

# Zpravodaj Československého sdružení uživatelů TeXu

---

Peter Wilson; Jan Šustek

Mělo by to fungovat VI - Odstavce

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 27 (2017), No. 3-4, 155–164

Persistent URL: <http://dml.cz/dmlcz/150281>

## Terms of use:

© Československé sdružení uživatelů TeXu, 2017

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 ukazuje možnosti nastavení  $\text{\TeX}$ ových odstavcových parametrů pro různá i méně typická zarovnání odstavce.

**Klíčová slova:**  $\text{\LaTeX}$ , balíček `shapepar`, odstavce, zarovnání.

Whose waves do glister by the Queen's bright beams.  
Which makes them murmur as they passe away.

*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.

Poslední dobou se ve skupině `comp.text.tex` vyskytla spousta otázek o formátování odstavců.

for life's not a paragraph  
and death I think is no parenthesis.

*Since feeling is first*

E. E. CUMMINGS

## 1. Běžné formátování odstavců

Běžný odstavec vysázený v  $\text{\LaTeX}$ u vypadá následovně.

Běžný odstavec je v  $\text{\LaTeX}$ u vysázený stejně jako tento. První řádek je zleva odsazený, další řádky jsou zarovnány do bloku a poslední řádek je zprava nezarovnaný.

Odsazení prvního řádku odstavce je dáno hodnotou `\parindent`. Pokud chceme mimořádně začít odstavec bez odsazení prvního řádku, použijeme na začátku odstavce příkaz `\noindent`.

---

Z anglického originálu *Glisterings* [8] přeložil Jan Šustek.

Další běžně používaná zarovnání odstavců jsou zarovnání doleva, zarovnání doprava a zarovnání na střed.

V některých případech je vhodné vytvořit odstavec s více než jedním odsazeným řádkem.  $\text{\TeX}$  pro tyto případy nabízí dva příkazy. (Balíček `hanging` [7] toto řeší formou  $\text{\LaTeX}$ ového prostředí.)

Šířku odsazení určuje příkaz `\hangindent<šířka>`, zatímco počet odsazených řádků určuje příkaz `\hangafter<číslo>`. Pokud je `<číslo>` kladné, pak budou odsazeny řádky `<číslo> + 1`, `<číslo> + 2` a další, zatímco pokud je `<číslo>` záporné, pak je odsazeno prvních `-<číslo>` řádků. Pokud je `<šířka>` kladná, pak jsou řádky odsazeny zleva o `<šířka>`, pokud je záporná, pak jsou řádky odsazeny zprava o `-<šířka>`. Například při nastavení

```
1 \hangindent=3pc \hangafter=-2
```

dostaneme následující výsledek

První dva řádky jsou odsazeny  
zleva o 3pc a další řádky odsazeny  
nejsou. První řádek je navíc odsazen ještě  
o `\parindent`. Tyto příkazy můžete použít jak  
přímo v  $\text{\TeX}$ u, tak také v  $\text{\LaTeX}$ u.

Příkazy `\hangindent` a `\hangafter` je nutné použít opakovaně pro každý odstavec, který má být odsazený.

$\text{\LaTeX}$  disponuje interním makrem `\@hangfrom`, které používá pro několik účelů, například pro sazbu nadpisů. Zde uvádím čitelnější verzi tohoto makra.

```
2 \makeatletter % pokud není v souboru .cls nebo .sty
3 \newcommand*{\hangfrom}[1]{%
4   \setbox\@tempboxa\hbox{#{1}}}%
5   \hangindent \wd\@tempboxa
6   \noindent\box\@tempboxa}
7 \makeatother % pokud není v souboru .cls nebo .sty
```

Makro `\hangfrom{<text>}` použité na začátku odstavce způsobí, že druhý a další řádek odstavce budou odsazeny o přirozenou šířku parametru `<text>`. Použití si ukážeme na příkladu.

```
8 \hangfrom{$\Longrightarrow$\quad}Tento odstavec má řádky
9   odsazené podle šířky argumentu makra \p{hangfrom}.
10  To se může někdy hodit.
```

$\Rightarrow$  Tento odstavec má řádky odsazené podle  
šířky argumentu makra `\hangfrom`. To  
se může někdy hodit.

T<sub>E</sub>X umožňuje vysázet odstavce i velmi netypických tvarů.

Libovolný tvar odstavce  
můžeme v T<sub>E</sub>Xu nastavit pří-  
kazem `\parshape`. Pro L<sup>A</sup>T<sub>E</sub>Xové  
uživatele Donald Arseneau vy-  
tvořil balíček `shapepar` [1],  
který umož-ňuje speci-  
ální tvary odstavce na-  
stavit příjemnějším způso-  
bem. Takové tvary odstavce  
však doporu-čuji používat  
pouze velmi zřídka a pouze  
v případě, že ta-kový tvar od-  
stavce opravdu musíme použít. Tento  
odstavec byl vysázen s pomocí  
balíčku `shapepar` s nastavením  
tvaru odstavce `\nutshape`.

Balíček `shapepar` má předdefinováno několik tvarů odstavce a obsahuje makra, která automaticky vygenerují příslušné parametry odstavce.

For precept must be upon precept; precept upon  
precept; line upon line; line upon line; here a little,  
and there a little.

*Isiah*

CH. 28, v. 10

## 2. Konkrétní formátování odstavců

Kromě příkazů `\hangafter`, `\hangindent` a `\parshape` má T<sub>E</sub>X další čtyři parametry, které ovlivňují tvar běžných odstavců. Délkový registr `\parindent` nastavuje odsazení prvního řádku odstavce. Mezery velikosti `\leftskip` a `\rightskip` jsou vloženy na začátek a na konec každého řádku. Mezera velikosti `\parfillskip` je vložena na konec posledního řádku odstavce. Poslední tři uvedené příkazy mají charakter pružné délky a jejich změnou dostaneme konkrétní zarovnání odstavce.

Předpokládejme, že jsme si nastavili následující délkové registry.

```
11 \newskip\Zeroskip \Zeroskip=0pt plus 0pt minus 0pt
12 \newskip\Flushglue \Flushglue=0pt plus 1fil
```

L<sup>A</sup>T<sub>E</sub>X definuje makra `\raggedright`, `\raggedleft` a `\centering`, která nastavují následující hodnoty mezer `\leftskip`, `\rightskip` a `\parfillskip`.

makro	(implicitně)	<code>\raggedright</code>	<code>\raggedleft</code>	<code>\centering</code>
<code>\leftskip</code>	<code>\Zeroskip</code>	<code>\Zeroskip</code>	<code>\Flushglue</code>	<code>\Flushglue</code>
<code>\rightskip</code>	<code>\Zeroskip</code>	<code>\Flushglue</code>	<code>\Zeroskip</code>	<code>\Flushglue</code>
<code>\parfillskip</code>	<code>\Flushglue</code>	<code>\Zeroskip</code>	<code>\Zeroskip</code>	<code>\Zeroskip</code>

S uvedenými mezerami si můžeme pohrát a docílit toho, že vnitřní řádky odstavce budou formátovány jinak než první nebo poslední řádek.

Následující makro `\poslednivpravo` zarovná odstavec do bloku a poslední řádek zarovná doprava.

```

13 \newcommand*{\poslednivpravo}{%
14   \leftskip=0pt plus 1fil
15   \rightskip=-\leftskip
16   \parfillskip=\leftskip
17   \parindent=0pt}

```

Tento odstavec není moc zajímavý. Je zarovnaný vlevo i vpravo. Výjimkou je poslední řádek, který je zarovnaný vpravo.

Makro `\posledninastred` zarovná odstavec do bloku a poslední řádek zarovná na střed.

```

18 \newcommand{\posledninastred}{%
19   \leftskip=0pt plus 1fil
20   \rightskip=0pt plus -1fil
21   \parfillskip=0pt plus 2fil\relax}

```

Všechny řádky tohoto odstavce jsou zarovnané vlevo i vpravo. Výjimkou je první řádek, který je odsazený, a poslední řádek, který je zarovnaný na střed.

Pokud nechceme mít první řádek odstavce odsazený, použijeme na začátku textu odstavce `\noindent`.

Makro `\prvnivlevopotomvpravo` nastaví, že první řádek odstavce bude zarovnaný vlevo, zatímco ostatní řádky odstavce budou zarovnány vpravo. Tento tvar odstavce se mi však zdá nepřírozený.

```

22 \newcommand*{\prvnivlevopotomvpravo}{%
23   \leftskip=0pt plus 1fill

```

```

24 \rightskip=0pt plus 1fil
25 \parfillskip=0pt
26 \everypar{\hskip 0pt plus -1fill\relax}%
27 \parindent=0pt}

```

Tento odstavec je podivně zarovnaný. Jeho první řádek je zarovnaný vlevo. Další řádky včetně posledního jsou zarovnané vpravo.

Množství způsobů různých zarovnání odstavce je dáno množstvím kombinací, které lze vytvořit z mezer (Donald Knuth používá pojem „glue“) vkládaných na kraje řádků. Detailní popis datového typu glue je v *T<sub>E</sub>Xbooku* [5, kap. 12], případně v knize Victora Eijkhouta [3, kap. 8].

V definici makra `\prvniivlevoptomvpravo` má mezera `\leftskip` pružnost `1fill`, což je nekonečněkrát více než pružnost `\rightskip`, která je `1fil`. To znamená, že se řádek zarovná doprava. V posledním řádku odstavce má mezera `\parfillskip` hodnotu `0pt` a to nijak neovlivní konec řádku. První řádek odstavce má pružnost `-1fill` z `\everypar` a pružnost `1fill` z `\leftskip`. Tyto pružnosti se pokrátí a zůstane pružnost `1fil` z `\rightskip`. To znamená, že první řádek se zarovná doleva.

Nikos Platis [6] potřeboval, aby bylo posledních několik slov odstavce zarovnaných doprava a pokud by vpravo nebyl dostatek místa, pak aby se tato slova přesunula na další řádek a zarovнала doprava.

Krátký řádek.	Text vpravo
Delší řádek než v prvním případě.	
Text vpravo	

Existuje více řešení tohoto problému. V Nikosových pokusech se jako nejlepší ukázalo řešení, které zaslal Dirk Schlimm a které původně navrhl Donald Knuth v *T<sub>E</sub>Xbooku* [5, str. 106]. Po převedení do *L<sup>A</sup>T<sub>E</sub>Xu* uvedené řešení vypadá následovně.

```

28 \newcommand*{\vpravo}[1]{\{%
29 \unskip\nobreak\hfil\penalty50
30 \hskip2em\hbox{}\nobreak\hfil#1%
31 \parfillskip=0pt\finalhyphendemerits=0\par}}

```

Makro `\vpravo` se používá takto.

```

32 Krátký řádek. \vpravo{Text vpravo}

```

Dalším častým požadavkem je, aby poslední řádek odstavce nebyl moc krátký. Pro tyto účely je možné použít makro `\nemockratky`, které zajistí, že poslední řádek bude alespoň `\minimum` dlouhý.

```

33 \newdimen\delka
34 \newdimen\minimum
35 \minimum=2em
36 \newcommand*{\nemockratky}{%
37   \delka=\hsize
38   \advance\delka -\minimum
39   \leftskip=0pt plus 0pt minus 0pt
40   \rightskip=\leftskip
41   \parfillskip=\delka minus \delka
42   \parindent=2em}

```

Poslední řádek tohoto odstavce by neměl být kratší než délka, kterou si nastavíme.

Poslední řádek tohoto odstavce by neměl být kratší než délka, kterou si předtím nastavíme.

U krátkých odstavců, jako jsou tyto, účinek makra tolik nevynikne. Lépe je to vidět u delších odstavců.

Peter Flynn [4] odpověděl na níže uvedenou otázku uživatele Marka a naprogramoval makra pro zarovnání, které nazval „pružný okraj“.

Ahoj, rád	...a nějaký text tady.
bych měl na	Zkoušel jsem
jednom řádku	<b>tabularx</b> a <b>TabularC</b> ,
text	ale tato prostředí
zarovnaný	nejsou dostatečně
vlevo	přesná na to, aby byl
i vpravo,	text zarovnán až
například	k okraji. Máte nějaký
nějaký text	nápad? Díky, Mark
tady...	

Tento odstavec byl vytvořen makrem

```

43 \pruzny{0.3}{0.5}{Ahoj, rád bych měl...}%
44   {\dots a nějaký text...}

```

kde `\pruzny` je drobně upravené Peterovo makro. První dva argumenty určují poměry celkové šířky řádku pro levý a pravý sloupec. Jejich součet musí být menší než jedna. Zbývající dva argumenty jsou texty, které se mají sázet.

```

45 \newcommand{\pruzny}[4]{%
46   \par\noindent\hbox to\columnwidth{\vtop{%

```

```

47 \hsize=#1\columnwidth\flushleft#3\par}\hss
48 \vtop{\hsize=#2\columnwidth\flushright#4\par}}}
```

Četl jsem právnické dokumenty, kde byl každý odstavec zakončen čarou, aby nebylo možné později připsovat další text. \_\_\_\_\_

Při sazbě této ukázky byla použita následující makra.

```

49 \let\origpar\par
50 \newcommand*{\parrule}{%
51   \hrule height 2.2pt depth -1.8pt\relax}
52 \newcommand*{\caranakonci}{%
53   \unskip\nobreak\space
54   \leaders\parrule\hskip\Flushglue
55   \vadjust{}}{\parfillskip=0pt\origpar}}
```

Pokud nastavíme

```
56 \let\par\caranakonci
```

pak budou všechny další odstavce zakončeny čarou. Je ale třeba být opatrný, protože L<sup>A</sup>T<sub>E</sub>X pracuje s mnoha věcmi jako s odstavci, takže se můžeme dočkat různých překvapení. Pokud se budeme chtít vrátit k normálně zakončeným odstavcům, napíšeme

```
57 \let\par\origpar
```

Můžeme také využít skupiny.

```

58 \begingroup
59 \let\par\caranakonci
60 Odstavec s čarou na konci...
61
62 Další odstavec s čarou...
63
64 \endgroup
65 Normální odstavec...
```

Běžně se odstavec ukončuje prázdným řádkem nebo příkazem `\par`. Pokud budeme chtít zakončit čarou pouze jediný odstavec, použijeme namísto prázdného řádku makro `\caranakonci`.



### 3. Ruské formátování odstavců

Procházel jsem staré články, abych protřídil ty, které už nejsou užitečné. Narazil jsem přitom na staré vydání časopisu Baskerville, které obsahovalo článek o sazbě odstavců v ruském stylu [2]. Zdá se, že podle ruské typografické tradice musí být poslední řádek odstavce buď dlouhý alespoň `\parindent` a být ukončený mezerou alespoň `\parindent`, nebo musí být zarovnaný na levý i pravý okraj.

Tento požadavek není možné splnit jednoduchým nastavením parametrů odstavce.

V uvedeném článku byla popsána dvě řešení. První řešení je od Petera Schmitta. Základní princip je zakončit odstavec posloupností `glue + hbox + glue`, kde prázdný `hbox` má šířku `\parindent`, šířka `glue + hbox` se pohybuje od `\parindent` do `\hsize - \parindent` a šířka `hbox + glue` se pohybuje od `\hsize - \parindent` do `\hsize`, přičemž `\hsize` znamená šířku sazby. Podle pravidel  $\text{\TeX}$  se řádek může zlomit buď před `glue + hbox` nebo ihned za `hbox`. V obou případech takto vznikne další, prázdný řádek odstavce, který se musí odstranit.

```
66 \def\ruskyPS{\ifhmode \unskip
67   \hskip-2\parindent minus -2\parindent
68   \hskip\hsize minus\hsize
69   \hbox{\hskip\parindent}%
70   \hskip0pt \hbox{\strut}%
71   \hskip-\parindent
72   \hskip\hsize plus\parindent
73   \vadjust{\nobreak\vskip-\baselineskip}%
74   \parfillskip0pt
75   \origpar
76   \fi}
```

Jak toto zarovnání vypadá v praxi, je ukázáno v rámečku. Použilo se nastavení

```
77 \parindent=2em
78 \let\par\ruskyPS
```

Druhé řešení pochází od Donalda Arseneaua.

```
79 \def\ruskyDA{\ifhmode \unskip
80   \strut\vadjust{}\nobreak
81   \discretionary{}%
82     {\hbox{\hskip2\parindent
83              \vrule depth 273sp
84              width 0sp height \ht\strutbox}}%
85     {\hbox{\hskip\parindent}}%
86   \hskip-2\parindent minus 2\parindent
```

<p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4 5</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4 5 6</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4 5 6 7</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4 5 6 7 8</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4 5 6 7 8 9</p>
---

```

87 \hskip\hsize minus\hsize
88 \kern0pt\parfillskip0pt
89 \origpar
90 \ifdim\prevdepth=273sp
91   \nobreak
92   \vskip-2\baselineskip
93   \hbox{\strut}%
94 \fi
95 \fi}

```

Toto řešení funguje podobně jako Peterovo řešení, na rozdíl od něj se však nemusí vytvořit přebytný prázdný řádek. Makra vloží do textu neviditelnou čáru konkrétní velmi malé hloubky. Pokud se tato čára dostane na další řádek, můžeme ji testem na `\prevdepth` odstranit.

## Seznam literatury

- [1] Arseneau, Donald. `shapepar.sty`, 2002. Dostupné na CTAN v adresáři `latex/macros/generic/shapepar`.
- [2] Carlisle, David; Schmitt, Peter. *Russian paragraph shapes*. Baskerville, 6(1):13–15, February 1996.
- [3] Eijkhout, Victor. *T<sub>E</sub>X by topic, A T<sub>E</sub>Xnician's Reference*. Addison-Wesley, 1991. Dostupné na [www.wijkhout.net/tbt](http://www.wijkhout.net/tbt).

- [4] Flynn, Peter. *Re: simultaneous justification in latex*. Příspěvek ve skupině `comp.text.tex`, 20. 9. 2006.
- [5] Knuth, Donald E. *The T<sub>E</sub>Xbook*. Addison-Wesley, 1984.
- [6] Platis, Nikos. *Justify at right margin or in next line*. Příspěvek ve skupině `comp.text.tex`, 21. 8. 2006.
- [7] Wilson, Peter. *The hanging package*, April 2004. Dostupné na CTAN v adresáři `latex/macros/contrib/hanging/`.
- [8] Wilson, Peter. Glisterings. *TUGboat*, 28(2):229–232, 2007.

## Summary

This paper demonstrates possibilities of setting the T<sub>E</sub>X paragraph parameters for different, even less usual, paragraph justifications.

**Keywords:** L<sup>A</sup>T<sub>E</sub>X, shapepar package, paragraph, justification.

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