

Zpravodaj Československého sdružení uživatelů TeXu

Jan Šustek

Jak umožnit stránkový zlom uvnitř vložených obrázků

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 33 (2023), No. 3-4, 102–110

Persistent URL: <http://dml.cz/dmlcz/151993>

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>

Jak umožnit stránkový zlom uvnitř vložených obrázků

JAN ŠUSTEK

V článku jsou definována a popsána makra \TeX u pro vložení objektů, které jsou natolik vysoké, že komplikují stránkový zlom. Může se jednat o obrázky, vysázený text nebo obecně o obsah boxu. Makra vloží objekt na aktuální místo sazby a umožní uvnitř objektu provést stránkový zlom.

Klíčová slova: rámečky, obrázky, řezání, stránkový zlom

1 Úvod

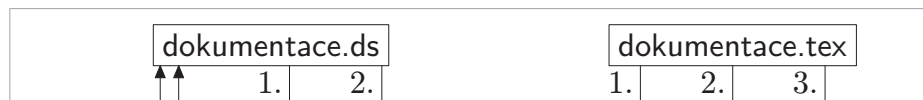
Když autor psal předchozí článek tohoto Zpravodaje začínající na straně 66, narazil na problém, že článek obsahuje velké množství ukázek, které bylo nutné opticky oddělit od okolního textu. Autor se rozhodl ukázky vložit do šedých rámečků. Rámečky však musejí být vysoké, jinak by ukázky nemohly ukázat vše potřebné. To samozřejmě velmi komplikuje stránkový zlom. Přitom využití plovoucích objektů by zhoršilo plynulost textu a odkazy na plovoucí objekty by zhoršily srozumitelnost vět.

Proto se autor rozhodl umisťovat rámečky přímo k příslušnému textu a v rámečcích umožnit stránkový zlom. Ten ale nelze provést v libovolném místě rámečku. Mohlo by se totiž stát, že se zlom provede uprostřed řádku. Ukázky přitom pocházejí z vnějších souborů a do těch autor článku často nemohl snadno zasáhnout.

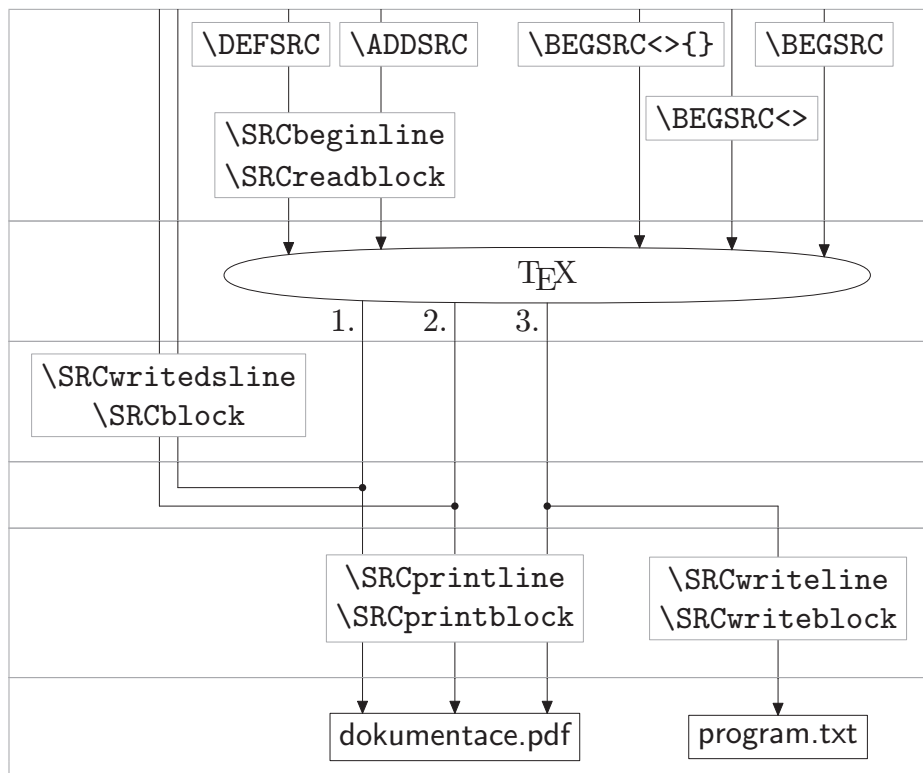
Níže definovaná makra umožní do rámečku vložit soubor s obrázkem¹ nebo obsah \TeX ového boxu a nastavit seznam míst, v nichž je možné provést stránkový zlom. Interně se provede to, že pomocí operátorů jazyka pdf se vložený obrázek ořeže v daných místech na nevysoké části, tyto části se poskládají pod sebe a mezi nimi se umožní stránkový zlom.

2 Použití

Použití maker ukážeme na grafu ze strany 80 tohoto Zpravodaje:



¹Podporovány jsou všechny formáty obrázků, které lze vložit primitivem `\pdfximage`, tj. jbig2, jpeg, pdf a png.



Graf jsme vložili pomocí následujících příkazů:

```

\input rezani
\ladenitrue
...
\ram{095,35,495,64,72,9}{graf-1.pdf}

```

Parametry určují, že ve vkládaném souboru `graf-1.pdf` umožníme stránkový zlom na souřadnicích 9,5 %, 35 %, 49,5 %, 64 %, 72 % a 90 % (měřeno od horního okraje obrázku). Při nastaveném `\ladenitrue` se navíc zobrazí místa možných řezů, abychom snadněji mohli určit jejich souřadnice.

Uvedená makra je také možné použít pro stránkový zlom uvnitř `\vbox`, v němž může být vysázen například verbatim výpis. Použijeme ukázkou ze strany 78 tohoto Zpravodaje. Ukázka je v `OPmacu`, ale s ohledem na šablonu `Zpravodaje` použijeme pro sazbu ukázkou `LATEX`. Přitom je použito prostředí `Verbatim` z balíčku `fancyvrb`, které umožňuje uvnitř verbatim výpisu volat makra, například k barevnému zvýraznění textu.

```

\def\zelena#1{\textcolor[cmym]{1,0,1,0}{#1}}
\def\fialova#1{\textcolor[cmym]{0,1,0,0}{#1}}
\nastavovrbox
\vskip-\medskipamount
\vskip0pt
\begin{Verbatim}[commandchars=/(, numbers=none]
\input gensrc
\def\SRChook{\longlocalcolor\Red}
\SRCFILENAME program.txt
Here we define a~block which will be inserted to another
  place.
/zelena(\BEGSRC<InternalLabel>{Displayed label})
second line
  /fialova(|<InnerBlock>)
/zelena(\ENDSRC)
...
\end{Verbatim}
\vskip-\medskipamount
\endnastavovrbox
\dily23 \ram{3,5,9,13,18}{-}

```

Makra `\nastavovrbox` a `\endnastavovrbox` jsou definována v sekci 3 a pomocí nich se vytvoří `\vbox`, do nějž je pak vložen obsah prostředí `Verbatim`. Tento box pak rozřežeme na 23 stejných dílů a umožníme stránkový zlom po 3, 5, 9, 13 a 18 dílech. Přitom musíme odstranit mezeru `\medskipamount`, kterou prostředí `Verbatim` vkládá nad a pod svůj obsah.

<pre> \input gensrc \def\SRChook{\longlocalcolor\Red} \SRCFILENAME program.txt </pre>
<pre> Here we define a~block which will be inserted to another place. </pre>
<pre> \BEGSRC<InternalLabel>{Displayed label} second line <InnerBlock> \ENDSRC </pre>
<pre> A~block can be defined by parts. \BEGSRC<InternalLabel> fourth line \ENDSRC </pre>

```
And here we insert the block.
```

```
\BEGSRC
```

```
first line
```

```
  |<InternalLabel>
```

```
\ENDSRC
```

```
Finally we define the inner block.
```

```
\BEGSRC<InnerBlock>{Inner block}
```

```
third line
```

```
\ENDSRC
```

```
\bye
```

3 Syntaxe

Je možné použít některou z následujících syntaxí:

- `\ram{obrázek}` vloží obrázek do rámečku, který se nezlomí.
- `\ram{1,23,4567,89}{obrázek}` vloží obrázek do rámečku a umožní stránkový zlom na souřadnicích 10 %, 23 %, 45,67 % a 89 % (měřeno od horního okraje obrázku).
- `\dily11 \ram{3,5,7,8}{obrázek}` podobně umožní stránkový zlom na souřadnicích $\frac{3}{11}$, $\frac{5}{11}$, $\frac{7}{11}$ a $\frac{8}{11}$. Tato syntaxe je výhodná, když víme, že v ukázce je zdrojový kód, který má ekvidistantní řádky, v tomto případě 11.
- `\dily11 \ud \ram{obrázek}` umožní stránkový zlom na stejných souřadnicích jako `\dily11 \ram{3,4,5,6,7,8}{obrázek}`. Jinými slovy místa řezů nastaví rovnoměrně,² přičemž vynechá řezy u horního a dolního okraje obrázku.
- `\Ram{obrázek}{strana}` u vícestránkových obrázků vloží do rámečku danou stranu obrázku.
- Podobně se pro makro `\Ram` použijí i zbývající výše uvedené syntaxe.
- `\setbox\obrbox\vbox{...} \ram{1,23,4567,89}{}` nevloží do rámečku obrázek, ale obsah boxu `\obrbox`.³ V tomto případě se samozřejmě název souboru nezadává, i tak ale složené závorky ohraničující druhý argument makra `\ram` musejí zůstat.

Pokud bychom chtěli mít vnitřek rámečku s `\obrboxem` stejně široký jako u rámečků s obrázkem, musíme si zajistit, že uvnitř `\vboxu` bude správně nastavena šířka sazby, například pomocí následujících maker:

```
\def\nastavobrbox{\setbox\obrbox\vbox\bggroup
  \hsize\dimexpr\hsize-2\hodstup\relax}
```

²Název makra `\ud` vznikl ze zkratky „u.d.“ = „uniformly distributed“.

³Aby mohl uživatel do rámečku vložit také verbatim výpisy nebo jiný delší text, je praktičtější i přehlednější si tento obsah rámečku předem připravit do boxu a ten pak použít.

```
\def\endnastavobrbox{\egroup}
```

a použitím následující konstrukce:

```
\nastavobrbox
\begin{verbatim}
...
\end{verbatim}
\endnastavobrbox
\dily11 \ud \ram{}
```

- Také v případě `\setbox\obrbox` je možné použít všechny výše uvedené syntaxe.

Pokud bychom kolem obrázku nechtěli kreslit rámeček, stačí předem provést následující nastavení:

```
\hodstup=0pt
\vodstup=0pt
\tloustkaramu=0pt
```

Makra jsou definována a odladěna v OPmacu. Přitom byl soubor s makry beze změny použit v předchozím článku tohoto Zpravodaje, který je vysázen v \LaTeX u. Soubor se jmenuje `rezani.tex` a je ke stažení online [1].

4 Implementace

Nejdříve nadefinujeme některá OPmacová makra pro případ, že používáme \LaTeX .

```
1 \unless\ifdefined\OPmac
2 \csname newcount\endcsname\tmpnum
3 \def\sdef#1{\expandafter\def\csname#1\endcsname}
4 \def\sxdef#1{\expandafter\xdef\csname#1\endcsname}
5 {\lccode'?'=\p \lccode'\!='\t
6 \lowercase{\gdef\ignorept#1!{#1}}}
7 \def\localcolor{}
8 \def\Grey{\color[gray]{0.5}}
9 \fi
```

Definujeme základní parametry, které budou postupně vysvětleny dále v textu.

```
10 \newdimen\hodstup \hodstup=3mm
11 \newdimen\vodstup \vodstup=2mm
12 \newdimen\tloustkaramu \tloustkaramu=0.4pt
13 \newskip\okoliramu \okoliramu\medskipamount
14 \def\barvaramu{\localcolor\Grey}
```

Přepínač `\ifladieni` určí, zda se v rámečcích mají zobrazovat místa řezů.

```
15 \newif\ifladieni
```

Makro `\svisleokraje` vycentruje svůj argument do `\hboxu` šířky `\hsize` a kolem něj vysází svislé linky.

```
16 \def\svisleokraje#1{\hbox to\hsize{\svisla\hss#1\hss\svisla}%
17 \nointerlineskip}
18 \def\svisla{{\barvaramu\vrule width\tloustkaramu}}
19 \def\vodorovna{{\barvaramu\hrule height\tloustkaramu}}
20 \def\svislekousky{\svisleokraje{\vbox to\vodstup{\vss}}}
```

Makro `\eitd` provede aritmetický výpočet, jehož výsledkem je délková hodnota v jednotkách `pt`, a na úrovni `expand` procesoru tuto hodnotu bez jednotky `pt` vrátí.

```
21 \def\eitd#1{\expandafter\ignorept\the\dimexpr#1\relax}
```

Makro `\nactirezy` načte seznam souřadnic oddělených čárkou a jednotlivá čísla uloží do maker `\csname rez:n\endcsname`. Formát souřadnic je následující:

- Jestliže je `\dily = 0`, pak jsou hodnoty dány jako zlomkové části čísel z intervalu $(0, 1)$, a to bez uvedení desetinné tečky.
- Jestliže je `\dily > 0`, pak jsou hodnoty dány jako čitatele zlomků, jejichž jmenovatel je roven `\dily`.

```
22 \def\nactirezy#1,#2\konec{%
23 \advance\pocetrezu1
24 \ifnum\dily>0
25 \sxdef{rez:\the\pocetrezu}{\eitd{#1pt/\dily}}%
26 \else
27 \sxdef{rez:\the\pocetrezu}{0.#1}%
28 \fi
29 \ifx:#2:%
30 \sdef{rez:\the\numexpr\pocetrezu+1}{1}%
31 \let\next\relax
32 \else
33 \def\next{\nactirezy#2\konec}%
34 \fi\next}
```

Makro `\ram{seznam}{obrázek}` vysází rámeček s daným obrázkem. Rámeček je na celou šířku `\hsize`. Horizontální a vertikální odstup rámečku od obrázku jsou dány registry `\hodstup` a `\vodstup`. V seznamu jsou čárkou oddělené hodnoty souřadnic, v nichž je možné provést stránkový zlom.

```
35 \def\ram#1#2{\ramA{#1}{#2}}
```

Makro `\Ram{seznam}{obrázek}{strana}` funguje zcela stejně, přičemž se do rámečku vloží daná strana obrázku.

```
36 \def\Ram#1#2#3{\ramA{#1}{page#3}{#2}}
```

Makro `\ramA{seznam}{parametr}{obrázek}` je společným jádrem maker `\ram` a `\Ram`. V registrech `\vyskaobrazkuBP` a `\sirkaobrazkuBP` je uvedena hodnota příslušného rozměru obrázku v jednotkách `bp` pronásobená jednotkou `pt`. V makru se obrázek načte do registru `\obrbox`, pokud je tento registr prázdný. Pokud je neprázdný, pouze se zajistí, že `\obrbox` je `\hbox`. Pak se v cyklu spočítají hodnoty potřebné uvnitř příkazů jazyka `pdf`, které obrázek ořežou a posunou na správné místo. V jednotlivých místech řezů je umožněn stránkový zlom s penaltou 100.

```
37 \newcount\pocetrezu
38 \newbox\obrbox
39 \newdimen\vyskaobrazkuBP
40 \newdimen\sirkaobrazkuBP
41 \newcount\dily
42 \def\ramA#1#2#3{\relax
43   \pocetrezu=0
44   \ifx@#1@%
45     \nactirezy0,\konec
46   \else
47     \nactirezy0,#1,\konec
48   \fi
49   %\vypisrezy
50   \dily=0
51   \ifvoid\obrbox
52     \pdfximage width \dimexpr\hsize-2\hodstup\relax #2{#3}%
53     \setbox\obrbox\hbox{\pdfrefximage\pdflastximage}%
54   \else
55     \ifvbox\obrbox
56       \setbox\obrbox\hbox{\box\obrbox}%
57     \fi
58   \fi
59   \vyskaobrazkuBP=0.996264\ht\obrbox
60   \sirkaobrazkuBP=0.996264\wd\obrbox
61   \vskip\okoliramu
62   \vodorovna
63   \svislekousky
64   \tmpnum=0
65   \ht\obrbox=0pt \dp\obrbox=0pt \wd\obrbox=0pt
```



```

66 \loop
67   \ifnum\tmpnum<\pocetrezu \advance\tmpnum1
68   \edef\ramXB{\eidd{\sirkaobrazkuBP} }%
69   \edef\ramYA{\eidd{\vyskaobrazkuBP-
70     \csname rez:\the\numexpr\tmpnum+1\endcsname
71     \vyskaobrazkuBP} }%
72   \edef\ramYB{\eidd{\vyskaobrazkuBP-
73     \csname rez:\the\tmpnum\endcsname
74     \vyskaobrazkuBP} }%
75   \svisleokraje{\hbox{%
76     \vrule height \dimexpr\ramYB bp-\ramYA bp\relax width0pt%
77     \pdfliteral{q 1 0 0 1 0 -\ramYA cm q
78       0 \ramYA m \ramXB \ramYA l \ramXB \ramYB l 0 \ramYB l
79       h W n}%
80     \copy\obrbox
81     \pdfliteral{Q Q}%
82     \hskip\ramXB bp}}
83   \ifnum\tmpnum<\pocetrezu
84     \ifladi \nobreak \vskip-\tloustkaramu \vodorovna \fi
85     \penalty100
86     \ifladi \vskip-\tloustkaramu \vodorovna \fi
87   \fi
88 \repeat
89 \svislekousky
90 \vodorovna
91 \vskip\okoliramu
92 {\setbox0\hbox{\box\obrbox}}

```

Makro `\ud#1` se expanduje na `#1{seznam}`, kde v závorce je seznam přirozených čísel od $(\udpreskoc + 1)$ do $(\dily - \udpreskoc - 1)$.

```

93 \newcount\udpreskoc \udpreskoc2
94 \def\udbezcarkey#1,\udbc{#1}
95 \def\ud#1{\def\poud{#1}%
96   \def\seznamud{}}%
97   \tmpnum\udpreskoc
98 \loop
99   \ifnum\tmpnum<\numexpr\dily-\udpreskoc-1\relax
100   \advance\tmpnum1
101   \edef\seznamud{\seznamud\the\tmpnum,}%
102 \repeat
103 \expandafter\expandafter\expandafter\poud
104 \expandafter\expandafter\expandafter{

```

```
105 \expandafter\udbezcariky\seznamud\udbc}}
```

Makro `\vypisrezy` vypíše seznam jednotlivých řezů. Toto makro je použito pouze pro účely ladění.

```
106 \def\vypisrezy{ {\tmpnum=0
107 \loop
108 \ifnum\tmpnum<\pocetrezu \advance\tmpnum1
109 \csname rez:\the\tmpnum\endcsname,
110 \repeat
111 \csname rez:\the\numexpr\pocetrezu+1\endcsname}}
```

Odkazy

1. ŠUSTEK, Jan. rezani.tex [online]. 2023-11-07. [cit. 2023-11-13]. Dostupné z: <https://github.com/jsustek/rezani/blob/main/rezani.tex>.

Summary: How to Enable Page Breaks in Embedded Images

The paper defines and describes $\text{T}_{\text{E}}\text{X}$ macros for inserting objects which are so high that the page breaking is difficult. These objects can be images, text examples or generally a content of a box. The macros insert the object on the current position and they allow page break in the middle of the object.

Keywords: frames, images, cutting, page break

Jan Šustek, jan.sustek@seznam.cz