AT_EXová makra při tom přepočítají parametry rozvržení stránky. Obecný postup je, že se aktuální stránka ukončí a vysází, změni se parametry rozvržení, nastaví se nový počet sloupců sazby a tyto sloupce se již sázejí s novými hodnotami parametrů. Pokud pracujeme v jednosloupcovém dokumentu, postup je tento:

```
53 \clearpage
54 % změna parametrů
55 \onecolumn
```

Při změně parametrů na řádku 54 můžete použít také makra z L^AT_EXového balíčku changepage.

Aby bylo zřejmé, co makro `\onecolumn` dělá, uvádím zde jeho definici:

```
56 \def\onecolumn{%
57   \clearpage
58   \global\columnwidth\textwidth
59   \global\hsize\columnwidth
60   \global\linewidth\columnwidth
61   \global\@twocolumnfalse
62   \col@number \@ne
63   \@floatplacement}
```

Definice makra `\twocolumn` je složitější. Mimo jiné má toto makro jeden nepovinný parametr⁴, který však nemá vliv na rozložení sazby.

Jako příklad si ukážeme, jak je možné zvětšit nebo zmenšit výšku a šířku textového bloku na některé skupině stránek.

```
64 \newcommand*{\addtotextheightwidth}[2]{%
65   \clearpage
66   \addtolength{\textheight}{#1}
67   \addtolength{\textwidth}{#2}
68   \onecolumn}
```

Až budeme chtít změnu provést, například pro stránky s přílohami, zavoláme

```
69 \addtotextheightweight{-1cm}{1cm}
```

⁴Obsah parametru je vysázen v jednosloupcovém režimu před započítáním dvousloupcové sazby. (pozn. překl.)

Záhlaví a zápatí

Další typ dotazu na rozvržení stránky, který jsem viděl, se týkal umístění slova „strana“ do záhlaví nad čísla stran ve vícestránkovém obsahu nebo seznamu obrázků. Řekněme, že chceme seznam obrázků vysázet tak, že v záhlaví stránky se seznamem bude slovo „obrázek“ zarovnané doleva a slovo „strana“ zarovnané doprava. Toto by se mělo opakovat na všech stranách seznamu obrázků. Dále chceme, aby stránka se seznamem obrázků měla své číslo zarovnané na střed v zápatí, podobně jako v L^AT_EXovém stylu stránky `plain`. Podobné požadavky budeme mít pro vysázení obsahu a seznamu tabulek. Zde si však ukážeme pouze sazbu seznamu obrázků.

I když ke změně stylu stránky můžete použít také balíček `fancyhdr` [4], zde si ukážeme třídu `memoir`, která nabízí podobné možnosti.

Třída `memoir` umožňuje definovat libovolné množství stylů. Potřebujeme vytvořit styl stránky pro navazující stránky v seznamu obrázků (a podobné styly pro obsah a pro seznam tabulek). Styl stránky pojmenujeme `lof`, záhlaví nastavíme tak, že vlevo bude slovo „obrázek“ a vpravo slovo „strana“, zatímco v zápatí uprostřed bude číslo stránky.

```
70 \makepagestyle{lof}
71 \makeevenhead{lof}{obrázek}{}{strana}
72 \makeoddhead{lof}{obrázek}{}{strana}
73 \makeevenfoot{lof}{}{\thepage}{}
74 \makeoddf脚{lof}{}{\thepage}{}
```

Musíme zajistit, že se styl `lof` nastaví během sazby seznamu obrázků. Toho docílíme tak, že nastavení stylu přidáme k definici makra `\listoffigures`. Ve třídě `memoir` je pro tyto účely definováno makro `\addtodef`. Makro má tři parametry – prvním je název upravovaného makra, druhým je T_EXový kód, který se přidá na začátek definice, a třetím T_EXový kód, který se přidá na konec definice.

```
75 \addtodef{\listoffigures}{%
76 \clearpage\pagestyle{lof}}{}
```

Ve třídě `memoir` je definováno makro, které se zavolá těsně před vysázením nadpisu seznamu obrázků, a makro, které se zavolá těsně po vysázení. Předefinováním těchto maker nastavíme, že první stránka seznamu obrázků bude mít prázdné záhlaví a že se výše uvedený text záhlaví vysází pod nadpis.

```
77 \renewcommand*{\afterloftitle}{%
78 \thispagestyle{plain}
79 \par\nobreak
80 {\normalfont\normalsize obrázek\hfill strana}
81 \par\nobreak}
```


Nastavení sazby obsahu a seznamu tabulek je prakticky identické, pouze patričným způsobem změním názvy použitých maker.

Je třeba si ještě dát pozor na to, že po vysázení seznamu obrázků zůstane nastavený styl `lof`. Proto je nutné po zavolání makra `\listoffigures` nastavit zpět původní styl. Předpokládejme, že základní styl dokumentu se jmenuje `mujstyl`. Pak můžeme psát

```
82 \documentclass[...]{memoir}
83 %% nastavení stylu mujstyl
84 %% úpravy \listoffigures apod. (řádky 70 až 81)
85 \pagestyle{mujstyl}
86
87 \begin{document}
88 %% úvodní strany, předmluva atd.
89 \tableofcontents
90 \clearpage\pagestyle{mujstyl}
91 \listoffigures
92 \clearpage\pagestyle{mujstyl}
93 \listoftables
94 \clearpage\pagestyle{mujstyl}
95 ...
96 \end{document}
```

Seznam literatury

- [1] Robin Fairbairns. *texfaq. A compilation of Frequently Asked Questions with answers*. Version 3.28. 10. čvn. 2014. URL: <https://www.ctan.org/pkg/texfaq>.
- [2] Roberto Ierusalimsky. *Programming in Lua*. 2. vyd. Rio de Janeiro, 2006.
- [3] Ruari McLean. *The Thames and Hudson Manual of Typography*. Thames and Hudson, 1980.
- [4] Pieter van Oostrum. *fancyhdr. Extensive control of page headers and footers in L^AT_EX 2_ε*. Version 4.0.1. 28. led. 2021. URL: <https://www.ctan.org/pkg/fancyhdr>.
- [5] Jan Šustek. „Sazba odstavců do textových oblastí“. In: *Zpravodaj C_STUG 19.3* (2009), s. 124–137.
- [6] Peter Wilson. *changepage. Margin adjustment and detection of odd/even pages*. Version 1.0c. 20. říj. 2009. URL: <https://www.ctan.org/pkg/changepage>.
- [7] Peter Wilson. „Glisterings“. In: *TUGboat 31.1* (2010), s. 90–93.
- [8] Peter Wilson. *memoir. Typeset fiction, non-fiction and mathematical books*. Version 3.7n. 4. říj. 2020. URL: <https://www.ctan.org/pkg/memoir>.

Summary: It Might Work X

This paper shows an example how to use Lua program to find the number of particular characters in a given text file. The second part of the paper shows ways to change the page layout in L^AT_EX, with applications for typesetting a multi-page list of figures.

Keywords: Word counting, character frequency, Lua, text file, page layout, quote environment, quotation environment, changepage package, memoir class, list of figures

*Peter Wilson, herries.press@earthlink.net
18912 8th Ave. SW
Normandy Park, WA 98166 USA*