

Zpravodaj Československého sdružení uživatelů TeXu

Peter Wilson

Mělo by to fungovat XII

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 32 (2022), No. 1-4, 77–88

Persistent URL: <http://dml.cz/dmlcz/151109>

Terms of use:

© Československé sdružení uživatelů TeXu, 2022

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 L^AT_EXu vytvořit rámeček kolem textu, přičemž uvnitř rámečku je možné provést stránkový zlom. Jsou popsána makra z balíčku `framed` a také je ukázáno, jak je možné na základě tohoto balíčku vytvořit vlastní rámečky.

Klíčová slova: Rámečky, balíček `framed`, L^AT_EX

Fame is no plant that grows on mortal soil,
Nor in the glistering foil
Set off to the world, nor in broad rumour lies:
But lives and spreads aloft by those pure eyes
And perfect witness of all-judging Jove;
As he pronounces lastly on each deed,
Of so much fame in heaven expect thy meed.

*Lycidas, Elegy on a Friend drowned
in the Irish Channel, 1637*

JOHN MILTON

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

I wuz framed!

Tradiční obhajoba zločince

1. Rámečky

Často na `comp.text.tex` padají dotazy, jak lze kolem textu umístit rámeček nebo jak podbarvit nějaký blok textu. Tyto problémy částečně řeší makro `\fbox`, které vloží svůj obsah do `rámečku`, nebo makro `\colorbox` z balíčku `color` nebo `xcolor`, které svůj obsah `podbarví`. Tato makra však neumožňují uvnitř rámečku provést stránkový zlom.¹ Někdy však potřebujeme zarámovat text, který bude stránkový zlom obsahovat. V tom případě lze použít balíček `framed` [2] Donalda Arseneaua. Tento balíček definuje čtyři prostředí, která si postupně ukážeme.

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

¹Tato makra pracují v horizontálním módu, kde svůj argument nejdříve uzavřou do `\hboxu`. Proto v nich není možné provést řádkový ani stránkový zlom. (pozn. překl.)

Prostředí **framed** z balíčku **framed** vloží svůj obsah do rámečku. Na rozdíl od jiných podobných maker a prostředí, toto prostředí umožňuje uvnitř rámečku provést stránkový zlom.²

Dalším prostředím je **shaded**. Toto prostředí podbarví svůj obsah, přičemž opět v něm umožňuje provést stránkový zlom. Před použitím prostředí je nutné nadefinovat barvu **shadecolor**. V této ukázce jsem použil barvu

```
1 \definecolor{shadecolor}{gray}{0.8}
```

Makro **\definecolor** je definováno v balíčku **color** nebo **xcolor**.

Prostředí **snugshade** je podobné jako prostředí **shaded**. Také podbarví svůj obsah a je možné v něm provést stránkový zlom. V této ukázce jsem použil barvu

```
2 \definecolor{shadecolor}{gray}{0.9}
```

Rozdíl je ten, že v prostředí **snugshade** podbarvení přesahuje okraj textu jen o málo.³

Prostředí **leftbar** vloží nalevo od svého obsahu svislou čáru. Jako všechny tři předchozí prostředí, i toto umožňuje uvnitř svého obsahu provést stránkový zlom.

Balíček **framed** může být užitečný i v případě, kdy víme, že ke stránkovému zlomu nemůže dojít. Týká se to například prostředí **figure**, kde můžeme chtít obrázek zarámovat, abychom jej opticky více odlišili od okolního textu. Obrázek 1 jsme vytvořili pomocí následujícího kódu.

```
3 \begin{figure}
4   \begin{framed}
5     \centering
6     zarámovaný obrázek
7   \end{framed}
8   \caption{Ukázka zarámovaného obrázku}
9   \label{obr1}
10 \end{figure}
```

Podobně můžeme zarámovat obrázek 2 včetně popisku.

```
11 \begin{figure}
12   \begin{framed}
```

²Jak jistě čtenáři pochopili, toto prostředí pracuje ve vertikálním módu. (pozn. překl.)

³Konkrétně lze tento přesah nastavit v registru **\fboxsep** podobně jako na řádce 30. (pozn. překl.)

obrázek

Obrázek 1: Ukázka zarámovaného obrázku

obrázek

Obrázek 2: Ukázka zarámovaného obrázku včetně popisku

```
13 \centering
14 obrázek
15 \caption{Ukázka zarámovaného obrázku včetně popisku}
16 \label{obr2}
17 \end{framed}
18 \end{figure}
```

Jestliže by úplný rámeček kolem obrázku byl příliš výrazný, můžeme použít namísto rámečku pouze vodorovné linky. K tomu nepotřebujeme žádný balíček. Aby byl příklad názornější, použil jsem v obrázku 3 velmi tlusté linky.

```
19 \begin{figure}
20 \centering
21 \rule{\linewidth}{2pt}
22 obrázek
23 \vspace{0.5\baselineskip}
24 \hrule
25 \caption{Ukázka obrázku s~vodorovnými linkami}
26 \label{obr3}
27 \rule{\linewidth}{2pt}
28 \end{figure}
```

Za určitých okolností může být lepší variantou obrázek podbarvit. K tomu použijeme prostředí `snugshade` z balíčku `framed`.

```
29 \begin{figure}
30 \setlength{\fboxsep}{0mm}
31 \begin{snugshade}
32 \vspace{0.5\baselineskip}
33 \centering
34 obrázek
35 \caption{Ukázka podbarveného obrázku včetně popisku}
36 \label{obr4}
```


Obrázek 3: Ukázka obrázku s vodorovnými linkami

Obrázek 4: Ukázka podbarveného obrázku včetně popisku

```

37 \vspace{0.5\baselineskip}
38 \end{snugshade}
39 \end{figure}

```

Ne všechny objekty je možné zapouzdřit do některého prostředí z balíčku **framed**. Konkrétně při použití plovoucích objektů, poznámek pod čarou, poznámek v okraji nebo značek `\mark` uvnitř některého z těchto prostředí se tyto objekty ztratí. Dále balíček **framed** také nespolutracuje s balíčkem **multicol** nebo s jinými makry používanými pro sloupcovou sazbu. V některé budoucí verzi balíčku možná budou tyto problémy vyřešeny.

Pomocí balíčku **framed** si můžeme vytvořit vlastní prostředí, v němž může nastat stránkový zlom. K tomu využijeme prostředí⁴ **MakeFramed** a nadefinujeme makro `\FrameCommand`, které příslušný „rámeček“ vykreslí. Pro prostředí **MakeFramed** má jeden argument, v němž je možné nastavit šířku sazby `\hsize` a definovat, jakým způsobem se za vysázeným rámečkem případná nová nastavení vrátí zpět. Balíček pro tyto účely definuje makro `\FrameRestore`, které je jednodušší variantou vnitřního L^AT_EXového makra `\@parboxrestore`. V délkovém registru `\width` je uložena šířka celého rámečku. Vše si ukážeme později. Ale pojďme od začátku.

Rámeček prostředí **framed** je definován pomocí makra `\fbox` následovně.

```

40 \providecommand{\FrameCommand}{%
41 \setlength{\fboxrule}{\FrameRule}%
42 \setlength{\fboxsep}{\FrameSep}%
43 \fbox}

```

Délkové registry `\FrameRule` a `\FrameSep` jsou zavedeny v balíčku a určují tloušťku čáry a mezeru na okraji rámečku. Samotné prostředí **framed** pak využívá makro `\MakeFramed`.

⁴Namísto použití maker `\begin{MakeFramed}` a `\end{MakeFramed}` můžeme přímo použít makra `\MakeFramed` a `\endMakeFramed`, což ostatně sám autor dále v textu dělá. (pozn. překl.)


```

44 \newenvironment{framed}{%
45     \MakeFramed {\advance\hsize-\width
46         \FrameRestore}}%
47 {\endMakeFramed}

```

Zbývající tři prostředí jsou definována podobně. Jako rámovací makro používají `\colorbox` nebo `\vrule`.

```

48 \newenvironment{shaded}{%
49     \def\FrameCommand{\fboxsep=\FrameSep
50         \colorbox{shadecolor}}%
51     \MakeFramed {\FrameRestore}}%
52 {\endMakeFramed}

```

```

53 \newenvironment{snugshade}{%
54     \def\FrameCommand{%
55         \colorbox{shadecolor}}%
56     \MakeFramed {\FrameRestore\@setminipage}}%
57 {\par\unskip\endMakeFramed}

```

```

58 \newenvironment{leftbar}{%
59     \def\FrameCommand{\vrule width 3pt
60         \hspace{10pt}}%
61     \MakeFramed {\advance\hsize-\width \FrameRestore}}%
62 {\endMakeFramed}

```

Povšimněte si, že uvnitř prostředí `framed` a `leftbar` je sazba zúžená. Naopak u prostředí `shaded` a `snugshade` se šířka sazby nemění.

Tyger, tyger, burning bright
 In the forests of the night,
 What immortal hand or eye
 Could frame thy fearful symmetry?

Songs of Experience
 WILLIAM BLAKE

2. Definice vlastních rámečků

V některých případech je celkem snadné nadefinovat si své vlastní rámovací prostředí založené na balíčku `framed`, avšak většinou to chce trochu experimentování.

Prostředí `narrowframe` zúží sazbu textu na danou šířku, text zarámkuje a rámeček vycentruje.

```
63 \newenvironment{narrowframe}[1][0.8\hsize]{%  
64     \MakeFramed{\setlength{\hsize}{#1}  
65     \FrameRestore}}%  
66     {\endMakeFramed}
```

Prostředí můžeme použít buď přímo

```
67 \begin{narrowframe} ... \end{narrowframe}
```

nebo s jedním nepovinným parametrem udávajícím šířku sazby, například

```
68 \begin{narrowframe}[10cm] ... \end{narrowframe}
```

V této ukázce je použito implicitní nastavení, kdy je šířka sazby uvnitř rámečku rovna 80 % šířky sazby v okolí rámečku.

Můžeme vyzkoušet také hladší rámečky. Prostředí `roundedframe` vytvoří zaoblený rámeček. K tomu potřebuje balíček `fancybox`⁵ a jeho makro `\ovalbox`. V definici prostředí samozřejmě můžeme parametry nastavit dle libosti, případně použít jiná makra z tohoto balíčku k vytvoření jiných rámečků.

```
69 \newenvironment{roundedframe}{%  
70     \def\FrameCommand{%  
71         \cornersize*{20pt}%  
72         \setlength{\fboxsep}{5pt}%  
73         \ovalbox}%  
74     \MakeFramed{\advance\hsize-\width  
75     \FrameRestore}}  
76     {\endMakeFramed}
```

Čas od času se na `comp.text.tex` vyskytne požadavek na vytvoření graficky odděleného číslovaného prostředí. Robert Nyqvist [3] v jedné z odpovědí poskytl následující kód založený na balíčku `framed`.

⁵Jak se ukázalo při sazbě tohoto článku, není balíček `fancybox` kompatibilní s balíčkem `fancyvrb`. Oba balíčky definují prostředí `Verbatim`, každý však jinak. (pozn. red.)


```

77 \makeatletter
78 \definecolor{rulecolor}{gray}{0.65}
79 \newcounter{ruledexample}
80 \newlength{\releftgap}
81 \setlength{\releftgap}{4pt}
82 \newlength{\rerightgap}
83 \setlength{\rerightgap}{1em}
84 \newlength{\rerule}
85 \setlength{\rerule}{1.25pt}
86 \newlength{\Eheight}
87 \newenvironment{ruledexample}[1][Příklad]{%
88     \settoheight{\Eheight}{\textbf{#1}}%
89     \addtolength{\Eheight}{-\rerule}%
90     \def\FrameCommand{\hspace{-\releftgap}%
91         {\color{rulecolor}%
92             \vrule width \rerule}%
93         \hspace{\releftgap}\hspace{-\rerule}}}%
94     \MakeFramed{\advance\hsize-\width}%
95     \refstepcounter{ruledexample}%
96     \makebox[Opt][l]{%
97         \hspace{-\parindent}%
98         \hspace{\rerightgap}%
99         {\color{rulecolor}\rule{1.5em}{\rerule}}}%
100     \quad
101     \raisebox{-0.5\Eheight}[Opt]{%
102         \textbf{#1\ \theruledexample}}%
103     }\\[.5\baselineskip]%
104     \noindent\ignorespaces}%
105 {\@afterheading\
106     \makebox[Opt][l]{%
107         \hspace{-\releftgap}%
108         {\color{rulecolor}
109             \rule{\columnwidth}{\rerule}%
110             \rule{\releftgap}{\rerule}}}%
111     }}
112 \endMakeFramed}
113 \makeatother

```

— Příklad 1

Tento příklad ukazuje použití prostředí `ruledexample`. Prostředí má svůj titulék, číslo a můžeme se na něj odkazovat přes `\label`. Pokud je to třeba, provede se uvnitř prostředí stránkový zlom. Zdrojový kód je založený na makrech Roberta

Nyqvista z roku 2003, přičemž jsem k nim přidal následující parametry, jimiž je možno upravit vzhled prostředí.

- `rulecolor` – barva linek
- `\rerule` – tloušťka linek
- `\releftgap` – vzdálenost, o jakou je svislá čára posunuta do okraje
- `\rerightgap` – odsazení linky v titulku
- nepovinný parametr – titulek prostředí, implicitně je „Příklad“

Jak můžete vidět na tomto příkladu, je možné uvnitř prostředí `ruledexample` používat výčtová prostředí jako `itemize`.

Někteří uživatelé si všimli, že se prostředí `framed` uvnitř jiného prostředí, jako například uvnitř `quote` nebo uvnitř `adjustwidth` z balíčku `changepage` [4], chová jinak, než by očekávali. Níže situaci popíšu blíže.

Zahájíme prostředí `quotation`. V něm můžeme použít prostředí `framed`, stejně jako libovolné další prostředí založené na prostředí `list`.

Nyní jsme uvnitř prostředí `framed`. Všimněte si, že rámeček je napříč celou původní šířkou textu. To není vždy žádoucí.

S Donaldem Arseneauem jsme diskutovali, jak problém vyřešit a udělat rámeček užší. Oba jsme vytvořili nějaký zdrojový kód a jako obvykle ten Donaldův byl lepší.

Nyní jsme uvnitř prostředí `qframe` vytvořeném Donaldem Arseneauem. Definici tohoto prostředí jsem poté vložil do mé třídy `memoir` [5]. Do ní jsem vložil také stínovanou variantu tohoto prostředí.

Jestliže použijete prostředí `adjustwidth` z balíčku `changepage`⁶ [4] nebo z třídy `memoir`, pak uvnitř něho prostředí `qframe` funguje lépe než prostředí `framed`.

Zde prostředí `quotation` ukončíme.

Definice prostředí `qframe` je následující.

```
114 \makeatletter
115 \newenvironment{qframe}{%
116   \def\FrameCommand##1{%
117     \setlength{\fboxrule}{\FrameRule}%
118     \setlength{\fboxsep}{\FrameSep}%
119     \hskip\@totalleftmargin\fbox{##1}%
```

⁶Balíček `changepage` je nástupce balíčku `chnpage`, který se již nepoužívá.


```

120      \hskip-\linewidth
121      \hskip-\@totalleftmargin
122      \hskip\columnwidth}%
123      \MakeFramed{\advance\hsize-\width
124      \advance\hsize \FrameSep
125      \@totalleftmargin\z@
126      \linewidth=\hsize}}%
127      {\endMakeFramed}
128 \makeatother

```

Dalším užitečným druhem rámečku je rámeček s popiskem, který se na následující stránce zopakuje. Kód prostředí je výrazně komplikovanější než dříve a musel jsem celkem hodně experimentovat, aby vše fungovalo. Nejdříve si ukážeme příklad.

Rámeček s popiskem

Prostředí `framewithtitle` vytvoří rámeček a na jeho začátek vloží popisek. Jestliže dojde ke stránkovému zlomu, bude za ním následovat rámeček s pokračovacím popiskem. Definice prostředí následuje níže.

Prostředí můžeme použít se dvěma parametry udávajícími pokračovací popisek a úvodní popisek

```

129 \begin{framewithtitle}[pokračovací]{úvodní}
130 ...
131 \end{framewithtitle}

```

První parametr je nepovinný a když jej vynecháme, bude pokračovací popisek stejný jako úvodní, ale následovaný textem „(pokr.)“

```

132 \begin{framewithtitle}{popisek}
133 ...
134 \end{framewithtitle}

```

Umístění popisku je definováno v makru `\frametitle`, které podle definice na řádku 138 vysází popisek doleva. Pokud bychom chtěli popisek vycentrovat, předefinujeme makro následovně.

```

135 \renewcommand*{\frametitle}[1]{%
136   \centerline{\strut#1}}

```

I uvnitř tohoto rámečku je možné sázet zdrojový kód.

Zdrojový kód prostředí `framewithtitle` je založený na námětech Donalda Arseneaua, které měl ke starší verzi balíčku `framed`. Velká část kódu se zabývá popiskem, aby na začátku rámečku byl úvodní popisek a po stránkovém zlomu následoval pokračovací popisek. Některé části kódu byly uloženy do maker, aby

bylo možné je použít i v dalších situacích. Nejsložitějším problémem je, že makra neustále přenastavují popisek, aby určila místo případného stránkového zlomu, a až pak příslušný popisek vysázejí. Toto je ošetřeno testem `\ifcontframe`, na jehož základě se nastaví konkrétní popisek. Celé toto testování se provádí uvnitř makra `\Fr@meSetup`. Makro `\FrameTitle` pak popisek vysází.

```

137 \makeatletter
138 \newcommand*{\frametitle}[1]{\strut#1}
139 \newif\ifcontframe
140 \newcommand*{\Fr@meSetup}[2]{%
141   \fboxrule=\FrameRule \fboxsep=\FrameSep
142   \global\contframefalse
143   \def\Fr@meFirst{\textbf{#2}}}%
144   \def\Fr@meCont{\textbf{#1}}}%
145   \def\FrameCommand##1{%
146     \Title@Fr@me{\Fr@meCurrent}{##1}%
147     \global\let\Fr@meCurrent\Fr@meNext
148     \ifcontframe
149       \global\let\Fr@meNext\Fr@meCont
150     \fi
151     \global\contframefalse}%
152   \global\let\Fr@meCurrent\Fr@meFirst
153   \global\let\Fr@meNext\Fr@meFirst}
154 \newcommand*{\FrameTitle}[1]{%
155   \nobreak \vskip -0.7\FrameSep
156   \rlap{\frametitle{#1}}%
157   \nobreak\nointerlineskip
158   \vskip 0.7\FrameSep}
159 \newenvironment{framewithtitle}[2][\Fr@meFirst\ (pokr.)]{%
160   \def\Title@Fr@me##1##2{%
161     \fbox{\vbox{\FrameTitle{##1}%
162       \hbox{##2}}}}}%
163   \Fr@meSetup{#1}{#2}%
164   \MakeFramed{%
165     \advance\hsize-\width
166     \FrameRestore}}%
167   {\global\contframefalse
168   \endMakeFramed}
169 \makeatother

```

Jako alternativu můžeme použít prostředí `titledframe`, v němž je popisek vysázen nad rámečkem. Parametry tohoto prostředí a horizontální pozice popisku se nastavují stejně jako u prostředí `framewithtitle`.

Popisek nad rámečkem

Prostředí `titledframe` vytvoří rámeček a nad něj vloží popisek. Jestliže dojde ke stránkovému zlomu, bude za ním následovat rámeček s pokračovacím popiskem. Definice prostředí následuje níže.

```
170 \makeatletter
171 \newenvironment{titledframe}[2][\FrameFirst\ (pokr.)]{%
172     \def\Title@Frame##1##2{%
173         \vbox{\FrameTitle{##1}%
174             \noindent\fbbox{##2}}}%
175     \FrameSetup{#1}{#2}%
176     \MakeFramed{%
177         \advance\hsize-\width
178         \advance\hsize -2\FrameRule
179         \advance\hsize -2\FrameSep
180         \FrameRestore}}%
181     {\global\contframefalse
182     \endMakeFramed}
183 \makeatother
```

Musím uznat, že jsem detaily zdrojových kódů příliš nepopisoval. Jedním z důvodů je, že by se tímto článek velmi prodloužil. Ale hlavním důvodem je, že příliš nerozumím, jak balíček `framed` provádí všechna ta svá kouzla. Proto jsem také hodně experimentoval (mnoho pokusů a ještě více omylů), než všechno začalo pořádně fungovat.

Odkazy

1. WILSON, Peter. Glisterings. *TUGboat* [online]. 2011, roč. 32, č. 1, s. 99–103 [vid. 2011-05]. Dostupné z: <https://tug.org/TUGboat/tb32-1/tb100glister.pdf>.
2. ARSENEAU, Donald. *framed: Framed or shaded regions that can break across pages* [online]. CTAN, 2007 [vid. 2011-05]. Dostupné z: <https://ctan.org/pkg/framed>. Verze 0.95.
3. NYQVIST, Robert. ‘example’ environment or command. *comp.text.tex* news-group, 2003-01-11.
4. WILSON, Peter. *changepage: Margin adjustment and detection of odd/even pages* [online]. CTAN, 2009-10 [vid. 2011-05]. Dostupné z: <https://ctan.org/pkg/changepage>.
5. WILSON, Peter. *memoir: Typeset fiction, non-fiction and mathematical books* [online]. CTAN, 2008 [vid. 2011-05]. Dostupné z: <https://ctan.org/pkg/memoir>.

Summary: It Might Work XII

This paper shows several ways in L^AT_EX how to create a frame around a text with a possible page break inside the frame. Some solutions using the `framed` package are given. Several possibilities to create new types of frames based on that package are also presented.

Keywords: Frames, framed package, L^AT_EX

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