

Zpravodaj Československého sdružení uživatelů TeXu

Peter Wilson

Mělo by to fungovat XIII

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 33 (2023), No. 1-2, 54–60

Persistent URL: <http://dml.cz/dmlcz/151759>

Terms of use:

© Československé sdružení uživatelů TeXu, 2023

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>

Článek ukazuje několik způsobů, jak lze v \LaTeX u použít verbatim text uvnitř argumentu některých maker. Dále jsou ukázány možnosti, jak zkrátit sazbu textu na danou šířku či výšku.

Klíčová slova: Makra, verbatim argumenty, zkracování textu, \LaTeX

... Cloath'd all in glistering coats,
which made a shew...

Poems and Fancies
MARGARET CAVENDISH

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.

Sir, I have found you an argument,
but I am not obliged to find you an understanding.
SAMUEL JOHNSON

1. Verbatim argumenty

Nedávno jsem byl upozorněn na problém, že verbatim text nemůže být použitý jako argument makra. Když například chceme vysázet text v zarámované `minipage`, lze to udělat jednoduše použitím prostředí `minipage` uvnitř argumentu makra `\fbox`.

```
1 \fbox{\begin{minipage}{0.97\columnwidth}  
2   Text uvnitř zarámované minipage  
3 \end{minipage}}
```

Toto funguje bez problémů až do doby, kdy uvnitř `minipage` použijeme verbatim text. Potom dostaneme zmatené chybové zprávy, a to i přesto, že bez použití rámečku vše uvnitř `minipage` funguje.

\LaTeX disponuje makrem `\newsavebox`, s jehož pomocí lze uložit vysázený text a ten pak použít prostřednictvím makra `\usebox`.

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

Zde je uvedena definice prostředí `framedminipage`, které umožňuje použít verbatim text uvnitř rámečku.¹

```
4 \newsavebox{\minibox}
5 \newenvironment{framedminipage}[2][c]{%
6   \begin{lrbox}{\minibox}
7     \begin{minipage}[#1]{#2}}%
8   {\end{minipage}\end{lrbox}}
9   \noindent\fbbox{\usebox{\minibox}}}
```

Při sazbě tohoto rámečku mělo prostředí `minipage` šířku rovnu 97 % šířky okolní sazby.

```
10 \begin{framedminipage}{0.97\columnwidth}
11   ...
12 \end{framedminipage}
```

Prostředí `lrbox` je analogií L^AT_EXových maker `\savebox` a `\sbox`. Zatímco dovnitř argumentu makra nelze přímo vložit verbatim text, dovnitř těla prostředí jej vložit lze. V uvedeném kódu se nejprve deklaruje registr `\minibox` typu `box`, do nějž lze uložit vysázený text. Dále je definováno prostředí `framedminipage`, a to podobně jako prostředí `minipage`, včetně nepovinného argumentu určujícího vertikální umístění sazby. Na začátku prostředí `framedminipage` se zahájí prostředí `lrbox` a `minipage` a na jeho konci se zase obě tato prostředí ukončí. Vysázený obsah `minipage` se tak uloží do registru `\minibox`, přičemž si musíme uvědomit, že verbatim text už je uvnitř `\minibox` vysázený. Poté se obsah registru `\minibox` jen vloží do makra `\fbbox`.

T_EXbook [2, str. 363] definuje makro `\footnote` tak, aby jeho argument mohl obsahovat verbatim text. Knuth píše, že makro je propracované a používá několik triků. Já tomu makru nerozumím, ale ukážu zde jeho hlavní myšlenku. Pro jednoduchost makro `\verbtext` pouze vysází svůj argument. Nejsem si jistý, zda jsem správně umístil makra `\color@...`, protože nic podobného v původních Knuthových makrech nebylo.

```
13 \makeatletter
14 \long\def\verbtext{\vtintro\futurelet\next\vte@t}
15 \def\vte@t{\ifcat\bgroup\noexpand\next
16   \let\next\vt@@t
17   \else \let\next\vt@t\fi \next}
18 \def\vt@@t{\bgroup\aftergroup\vtend\let\next}
19 \def\vt@t#1{%
```

¹V původním článku nebyl na řádce 9 použit příkaz `\noindent`, což mělo za následek, že makro `\fbbox` při přechodu z vertikálního módu zahájilo odstavec a vložilo odstavcovou zarážku, což posunulo celý rámeček doprava. (pozn. překl.)

```

20 \color@begingroup
21 #1\vtmid
22 \color@endgroup}
23 \let\vtintro\relax
24 \let\vtmid\relax
25 \let\vtend\relax
26 \makeatother

```

Makra `\vtintro` a `\vtend` se expandují před a za argumentem a jejich nadefinováním je možné dělat různé triky. Může se stát, že i definovat makro `\vtmid` bude užitečné.²

Jednoduché použití makra `\verbtext` může být

```

27 \verbtext{Argument makra \verb-\verbtext- může
28 obsahovat \verb-\verb- text.}

```

s očekávaným výsledkem

Argument makra `\verbtext` může obsahovat `\verb` text.

Následující kód ukazuje použití maker `\vtintro` a `\vtend` pro sazbu textu kapitálkami.

```

29 \makeatletter
30 \newcommand*{\fred}[1][\@empty]{Frederick%
31 \ifx\@empty #1\else~#1\fi}
32 \makeatother
33 \def\vtintro{\begingroup\scshape}
34 \def\vtend{\endgroup}
35 \verbtext{Makro \verb-\fred[III]-
36 vypíše \uv{\fred[III]}, zatímco makro
37 \verb-\fred- vypíše pouze \uv{\fred}.}

```

²Na řádce 15 se testuje, zda je argument makra `\verbtext` uzavřen do složených závorek (tj. za tokenem `\verbtext` následuje otevírací závorka), nebo zda je argumentem jediný token neuzavřený do složených závorek (v tomto případě samozřejmě nemůže nastat situace, že by argument obsahoval verbatim text). Na základě toho se pak provede buď makro `\vt@@t`, nebo makro `\vt@t`.

V makru `\vt@@t` se pomocí `\bgroup` otevře skupina, která se pak uzavře uživatelskou pravou závorkou. Nastaví se, že po uzavření této skupiny se provede `\vtend`. Konstrukce `\let\next` odstraní uživatelskou levou závorku. Poté se běžným způsobem načte a zpracuje vnitřek uživatelských složených závorek.

Makro `\vt@t` by mělo provést totéž, avšak bez nutnosti práce se skupinami. Pro stejnou funkci jako makro `\vt@@t` by stačilo definovat `\def\vt@t#1{#1\vtend}`.

Autor zřejmě chtěl ukázat, že je možné definovat různé chování makra na základě způsobu jeho zavolání. (pozn. překl.)

MAKRO `\fred[III]` VYPÍŠE „FREDERICK III“, ZATÍMCO MAKRO `\fred` VYPÍŠE POUZE „FREDERICK“.

V tomto konkrétním příkladě by šlo jednodušeji psát

```
38 {\scshape\verbtext{...}}
```

a nemuseli jsme se vůbec zatěžovat makry `\vtintro` a `\vtend`. Nicméně se určitě najdou situace, kdy se tato makra budou hodit.

Wickedness is always easier than virtue;
for it takes a short cut to everything.

SAMUEL JOHNSON

2. Zkracování textu

Další z dotazů na `comp.text.tex` se týkal problému, kdy je třeba dlouhý text zkrátit například na dva nebo tři řádky.

Již v tu dobu existoval balíček `truncate` Donalda Arseneaua [3], který umožňuje zkrátit text na danou šířku. Na konci zkráceného textu se vytiskne „...“ (`\ldots`), případně jiná značka, aby se čtenář dozvěděl, že text ještě pokračuje. Například

```
39 \truncate{0.9\columnwidth}{Balíček \textsf{truncate} definuje  
40 makro pro zkracování textu tak, aby nepřekročil zadanou šířku.}
```

vysází

Balíček `truncate` definuje makro pro zkracování textu tak, aby...

Nicméně dotaz se týkal vertikální analogie makra `\truncate`. V odpovědi [4] na dotaz Donald definoval makro `\vtruncate` následovně.

```
41 \newsavebox\descbox  
42 \newsavebox\partialbox  
43 \newcommand{\vtruncate}[2]{%  
44   \setbox\descbox\vbox{#{2}\par}}%  
45   \setbox\partialbox\vsplit\descbox to #1\relax  
46   \vtop{\unvbox\partialbox}%  
47   % or use  
48   % \par\unvbox\partialbox  
49 }
```

Makro má dva argumenty. Druhým argumentem je text, který se zkrátí na výšku danou prvním argumentem.

V další odpovědi [5] Will Robertson definoval prostředí `cutlines`.

```

50 \makeatletter
51 \newbox\cut@desc
52 \newenvironment{cutlines}[1][2]{%
53   \@tempcnta=#1\relax
54   \setbox\cut@desc\vbox\bgroup
55   \parskip=0pt}{%
56   \egroup
57   \vsplit\cut@desc to \@tempcnta\baselineskip}
58 \makeatother

```

Prostředí má jeden nepovinný argument (uzavřený do hranatých závorek), který udává, na kolik řádků se má text uvnitř prostředí zkrátit.

Vyzkoušel jsem obě dvě řešení a narazil jsem na následující možné problémy.

1. Textový argument makra `\vtruncate` nemůže obsahovat žádný verbatim text. (To nemusí až tolik vadit.)
2. Jestliže v prostředí `cutlines` je zadán počet řádků větší než počet řádků textu, potom je pod text vložena vertikální mezeru taková, aby celková výška odpovídala zadanému počtu řádků.
3. V obou řešeních se stává, že výsledná výška je jiná, než bylo zadáno. Vždy se však liší maximálně o jeden řádek. Zdá se, že `cutlines` je přesnější než `\vtruncate`.
4. Zkrácený text je umístěn ve `\vboxu`, v němž nemůže dojít ke stránkovému zlomu.

Po dlouhém ladění jsem dospěl k definici prostředí `vcutlines`, které je kombinací obou řešení a které řeší první dva problémy a možná i třetí problém. Čtvrtý problém se týká všech řešení.³ Prostředí `vcutlines` má jeden nepovinný argument, který udává, na jakou výšku se má text uvnitř prostředí zkrátit.

```

59 \newsavebox\descbox
60 \newsavebox\partialbox
61 \newlength{\vcutl}% for the limit height
62 \newlength{\Vcutl}% height of full text
63 \newenvironment{vcutlines}[1][2\baselineskip]{%
64   \setlength{\vcutl}{#1}%
65   \setbox\descbox\vbox\bgroup
66   \parskip=0pt\relax
67   }{%
68   \egroup
69   \Vcutl=\ht\descbox
70   \advance\Vcutl \dp\descbox

```

³I původní článek obsahoval řádek 48. Pokud by se tento řádek použil namísto řádků 46 a 72, pak by ke čtvrtému problému nedošlo. (pozn. překl.)

```

71 \setbox\partialbox\vsplit\descbox to \vcutl\relax
72 \vtop{\unvbox\partialbox}
73 \ifdim \vcutl<\Vcutl \vtruncont \fi}
74 \newcommand*{\vtruncont}{\noindent\strut\ldots}

```

V následujících příkladech pro testování použijeme text

```

75 {\itshape Donald Arseneau vytvořil makro \verb-\vtruncate- a
76 Will Robertson vytvořil prostředí \verb-cutlines-. Obě řešení
77 zkrátí text, pokud by po vysázení měl větší než zadanou výšku.
78 V tomto článku je vytvořeno nové prostředí \verb-vcutlines-,
79 které je kombinací obou předchozích řešení.}

```

který obsahuje i několik verbatim textů.

Pro začátek vyzkoušejme prostředí `vcutlines` s omezením na 20 řádků (tj. na výšku `20\baselineskip`).

Donald Arseneau vytvořil makro `\vtruncate` a Will Robertson vytvořil prostředí `cutlines`. Obě řešení zkrátí text, pokud by po vysázení měl větší než zadanou výšku. V tomto článku je vytvořeno nové prostředí `vcutlines`, které je kombinací obou předchozích řešení.

A nyní stejný text s omezením na 3 řádky.

Donald Arseneau vytvořil makro `\vtruncate` a Will Robertson vytvořil prostředí `cutlines`. Obě řešení zkrátí text, pokud by po vysázení měl větší než zadanou výšku. V tomto článku je vytvořeno nové prostředí `vcutlines`, které je kombinací ...

Jestliže dojde ke zkrácení textu, vloží se na konec prostředí makro `\vtruncont`, které implicitně vloží řádek obsahující pouze „...“ (`\ldots`). Pro zjištění, zda došlo ke zkrácení textu, se výška celého vysázeného textu porovnává se zadanou výškou.

Makro `\vtruncont` můžeme například nadefinovat na nic.

```

80 \renewcommand*{\vtruncont}{}

```

čimž dostaneme

Donald Arseneau vytvořil makro `\vtruncate` a Will Robertson vytvořil prostředí `cutlines`. Obě řešení zkrátí text, pokud by po vysázení měl větší než zadanou výšku. V tomto článku je vytvořeno nové prostředí `vcutlines`, které je kombinací V tom případě se ale čtenář nedozví, zda text ještě pokračuje, nebo ne.

Odkazy

1. WILSON, Peter. Glisterings. *TUGboat* [online]. 2011, roč. 32, č. 3, s. 339–341 [vid. 2011-12]. Dostupné z: <https://tug.org/TUGboat/tb32-3/tb102glisters.pdf>.

2. KNUTH, Donald E. *The T_EXbook*. Sv. A. Reading, MA: Addison-Wesley, 1984. Computers & Typesetting. V současnosti jsou dostupné 35. výtisk (měkká vazba, 2017) a 23. výtisk (pevná vazba, 2021).
3. ARSENEAU, Donald. `truncate.sty`: *Truncate text to a specified width* [online]. CTAN, 2001 [vid. 2011-12]. Dostupné z: <https://ctan.org/pkg/truncate>. Verze 3.6.
4. ARSENEAU, Donald. *Re: How to limit/cut off text after a number of lines?* `comp.text.tex` newsgroup, 2008-07-16.
5. ROBERTSON, Will. *Re: How to limit/cut off text after a number of lines?* `comp.text.tex` newsgroup, 2008-07-16.

Summary: It Might Work XIII

This paper shows several ways in L^AT_EX how to use verbatim text within argument of some macros. We also show how to truncate text to a given width or height.

Keywords: Macros, verbatim arguments, truncating text, L^AT_EX

*Peter Wilson, `herries.press@earthlink.net`
12 Sovereign Close
Kenilworth, CV8 1SQ, UK*