

# Zpravodaj Československého sdružení uživatelů TeXu

---

Vít Zýka

Zvýraznění bloků textu s možností stránkového zlomu

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 9 (1999), No. 3, 146–153

Persistent URL: <http://dml.cz/dmlcz/149846>

## Terms of use:

© Československé sdružení uživatelů TeXu, 1999

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>

Tímto článkem se snažím oslovit dva druhy čtenářů. V první části popisují – pro milovníky cizích makrobaliček – použití maker `backgr`. Ta zvýrazňují bloky textu (např. rámečkem, podbarvením, svislou linkou), které mohou podléhat stránkovému zlomu. Makra pracují ve formátu `plainTeX` i `LATeX`. Ve druhé části textu vysvětluji – pro hlubinné `TeX`ové hloubaly – způsob řešení tohoto problému. Popisují omezení této implementace a ukazují obtížnost realizace této třídy úloh prostřednictvím `TeX`u.

## 1. Zvýraznění bloku textu

*Blokem textu* rozumíme v tomto článku část sazby, u které můžeme vertikálně označit její začátek a konec. Tento blok může být libovolně dlouhý, např. jeden box, jeden odstavec, nebo několik stránek. Kromě hladkého textu může obsahovat nadpisy, matematické rovnice, poznámky pod čarou, obrázky, tabulky a jiný sazební materiál.

Tímto problémem jsem se začal zabývat v souvislosti se sazbou knihy [2], která obsahuje kapitoly, ve kterých autoři diskutují s domnělým studentem. Bylo žádoucí, aby odstavce, ve kterých promlouvá student, byly zřetelně odlišeny od ostatního textu. Nejprve mě napadlo změnit řez písma na písmo skloněné. Autorům se však zdálo toto rozlišení příliš málo nápadné. Uvažovali jsme tedy o následujících způsobech zvýraznění:

- výraznější znaková sada stejné písmové rodiny,
- změna šířky tiskového zrcadla,
- symbolem (iniciálou) na začátku bloku,
- pruhem vedle textu (např. vlnitou čarou),
- podbarvením textu (např. světlou šedí).

Poslední bod se zdál být natolik zajímavý, že jsem se začal ohlížet po jeho řešení, aby mohl být posouzen jeho funkční i estetický účinek<sup>1</sup>. Nalezené techniky se však omezovaly jen na podbarvení nezlomitelného boxu. Přistoupil jsem tedy

---

<sup>1</sup>Ukázalo se, že je třeba rozlišovat zvýraznění a odlišení textu. Např. šedý podklad upoutá na první pohled a čtenář má dojem, že jde o nejdůležitější pasáže knihy. V našem případě bylo třeba text odlišit, ne zvýraznit. Šedý podklad tedy nakonec nebyl použit.

k napsání vlastních maker nazvaných `backgr`, která rozdělí zvýrazněný text na části podle stránkového zlomu. Tyto části se pak zvýrazňují samostatně.

## 2. Makra `backgr` z pohledu uživatele

Popis se vztahuje k balíčku maker `backgr` verze 1.0 z května 1999. Je k dispozici na adrese <http://cmp.felk.cvut.cz/~zyka/zykatex.html>.

### 2.1. Použití

#### Plain $\TeX$

Načteme makra `\input backgr.tex` a označíme začátek zvoleného bloku textu pomocí značky `\backgr` a konec značkou `\endbackgr`.

#### $\LaTeX$ 2 $\epsilon$ i $\LaTeX$ 2.09

Způsob značení z plainu funguje i v  $\LaTeX$ u, kromě něj je definováno i prostředí `backgr`. Lze tedy použít obvyklou syntaxi  $\LaTeX$ u uzavřením bloku do dvojice `\begin{backgr}` a `\end{backgr}`.

### 2.2. Specifikace umístění značek začátku a konce bloku

Značky začátku nebo konce bloku můžeme umístit pouze do hlavního vertikálního módu [1, str. 85]. Značka vložená do odstavcového horizontálního módu nejdříve automaticky uzavře odstavec příkazem `\par` a přejde tak do žádoucího hlavního vertikálního módu. V ostatních módech nelze značek použít. Zvýrazněné bloky nelze do sebe vnořovat.

Logická proměnná `\ifbackgrstrut` ovlivňuje vertikální zarovnání zvýraznění. Pravdivá (přednastavená) hodnota `\backgrstruttrue` způsobí sazbu zvýraznění o hloubku „strut“ výplňku (hloubku kulaté závorky) níže než poslední účaří. Při `\backgrstrutfalse` se poloha zvýraznění kryje s hloubkou předcházejícího boxu.

### 2.3. Ruční zpřesnění polohy

Z níže uvedených důvodů není vertikální usazení zvýraznění vždy správné. Proto je třeba při konečné úpravě dokumentu, kdy již nebude měněn stránkový zlom, ručně posunout některé začátky a konce zvýraznění. Provádí se to pomocí registru typu `\dimen` `\backgrcorrection` před vlastní značkou bloku. Např. nastavení

```
\backgrcorrection=1mm      % LaTeXový zjednodušený zápis:
\backgr                    % \Backgr[1mm]
Blok textu učený ke zvýraznění.
```

```

\backgrcorrection=-.5ex
\endbackgr % \endBackgr[-.5ex]

```

posune začátek zvýraznění o jeden milimetr níže, jeho konec o půl střední výšky písma výše, než je vypočtená hodnota. Po provedení značky `\backgr` nebo `\endbackgr` (i jejich L<sup>A</sup>T<sub>E</sub>Xových ekvivalentů) je hodnota `\backgrcorrection` vrácena na velikost v registru `\backgrconstcorr`, jehož počáteční rozměr je  $0\text{ pt}$ . Toto nastavení způsobí, že další značka nebude posunuta vůči vypočtené hodnotě, nenastavíme-li znovu parametr `\backgrcorrection` na nenulový rozměr. Registrem `\backgrconstcorr` lze korigovat více značek o stejně velké vzdálenosti najednou.

Dojde-li uvnitř zvýrazněného bloku textu k přechodu na novou stránku, vytvoří se v místě zlomu dvě pomocné značky automaticky. Jedna na spodní hraně strany a druhá na začátku strany následující. Chceme-li korigovat polohu těchto pomocných značek, nastavíme registry `\backgrbotcorrection` a `\backgrtopcorrection` na žádoucí rozměry.

## 2.4. Definice způsobu zvýraznění

Balíček `backgr` obsahuje definice následujících způsobů zvýraznění bloku textu nazývaných *styly zvýraznění*:

- zigzag – klikatá linka vedle textu,
- frame – rámeček okolo textu (přednastaveno),
- gray – šedé pozadí (vyžaduje PostScriptový ovladač nebo pdf<sub>T</sub>E<sub>X</sub>),
- grayframe – šedé pozadí s rámečkem (PostScriptový ovladač nebo pdf<sub>T</sub>E<sub>X</sub>),
- empty – žádné zvýraznění (pro dočasné vypnutí).

Lze je měnit makrem, jehož jméno vznikne spojením řetězce `\bgstyle` a názvu stylu. Např. světle šedého podbarvení textu dosáhneme makrem `\bgstylegray`. Změna stylu musí být provedena před značkou začátku zvýraznění.

Chceme-li si vytvořit vlastní styl, nadefinujeme makro s šesti parametry. Ty mají následující význam:

1. Logická proměnná, která udává, zda blok textu pokračuje z předcházející strany. Hodnotu zjistíme testem na shodnost s primitivem `\relax`.
2. Šířka zvýraznění. Je rovna hodnotě makra `\bgwd` při použití značky začátku bloku. Přednastavená hodnota je rovna šířce tiskového zrcadla.
3. Horizontální odsazení vůči levému okraji textu. Rovno okamžité hodnotě makra `\bgind`, přednastaveno  $0\text{ pt}$ .
4. Vertikální vzdálenost horního okraje zvýrazňujícího boxu od horního okraje stránky.
5. Výška zvýraznění.
6. Logická proměnná udávající, zda blok textu pokračuje i na další stranu.

Důležité je, aby makro neposunulo referenční bod sazby, tj. aby zvýrazňující materiál byl sázen ve vertikálním boxu nulového rozměru. Tento box pak výstupní

rutina umístí na začátek sázené stránky. Příklady definic stylů najde čtenář ve zdrojovém souboru `backgr.tex`. Přepnutí na náš nový styl zařídíme následující definicí `\def\backgrstyle{\noexpand\makro realizující náš styl}`.

Pokud chceme ovlivnit vlastní zvýrazněný text, můžeme předefinovat makra `\bgbuser` a `\bgeuser`. Ty se expandují v místě značky `\backgr`, respektive `\endbackgr`. Například pro styl `frame` je vhodné nastavit

```
\def\bgbuser{\bgroup\leftskip=4pt \rightskip=4pt
\advance\parindent by-4pt }
\def\bgeuser{\egroup}
```

Rámeček pak může být sázen na šířku tiskového zrcadla, aniž by byl text nalepen na rámeček.

## 2.5. Kdy makra selžou

Automatické umístění zvýraznění pracuje zcela přesně v hladkém textu. V následujících případech může dojít k odchylce, kterou je možné ručně kompenzovat. Způsob nápravy je uveden itálikou.

- Je-li v hlavním vertikálním seznamu `\mark` nebo `\write`. *Použij `\backgrcorrection`, viz. 2.3.*
- Je-li vložen plovoucí insert. Některé typy insertů jsou poměrně složité a neelegantně ošetřeny, ale např. v současné verzi dojde k hrubé chybě, odsune-li se insert až na další stranu, než byl vložen. *Vlož insert až na té straně, kam byl odsunut.*
- Je-li značka zpracována na konci strany, ale již se tam nevešla. *Zlom ručně stranu (`\eject` nebo `\newpage`) v místě, kde to udělal  $\TeX$  sám.*

Představu o úspěšnosti správného umístění zvýraznění si lze udělat z následujícího reálného textu: v něm z celkového počtu 172 značek bylo třeba 26 poopravit.

## 3. Makra `backgr` z pohledu řešitele

Tato sázečí úloha může být řešena dvěma způsoby:

Synchronně – mezi vstupní materiál se vloží značky a zároveň s textem se zpracovává jeho zvýraznění.

Problémy úlohy *a problémy  $\TeX$ u*:

- Řešení systémově závislé. *Zvýraznění musí umět použitý `dvi` ovladač. Nejsnáze snad vyhoví PostScriptové specialy.*

Asynchronně – změří se velikost a poloha zvýrazněného bloku textu a pak se zvlášť vysází text a zvlášť jeho zvýraznění. Důležité je zajistit, aby obě části na sobě přesně seděly.

Problémy úlohy *a problémy  $\TeX$ u*:

- Rozdělení bloku přesahujícího jednu stranu na více částí podle stránkového zlomu. *Expanze maker může probíhat na jiné straně, než*

*sazba materiálu obsažených v těchto makrech.*

- Počítat s pružnými výplňky hlavního vertikálního módu. *Nelze přímo zjistit součet pružných výplňků do místa optimálního, T<sub>E</sub>Xem nalezeného, zlomu strany.*
- Vyrovnat se s plovoucími objekty typu insert. *Insert se může objevit na začátku strany (tj. před značkami, které do strany již vstoupily i těmi, které teprve přijdou) i na jejím konci (tj. za značkami) a také může odplout na následující stranu.*

Protože rozumné synchronní řešení vede na ne zcela triviální použití PostScriptu (a ten znám podstatně méně než jazyk T<sub>E</sub>Xu) a protože jsem si neuvědomoval nastíněné omezení T<sub>E</sub>Xu, zvolil jsem asynchronní implementaci. V dalším textu vysvětlím její nedostatky.

### 3.1. Nástin řešení

Makra `backgr` pracují ve dvou fázích. V první expandují značky vertikálně ohraničující zvýrazněný text. Druhá fáze probíhá ve výstupní rutině a realizuje vlastní sazbu boxu se zvýrazňujícím materiálem. Podívejme se podrobněji, co je třeba v každé fázi udělat.

1. Značky bloku textu ke zvýraznění. Každá značka způsobí, že T<sub>E</sub>X zjistí výšku sazby na aktuální straně pomocí primitivů `\pagetotal` a jemu podobných a zapíše ji do *(token)* registru `\backgrlist`. Značka `\backgr`, otevírající zvýrazněný text, dále nastaví logickou proměnnou `\ifmiddlebg` na pravdivou hodnotu, uzavírací značka `\endbackgr` na nepravdivou. Víme tak v každém místě sazby, zda jsme uvnitř zvýrazněného bloku nebo ne.
2. Výstupní rutina. V ní provedeme následující kroky:
  - (a) Uzavření boxu. Pokud je `\ifmiddlebg` pravda, pak zvýrazněný blok textu přechází z kompletované strany na další. Uzavřeme tedy blok na této straně vložením značky `\endbackgr` s takovými parametry, jako by byla na dolním konci tiskového zrcadla aktuální strany (dáno výškou a hloubkou boxu 255).
  - (b) Tisk zvýraznění. Vysází se obsah registru `\backgrlist`. Každý pár otevírací a uzavírací značky do něj přidá makro s názvem `\backgrbox` s množstvím parametrů o aktuálním místě sazby. Toto makro spočítá usazení zvýrazňujícího boxu na straně. Poté dle nastaveného stylu zvýraznění vysází (před vlastní obsah strany) zvýrazňující vbox nulových rozměrů. Po sazbě zvýraznění se registr `\backgrlist` vyprázdní.
  - (c) Otevření boxu. Pokud je `\ifmiddlebg` pravda, pak zvýrazněný text pokračuje z předchozí strany. Proto do `\backgrlist` vložíme značku otevření bloku s nulovými parametry výšky aktuálního bodu sazby.

### 3.2. Bod sazby a nestejnostranná expanze a sazba

Jaké nám  $\TeX$  nabízí informace o aktuální pozici sazby na kompletované straně? Jsou to údaje uložené v registrech začínající jménem `\page`. Zde je jejich přehled:

`\pagetotal` ( $t$ ) popisuje výšku dosud vloženého materiálu, tedy součet výšek všech boxů, linek, kernů a mezer typu  $\langle glue \rangle$ .

`\pagegoal` udává cílovou výšku materiálu. Je roven `\vsize`, ale vložením každého insertu je zmenšena o jeho skutečnou výšku.

`\pagedepth` ( $d$ ) je hloubka posledního materiálu; je-li jím kern nebo mezera typu  $\langle glue \rangle$ , je roven  $0\ pt$ .

`\pageshrink` shromažďuje stažení všech registrů typu  $\langle glue \rangle$  řádu 0. Hodnoty stažení vyšších řádů se nemohou v hlavním vertikálním módu vyskytnout.

`\page[|fi[l|ll|lll]]stretch` reprezentují hodnoty roztažení všech registrů typu  $\langle glue \rangle$  zvlášť pro každý řád roztažení 0 až 3.

Hodnoty těchto registrů se počítají z materiálu na aktuální straně a aktualizují se při vložení boxu, linky, kernu nebo mezery do hlavního vertikálního módu.

Pomocí těchto registrů můžeme spočítat *aktuální výšku sazby*. Opomineme-li prozatím pružné výplňky a plovoucí objekty vkládané před značku, platí

$$v_{\text{akt}} = t + d$$

Tuto hodnotu postoupíme spolu s hodnotami stažení a roztažení výstupní rutíně prostřednictvím  $\langle token \rangle$  registru `\backgrlist`.

Takto zjištěný aktuální bod sazby je platný jen tehdy, pokud se všechen expandovaný materiál ocitne na kompletované straně. To se však nemusí stát, protože algoritmus hledání optimálního zlomu často načte a expanduje více sazby, než se vejde na stranu. Mezi tím textem, který se nevešel, mohly být také naše značky `\backgr` nebo `\endbackgr`. Jejich expanzí se zjistily hodnoty aktuálního bodu sazby pod hranicí tiskového zrcadla a podle nich se vysází zvýraznění. Avšak přetečený (ale již expandovaný) text se vrátí do přípravné oblasti a objeví se až na straně následující [1, str. 238].

Rozumné řešení tohoto problému *nestejnostranné expanze a sazby*<sup>2</sup> části tiskového materiálu neexistuje. Řešení by vyžadovalo obejít algoritmus hledání optimálního zlomu strany. Mohli bychom ručně přidat penaltu menší nebo rovnou  $-10000$  a vynutit zlom, nebo střídat materiál do pomocných boxů a použít primitiv `\vsplit`. Kromě jisté těžkopádnosti nám může vadit, že tím dosáhneme nekompatibility se standardním zalámáním textu. (Když ne pro nic jiného, tak pro nemožnost použití mechanismu insertů.)

### 3.3. Výpočet pružných výplňků na straně

Nyní budeme pokračovat v určení referenčního bodu sazby v místě značek. Z předchozí podkapitoly víme, jak spočítat jeho polohu, neuplatní-li se pružné

---

<sup>2</sup>Toto pojmenování se mně zdá výstižnější než používaný termín asynchronní překlad.

výplňky hlavního vertikálního módu. Nespokojme se s tímto omezením a snažme se spočítat korekci polohy podle vložených výplňků. Tento výpočet bude probíhat již ve výstupní rutině.

Pokud je celkový materiál do místa zlomu vyšší než cílová výška strany (`\ht255>\pagegoal`), pak se uplatní stažení. Počítaný referenční bod se posune vzhůru úměrně poměru stažení v tomto místě a stažení celé kompletované strany. Podobně, je-li celkový materiál nižší než cílová výška strany, pak se uplatní roztažení nejvyššího přítomného řádu.

Zatímco hodnoty stažení a roztažení v místě značek jsou známy (značka je přečte z registrů `\page*` a uloží do `\backgrlist`), součet všech stažení a roztažení až do místa optimálního zlomu, stejně jako cílová výška strany, znám není. Registry `\page*` totiž ve výstupní rutině uchovávají hodnoty veškerého expandovaného materiálu, tedy i toho, který je až *za* optimálním zlomem (vliv nestejnostranné expanze a sazby). Registry `\pagetotal` a `\pagedepth` lze zjistit změřením boxu 255, ale pro pružné výplňky nic takového T<sub>E</sub>X přímo nenabízí [1, pozn. k `\pagetotal`, str. 415].<sup>3</sup>

### 3.4. Plovoucí objekty a bod sazby

Standardní plovoucí objekty v plainu i L<sup>A</sup>T<sub>E</sub>Xu se mohou vyskytnout na začátku strany, na pozici vložení, na konci strany a mohou být i odsunuty dále a na aktuální straně se vůbec neobjevit. Jejich konečné umístění má vliv na bod sazby značek zvýraznění v případě, že se vloží před značku. Přestože používají primitiv T<sub>E</sub>Xu `\insert`, jsou algoritmy vkládání boxů s plovoucím materiálem závislé na formátu. Z důvodů značné složitosti kompletního ošetření vlivu plovoucích objektů na bod sazby, makra `backgr` téměř všechny inserty ignorují.<sup>4</sup> Jen inserty patřící nahoru (v L<sup>A</sup>T<sub>E</sub>Xu ty, které mají nastaven příznak `top`, v plainu třída `\topinsert`) způsobí připočtení výšky materiálu těchto insertů k dosud spočítanému bodu sazby. Započítají se i případné pružné výplňky uvnitř insertů.

Sečteme-li hodnotu referenčního bodu sazby, korekce vlivem pružných výplňků, vlivu „top“ insertů a uživatelem zadaného zpřesnění, dostaneme (z obou značek bloku) čtvrtý a pátý parametr makra provádějící vlastní zvýraznění (kap. 2.4). Je již na uživateli, jak spolu s dalšími čtyřmi údaji tuto informaci využije v definici stylu zvýraznění.

---

<sup>3</sup>Makra `backgr` se snaží tento problém řešit rozkladem boxu 255 na jednotlivé elementy, počítat výplňky a výsledek separovat na stažení nebo roztažení daného řádu pomocí výpisu výsledku primitivem `\the`. Tento algoritmus však často selže, protože T<sub>E</sub>X neposkytuje primitiva `\unmark` a `\unwrite` potřebná k rozložení libovolného boxu. V takovém případě jsou použity (obvykle větší) hodnoty z `\page*`. Jiným, snad úplným, ale krkolomnějším, řešením by bylo dvoupřechodové načtení log souboru s nastavením `\tracingpages` na kladnou hodnotu a extrakci potřebných hodnot z něj.

<sup>4</sup>Takovéto chování je správné pro nemalou množinu insertů: umístěné dole na straně (i `\footnote`), na samostatné straně, ty, co se zatím nevešly a plainový `\midinsert`, který není insertem.



## 4. Co jsem v T<sub>E</sub>Xu postrádal

Zde shrnuji omezení T<sub>E</sub>Xu, která činila řešení popisované úlohy obtížné nebo nedostatečně přesné.

- Možnost znovuzpracování expandovaného ale dosud nepoužitého materiálu z předchozí strany k opětovné expanzi. Toto zlepšení by vyřešilo mnoho jiných problémů, např. sazbu do různě širokého tiskového zrcadla na dvou po sobě jdoucích stranách.
- Možnost získat ve výstupní rutině informace o realizovaném stránkovém zlomu, tj. hodnoty registrů `\page*` (3.2) pro optimální cenu zlomu. Nebo obecněji, získat informace o hodnotách stažení a roztažení boxu podobně, jak lze primitivy `\wd`, `\ht` a `\dp` zjistit jeho rozměry.
- Zjistit polohu vložené značky v rámci boxu.
- Nepřítomnost primitiv umožňující odebrat z boxu mark a write, tedy něco jako `\unmark` a `\unwrite`.

## Odkazy

- [1] Petr Olšák. *T<sub>E</sub>Xbook naruby*. Konvoj Brno, 1. vydání, 1997.
- [2] Michail I. Schlesinger a Václav Hlaváč. *Matematická teorie rozpoznávání*. Vydavatelství ČVUT, Praha, 1. vydání, během 1999.

Vít Zýka, [zyka@cmp.felk.cvut.cz](mailto:zyka@cmp.felk.cvut.cz)

---

---

## Recenze systému pdfT<sub>E</sub>X

JIRÍ MŽOUREK

## Úvod

PdfT<sub>E</sub>X spojuje, jak již sám název napovídá, na jedné straně typografický systém T<sub>E</sub>X, na straně druhé hypertextový formát PDF firmy Adobe.

Cílem vývoje bylo vytvořit systém umožňující jak přímý hypertextový výstup do formátu PDF, tak i výstup do DVI. Toho bylo dosaženo a konečný výsledek vypadá v obou formátech identicky.

PdfT<sub>E</sub>X je zpětně kompatibilní s T<sub>E</sub>Xem. Pro hypertextové rozšíření se používají nové příkazy. Současně je k dispozici i pdfL<sup>A</sup>T<sub>E</sub>X.