

Zpravodaj Československého sdružení uživatelů TeXu

Zdeněk Wagner

LaTeX v sazečské praxi

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 18 (2008), No. 3, 161–174

Persistent URL: <http://dml.cz/dmlcz/150058>

Terms of use:

© Československé sdružení uživatelů TeXu, 2008

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>

T_EX je znám zejména v akademickém světě a používán při psaní odborných publikací. Mnoho lidí ví, že pomocí T_EXu lze vytvořit kvalitní sazbu. V dnešní době, kdy převládají programy s grafickým uživatelským rozhraním, v němž jsou důležité informace před uživatelem skryty, je však obtížné získat návod, jak připravit pomocí T_EXu soubor pro osvit či digitální tisk. V přednášce budou předvedeny postupy, jak lze v sazečské praxi využít L^AT_EX. Bude představeno několik nových makrobálíčků, které usnadňují přípravu letáků, pozvánek i sazbu knih včetně obálek a přebalů.

1. Úvod

Sazečská práce L^AT_EXisty má dva aspekty. První aspekt je typografický. Pravidla typografie však nejsou závislá na použitém nástroji. S nástrojem buď zacházet umíme, nebo neumíme. Neexistuje L^AT_EXová typografie, máme jen typografii jako takovou. Pokud výsledek není kvalitní, nebývá to chybou nástroje. Druhý aspekt je ryze technický. Mít dobrý nástroj nestačí. Máme-li dosáhnout kvalitního výsledku, musíme s tím nástrojem umět pracovat.

Typografická pravidla jsou skoro stejně stará jako písmo samo. Učebnice typografie tedy najdeme celkem snadno. V tomto článku se proto zaměříme výhradně na vysvětlení technických postupů, a to jen těch, kdy řešíme jistou problematiku pomocí L^AT_EXu. Takové návody jsou totiž různě roztrženy po internetu i jiných zdrojích. Navíc mají úzkou návaznost na technická zařízení, jež se rychle vyvíjejí, a návody tudíž zastarávají.

Uživatel komerčních sazečských systémů je v poněkud jiné situaci. V grafickém uživatelském rozhraní zvolí, pro jaké výstupní zařízení se má sazba připravit. Program nastaví správné parametry, které sazeč vůbec nemusí znát. Nemusí vůbec rozumět principu, jak dané zařízení pracuje. T_EX žádné takové rozhraní nemá. Uživatel musí vědět, jak má výstupní ovladač instruovat, aby výsledek jeho práce byl použitelný. Tato zdánlivá nevýhoda se změní ve výhodu v okamžiku, kdy se objeví nové zařízení. Uživatel komerčního programu, který takové zařízení v nabídce nemá, si musí koupit novou verzi pracovního nástroje, zatímco T_EXista jen ve svém postupu změní několik parametrů.

Konkrétní metody použití L^AT_EXu ve spojení se současnými výstupními zařízeními (osvitovými jednotkami, tiskovými stroji CTP) také nutně zastarávají. Zaměříme se tedy spíše na logiku použití L^AT_EXu, abyste metody, vysvětlené v tomto článku, mohli v budoucnu modifikovat a použít s modernější technikou.

2. Rozvržení stránky

\TeX tradičně počítá souřadnice od počátku vzdáleného (1 in, 1 in) od levého horního rohu papíru. \LaTeX ista však může být v roli pouhého sazeče majícího za úkol vytvořit sazbu podle dodaného grafického návrhu. Grafik jen zřídka přemýšlí v \TeX ových pojmech. Téměř vždy měří rozměry tiskového zrcadla od okraje papíru. Když pak po několikáté píšete v preambuli svého dokumentu

```
\addtolength{\topmargin}{-1in}
\addtolength{\oddsidemargin}{-1in}
\addtolength{\evensidemargin}{-1in}
```

usoudíte, že nadešel čas k tomu, abyste si vytvořili vlastní balíček maker, který přizpůsobí zadávání rozměrů sazby přemýšlení grafika. Při analýze problému pokročíte ještě dále. Grafik někdy určí velikost obou okrajů, ale \LaTeX očekává zadání levého okraje a šířky textu. Je tedy na uživateli, aby údaje od grafika převedl na hodnoty vyžadované \LaTeX em. Totéž platí pro horní a dolní okraj. Protože rozměry sazby nastavujeme v každém dokumentu, je volba pohodlného balíčku maker nejdůležitější úlohou každého sazeče.

2.1. Proč ne geometry?

Odpověď na otázku, položenou v názvu této sekce, je do značné míry subjektivní. Na balíčku `geometry` není vůbec nic špatného. Osobně jej odmítám v podstatě z důvodů historických.

Vlastní makra pro nastavení rozměrů sazby jsem začal vyvíjet hned na počátku své \LaTeX ové éry, tj. v době, kdy balíček `geometry` ještě neexistoval. Makra jsem vyvíjel podle svých vlastních potřeb, aby se mi s nimi dobře pracovalo. Logika použití balíčku `geometry` je velmi podobná tomu, co v mé dílně vzniklo postupným vývojem během 17 let. Přestože jsou možnosti balíčku `geometry` velké, chybí mi v něm vlastnosti, které pro svoji práci potřebuji. Balíček je dosti rozsáhlý a budování jeho nadstavby s rizikem, že se budoucí verze mohou lišit v implementaci některých maker, je pracnější než pokračování vývoje vlastního balíčku. Výsledkem vývoje, během něhož byl několikrát odeslán soubor do míst, kam chodí mrtvé bajty, a napsán znovu od začátku, je balíček `zwpagelayout`, jehož použití se zdá být zcela univerzálním.

V okamžiku psaní tohoto článku je balíček již odladěn a rutinně používán, ale není zveřejněn, protože není dokončena dokumentace. Hlavní vlastnosti balíčku si popíšeme v následujících sekcích. Zveřejnění balíčku na CTAN bude oznámeno na webu autora [13].

2.2. Rozměry papíru

Běžné distribuce T_EXu dovolují nastavení defaultních rozměrů papíru na A4, nebo letter. V typografické praxi si však s těmito formáty nevystačíme. Několik voleb najdeme ve standardních L^AT_EXových třídách, pravděpodobně bychom našli též existující balíčky. Náš balíček by tím však byl neúplný. Rozměry standardních formátů nechceme nosit v hlavě. Pomocí parametrů tedy můžeme zvolit libovolný formát z řad A, B, nebo C, ale i některé formáty americké. Můžeme tedy psát:

```
\usepackage[b5]{zwpagelayout}
```

Orientaci papíru můžeme ovlivnit použitím parametru PORTRAIT, nebo LANDSCAPE. Standardně se předpokládá orientace na výšku.

Občas potřebujeme tisknout na papír jiných rozměrů. Parametr PAPERSIZE nám umožní zadat libovolnou velikost. Při tvorbě vizitky standardní velikosti tedy použijeme:

```
\usepackage[papersize={9cm,5cm}]{zwpagelayout}
```

Tím ještě možnosti definice velikosti nekončí, ale více si ukážeme v sekci 2.5, kde budeme vytvářet obálku a přebal knihy.

Příkazy pro nastavení velikosti papíru uloží hodnoty do registrů `\paperheight` a `\paperwidth`. Tyto hodnoty jsou v okamžiku zpracování `\begin{document}` použity k nastavení rozměrů tiskového média podle toho, jaký výstupní ovladač je detekován. Balíček umí nastavit papír pro pdfL^AT_EX, dvips a (x)dvi_{pdfm}(x). Nastavení rozměrů pro výstupní ovladač je odloženo, protože některé balíčky, např. `hyperref`, nastavují rozměry podle svého uvážení. Zmíněné ovladače se bez výjimky řídí posledním příkazem, takže `zwpagelayout` vyhrává.

2.3. Ořezové značky

Ořezové značky potřebujeme zejména v případě, kdy tiskneme na papír nestandardních rozměrů. Starší balíček [14] byl závislý na postscriptu. Přitom ořezové značky lze tvořit prostým primitivem `\vrule`, tedy nezávisle na výstupním ovladači. Umístění ořezových značek odvodíme z hodnot registrů `\paperwidth` a `\paperheight`. Je tedy přirozené, že makra pro ořezové značky nadefinujeme v balíčku `zwpagelayout`.

Pro tvorbu ořezových značek jsou nejdůležitější čtyři parametry. Parametr CROPMARKS zapne tisk ořezových značek. Programově je to realizováno tak, že se zvětší hodnoty `\paperwidth` a `\paperheight` a celá sazba se použije nastavením kladných hodnot do registrů `\hoffset` a `\voffset`. Běžný uživatel o těchto registrech neví a vzdělanější uživatel si musí pamatovat, že jsou pro něj tyto registry tabu.

Při vizuální kontrole se hodí, když máme okolo stránky rámeček. Ten zapneme parametrem CROPFRAME. Ve finální verzi jej samozřejmě vypneme.

Standardně se očekává jednobarevný tisk, takže ořezové značky jsou tištěny černě. Barvu zapneme parametrem `COLOR`. I zde máme mnoho dalších možností, o nichž se zmíníme později.

Kromě ořezových značek potřebujeme též stránky označit vhodným identifikačním textem. Takový text nastavíme v parametru `CROPTITLE`. Pokud text obsahuje čárku, musíme jej vložit do složených závorek. Při zpracování parametrů ignoruje příkaz `\usepackage mezery`, proto je musíme vložit použitím kontrolního symbolu `_`.

Příklady použití uvidíte v následujících sekcích tohoto článku. Proto jsme úvod k ořezovým značkám předřadili.

2.4. Rozměry tiskového zrcadla

Nastavení levého a pravého okraje při oboustranné sazbě je ve standardním \LaTeX u poněkud těžkopádné. Uživatel musí vypočítat správné hodnoty `\oddsidemargin`, `\evensidemargin` a `\textwidth`. Balíček `zwpagelayout` nastavení usnadňuje. Předpokládá se, že nastavujeme rozměry textu na liché straně, tj. té, která je v otevřené knize vpravo. Okraje na protilehlé straně budou dopočítány symetricky. Rozměry nastavíme pomocí parametrů `LEFTMARGIN`, `RIGHTMARGIN` a `TEXTWIDTH`. Měříme od kraje papíru. Protože jsme však již rozměr papíru zadali, uvedeme pouze dvě z výše uvedených hodnot. Zadáme-li pouze jeden rozměr, předpokládá se, že levý i pravý okraj jsou stejné.

Horní a dolní okraj nastavíme pomocí parametrů `TOPMARGIN` a `BOTMARGIN`. Parametry `HEADHEIGHT`, `HEADSEP` a `FOOTSKIP` slouží k nastavení rozměrů odpovídajících parametrů \LaTeX u. Parametrem `TEXTHEIGHT` nastavíme výšku textu včetně záhlaví a zápatí. Opět stačí zadat dva z trojice rozměrů, třetí rozměr se dopočítá z rozměru papíru. Hlavní odlišnost však spočívá v tom, že při sazbě knih většinou dodržujeme řádkový rejstřík. Hodnotu rozměru `\textheight` proto musíme vypočítat jako násobek `\baselineskip` zvětšený o `\topskip`. Při sazbě akcidenčních tiskovin však takovou korekci obvykle nechceme. Zakážeme ji parametrem `STRICTHEIGHT`. Bez tohoto parametru algoritmus implementovaný v balíčku `zwpagelayout` zmenší hodnotu `\textheight`. Aby zůstal zachován součet všech rozměrů určených uživatelem, zvětší se o rozdíl hodnota registru `\footskip`. Pokud má kniha živé záhlaví a nemá zápatí, dali bychom přednost úpravě velikosti `\headsep`. Dosáhneme toho parametrem `ADJUSTHEADSEP`.

Oba způsoby zarovnání jsou ukázány na obrázku 1. Ukázka vlevo byla vytvořena příkazem:

```
\usepackage [c8,margins=6mm,headheight=4mm,headsep=4mm,
  cropmarks,cropframe,croptitle=Adjust\footskip]{zwpagelayout}
```

Ukázku vpravo jsme získali pomocí:


```
\usepackage[c8,margins=6mm,headheight=4mm,headsep=4mm,
  adjustheadsep,
  cropmarks,cropframe,croptitle=Adjust\headsep]{zwpagelayout}
```

Všimněte si použití parametru `MARGINS`. Ten slouží k nastavení všech okrajů na stejnou hodnotu.

2.5. Obálka a přebal

V sekci 2.4 jsme uvedli, že chybějící hodnoty rozměrů tiskového zrcadla lze dopočítat z velikosti papíru. Balíček `zwpagelayout` však disponuje i opačnou funkcí: výpočtem velikosti papíru z rozměrů tiskového zrcadla a okrajů. Lze dopočítat oba rozměry, nebo jen jeden z nich. Tuto možnost oceníme zejména při tvorbě knižních obálek a přebalů.

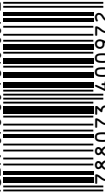
V sazečské praxi se často octneme v situaci, kdy kniha ještě není hotova, není vybrán papír, takže nelze určit tloušťku hřbetu. Přesto by zákazník chtěl vidět návrh obálky a přebalu. Tloušťku hřbetu lze pouze přibližně odhadnout. Jaká je tedy celková šířka přebalu? Pokud si ji vypočteme sami, můžeme se dopustit chyby. Automatický výpočet nás tohoto rizika zbaví. Přebal, který vidíte na obrázku 2, jsme vytvořili následujícím kódem:

Výšku přebalu známe, proto jsme ji mohli zadat, vynechali jsme pouze šířku. Povšimněte si též použití parametru `BOTMARGIN` bez hodnoty. Spodní okraj pak převezme velikost z parametru `TOPMARGIN`. Parametr `SPINE` určuje tloušťku hřbetu. Tím se též mění význam parametru `TEXTWIDTH`, jímž nyní určujeme šířku přední, resp. zadní strany. Celková šířka (bez prostoru pro ořezové značky) tedy bude 148 mm. Pro kontrolu využijeme též rámeček, jenž vyznačí nejen obrysy přebalu, ale i chlopně a hřbet. Hodnoty uvedené v parametrech jsou uschovány v makrech, která využijeme pro nastavení rozměrů jednotlivých boxů. Když se pak dozvíme skutečnou tloušťku hřbetu, změníme pouze hodnotu parametru `SPINE`. Celá sazba přebalu se pak přepočte automagicky.

Při tvorbě přebalu jsme si též představili balíček `ean13isbn` [12] pro tisk čárového kódu a ISBN ve formátu platném od 1. ledna 2007.

3. Barevný tisk

$\text{T}_{\text{E}}\text{X}$ nemá žádné nativní nástroje pro práci s barvou. Práce s barvou je závislá na schopnostech výstupního ovladače, jemuž $\text{T}_{\text{E}}\text{X}$ odesílá povely primitivem `\special`, pdf $\text{T}_{\text{E}}\text{X}$ lze ovládat pomocí primitivu `\pdfliteral`, což bylo přehledně popsáno v článku F. Chvály [5]. Balíček `color` obsahuje makra pro změnu barvy písma, jež expandují na příslušné povely pro daný výstupní ovladač. V dokumentaci najdete popis příkazů, ale nedočtete se nic o tom, jak připravit soubor pro osvit.

Zadní záložka	Zadní strana ISBN 978-80-7340-097-2  9 788073 400972	Hřbet	Přední strana L ^A T _E X	Přední záložka
---------------	---	-------	--	----------------

Obrázek 2: Ukázka přebalu

Možností použití barev v tiskovinách je několik. V dalších sekcích si probereme hlavní možnosti a ukážeme jejich realizaci v L^AT_EXu. Použití barvy je však poměrně komplikovaná tematika, zde bude vysvětlena pouze její část.

3.1. Jednobarevný ofsetový tisk

Jednobarevný ofsetový tisk je z hlediska T_EXu zcela jednoduchý. Od sazeče vlastně nic speciálního není očekáváno. Sazba je připravena černě, jako by se o barevnou sazbu nejednalo. Vše se odehraje až při tisku. Místo černé barvy tiskař nalije do stroje barvu jinou.

3.2. Plnobarevný tisk

Balíčky pro práci s barvou nabízejí použití dvou různých barevných prostorů, RGB a CMYK. Z tabulky v článku F. Chvály [5] by se mohlo zdát, že na volbě barevného prostoru nezáleží. Opak je však pravdou. Tiskové stroje pracují výhradně v prostoru CMYK. Navíc teoretické poučky o míchání barev platí jen přibližně. Skutečné procesní (tiskové) barvy se od ideálních, spektrálně čistých barev, na nichž jsou teoretické poučky založeny, značně liší. Směs stejných množství procesních barev tedy nedává šedé, ale hnědé tóny. Černá se přidává ze dvou důvodů, jednak proto, abychom skutečně mohli tisknout šedé tóny, a jednak proto, že jedna barva schne rychleji než směs tří tiskových barev. Situace je však ještě složitější, protože existuje více prostorů RGB a více prostorů CMYK. Zejména rozdíl mezi evropskými prostory Euroscale a americkými SWOP je značný. Pro přepočítání mezi barevnými prostory neplatí žádný jednoduchý vzoreček, ale používají se ICC profily [6]. Problematika však přesahuje rámec L^AT_EXového řešení. Srozumitelné vysvětlení s odkazy na další zdroje najdete v seriálu *Grafika v UNIXu* [2].

Pamatujme si tedy, že sazbu barevného dokumentu musíme připravit v prostoru CMYK, a to v tom správném, který odpovídá použitým tiskovým strojům. Nezbyvá tedy než navštívit tiskárnu, kde se bude tiskovina vyrábět, zjistit, jaké stroje budou použity, a případně z tiskárny získat ICC profil, nebo vytisknutý barevný vzorník.

Plná barevnost je realizována tiskem čtyřmi procesními barvami, azurovou (CYAN), purpurovou (MAGENTA), žlutou (YELLOW) a černou (BLACK). Barevný dokument musíme tedy rozdělit na čtyři barevné pláty. Než však začnete studovat, jak se barevné separace v praxi dělají, poraďte se s pracovníky DTP studia, kde se bude soubor zpracovávat. Separace jsou totiž v současné době vyžadovány jen zřídka. Osvitové jednotky očekávají na vstupu jediný soubor, jaký byste tiskli doma na vlastní inkoustové či laserové tiskárně. Může to být postscript, nyní stále častěji PDF. Algoritmy pro vytvoření separací obsahuje software osvitové jednotky. Digitální tiskové stroje (CTP) pracují, laicky řečeno, v podstatě stejně jako domácí barevné tiskárny, jsou jen větší a rychlejší. Ani v tomto případě proto separace dělat nebudeme.

Postscript [10] obsahuje řadu operátorů pro převod mezi barevnými prostory. Očekávali bychom tedy, že barvu při zadání barvy v jakémkoliv prostoru osvitka vše zařídí pomocí operátorů `/setblackgeneration`, `/setundercolorremoval` a jiných. Jak jsme si již řekli, barevných prostorů je mnoho. DTP studia nechtějí riskovat, že barvy převedou z jiného RGB do jiného CMYK. Barevné konverze jsou proto záměrně vypnuty a je na sazeči, aby sám převedl barvy do správného barevného prostoru, z nichž osvitová jednotka jen triviálním postupem vytvoří barevné separace.

3.3. Dvoubarevný a třibarevný tisk přímými barvami

V barevném prostoru CMYK lze realizovat jen omezený počet barev. Sytější barvy lze sice fyzicky namíchat, ale v prostoru CMYK je vytisknout nemůžeme. Pokud nepotřebujeme plnou barevnost, můžeme tisknout přímými barvami. Bývá to výtvarně zajímavější i levnější.

Pro tisk musíme opět připravit separace. Vytváření vlastního software, nebo definice přímých barev tak, aby tomu běžné programy rozuměly, může být namáhavé. Lze však simulovat přímé barvy pomocí barev procesních. Ukažme si to na jednoduchém dokumentu, který vidíte na obrázku 3. Zde máme černý a červený tisk na žlutém pozadí¹. Červenou barvu přiřadíme purpurové separaci (MAGENTA). Azurová (CYAN) zůstane nepoužita.

Balíček `zwpagelayout` při barevném tisku předpokládá prostor CMYK, ale barvy lze předefinovat. V parametru `COLORS` uvedeme jména barev a jejich hodnoty v barevném prostoru použitém pro osvit, v našem případě to bude defaultní CMYK. Zde je celý zdrojový kód:

V DTP studiu pak řekneme, že azurový plát je prázdný a nemá se svítit. Aby skutečně zůstal prázdný, předefinovali jsme parametrem `CROPCOLOR` barvu ořezových značek.

Osvitové jednotky mívají zapnutý přetisk černé. Na žlutém plátu proto vidíte bílý nápis v místě, kde bude červený text. V místě černého textu však bílý nápis nenajdeme. Před přiřazením přímých barev procesním barvám tedy musíme trochu přemýšlet.

3.4. Barevné separace pomocí maker

Polotónový tisk vyžaduje použití rastru. Pokud navíc tiskneme barevně, musíme též měnit úhel rastru. Postará se o to software osvitové jednotky. Použití rastru však snižuje čitelnost písma. Pokud se přímé barvy nepřekrývají, bývá výhodnější, když rastr použit není, nebo má úhel 45°, což poskytuje nejlepší vizuální vjem.

¹V nebarevně vytištěném Zpravodaji si musíte barvy domyslet.



Obrázek 3: Ukázka tisku přímými barvami; neseparovaný dokument s kontrolním rámečkem (nahore) a žlutý plát (dole)

Pak ovšem nelze využít standardní software, ale separace si musíme udělat sami. Jednou z možností je řešení na úrovni maker. Tak jsme udělali žlutý plát na obrázku 3. Zde je kompletní kód:

Všimněte si, že jsme změnili barevný prostor a předefinovali všechny barvy. Dokument pak zůstal stejný. Obecný postup, jak dělat separace, nejspíš neexistuje. Stačí však trochu invence k vytvoření vlastních maker.

3.5. Barevné separace pomocí PostScriptu

V úvodu jsem si řekli, že se budeme zabývat pouze L^AT_EXovými řešeními. Zde uděláme výjimku. Způsob míchání barev může být komplikovaný a programování vlastních separací pomocí makrojazyka T_EXu zbytečně pracné. Pokud při zpracování dokumentu využíváme ovladač `dvips`, nabízí se ještě jedna zajímavá možnost: separaci barev provést balíčkem Aurora [3].

4. Příprava ke korekturám

Korektury jsou nezbytnou součástí práce. Korektury nikdy neděláme na obrazovce, vždy ve vytištěné verzi. A neděláme je sami, navíc stejně potřebujeme imprimatur od autora či nakladatele. Formát knihy je často A5 či menší, zatímco naše stolní tiskárna používá formát A4. Abychom šetřili naše lesy, raději poskládáme na papír dvojice stránek tak, jak budou v otevřené knize, tj. liché stránky vpravo. První papír tedy bude obsahovat pouze stranu 1. Úlohu lze řešit mnoha různými způsoby. My si předvedeme jednoduché makro, které poskládá již hotový PDF.

První, co musí naše makro udělat, je zjištění počtu stran. Vyřešíme to pomocným makrem se dvěma parametry. Prvním parametrem je jméno PDF souboru, druhým parametrem je jméno sekvence, v níž bude počet stran uložen jako makro.

```
\def\getpdfnumpages#1#2{\pdfximage{#1}%  
  \edef#2{\the\pdfastximagepages}}
```

Další pomocné makro vytiskne zvolenou stránku v rámečku.

```
\def\ZW@pdfbox#1#2{\fbox{\includegraphics [page=#1]{#2}}}
```

Vlastní definici uvedeme bez vysvětlení. Všimněte si, že jména maker ob-
sahují zavináče. Musíte proto vložit definici do stylového souboru, nebo mezi
`\makeatletter` a `\makeatother`.

```
\def\DvaVedleSebe#1{\getpdfnumpages{#1}\ZW@pdfnumpages  
  \fboxrule 1pt \fboxsep -\fboxrule  
  \typeout{#1, number of pages = \ZW@pdfnumpages}}
```

```

\ZW@pdfpgnum2
\ vbox to \textheight{\hsize\textwidth
\ hbox to \textwidth{\hfill \ZW@pdfbox{1}{#1}}\vss}
\ loop
  \ZW@tempnum\ZW@pdfpgnum
  \advance\ZW@tempnum1
  \ifnum\ZW@pdfpgnum<\ZW@pdfnumpages
    \ vbox to \textheight{\hsize\textwidth
    \ hbox to \textwidth{\hss
      \ZW@pdfbox{\number\ZW@pdfpgnum}{#1}%
      \ZW@pdfbox{\number\ZW@tempnum}{#1}\hss}\vss}
  \else
    \ vbox to \textheight{\hsize\textwidth
    \ hbox to \textwidth{%
      \ZW@pdfbox{\number\ZW@pdfpgnum}{#1}\hfill}\vss}
  \fi
  \ifnum\ZW@tempnum<\ZW@pdfnumpages
    \advance\ZW@pdfpgnum2
  \repeat}

```

Makro potřebuje pro svoji práci dva čítače:

```

\newcount\ZW@pdfpgnum
\newcount\ZW@tempnum

```

5. Čárové kódy

V sekci 2.5 jsme si předvedli čárový kód s ISBN. Není to zdaleka jediné použití čárového kódu. Jinou aplikaci představili Sojka a Hrad [4]. Čárových kódů však existuje více druhů [1] a vzhledem k jejich důležitosti byly vytvořeny komerční i volně šiřitelné fonty. Sazba čárových kódů pomocí fontů ovšem naráží na různá úskalí. Čárové kódy skupiny 2/5 a Code39 mají velmi variabilní rozměry. Liší se nejen poměr výšky a šířky, ale i poměr mezi slabou a tlustou čarou. Je dost pravděpodobné, že konkrétní snímač nebude schopen přečíst čárový kód vytištěný fontem, který byl původně vytvořen pro jinou aplikaci.

Elegantní řešení nabízí balíček `ean13.tex` [7], který vytvořil Petr Olšák a popsal jej česky ve Zpravodaji [8] i anglicky v TUGboatu [9]. Čárový kód je vytvořen pomocí maker primitivem `\vrule`, žádný font není nutný. Makra lze parametrizovat. Čárový kód můžeme tedy snadno přizpůsobit požadovaným rozměrům.

Ze stejné myšlenky vychází balíček `makebarcode` [11]. V něm je definováno obecné makro, jež generuje čárový kód pomocí tabulky a několika parametrů.

Můžeme jej využít např. pro identifikaci záznamovým médií HP, jak je ukázáno na obrázku 4.



Obrázek 4: Čárový kód CLNU01L1 pro čisticí kazetu HP

6. Závěr

Při čtení tohoto článku jste se mohli přesvědčit, že nasazení \LaTeX u v profesionální praxi, jakkoliv se zdá být složitým, je zvládnutelné. Vyžaduje dosti velké znalosti, při jejichž získávání nám pracovníci DTP studií, zvyklí na klikací programy, pomohou jen zřídka. Později však budeme omezeni jen hranicemi své invence. Chcete-li také proniknout do světa profesionální sazby kterýmkoliv \TeX ovým formátem, řiďte se arabským příslovím z obrázku 5, který byl vysázen \XeTeX em.

الشروع نصف العمل.

Obrázek 5: Arabské přísloví: „Začátek je polovina práce.“

Reference

- [1] Benadiková A., Mada S., Weinlich S.: *Čárové kódy, automatická identifikace*. Grada 1994, 272 pp., ISBN 80-85623-66-8.
- [2] Brabec S.: *Seriál Grafika v UNIXu*. <http://www.root.cz/serialy/grafika-v-unixu/>
- [3] Freeman T. G.: *Aurora: Colour Separation with PostScript Devices*. <http://mirror.ctan.org/support/aurora/>
- [4] Hrad M., Sojka P.: *Automatizace sazby a skenování formulářů*. Zpravodaj Československého sdružení uživatelů \TeX u, **12** (3–4), 123–139 (2002).
- [5] Chvála F.: *O možnostech pdf \TeX u*. Zpravodaj Československého sdružení uživatelů \TeX u, **15** (1), 2–85 (2005).
- [6] International Color Consortium. <http://www.color.org>
- [7] Olšák P.: <http://mirror.ctan.org/macros/generic/ean.zip>
- [8] Olšák P.: *Čárové kódy EAN v \TeX u*. Zpravodaj Československého sdružení uživatelů \TeX u, **6** (2), 86–96 (1996).

- [9] Olšák P.: *The EAN barcodes by T_EX*. TUGboat 15 (December 1994), No. 4, pp. 459–464. <http://www.tug.org/TUGboat/Articles/tb15-4/tb45olsa.pdf>
- [10] *PostScript language reference manual / Adobe Systems Incorporated*. — 3rd ed. Addison-Wesley, 1999. ISBN 0-201-37922-8. <http://www.adobe.com/products/postscript/pdfs/PLRM.pdf>
- [11] Wagner Z.: *Čárové kódy 2/5 a Code39*. <http://mirror.ctan.org/macros/latex/contrib/makebarcode.zip>
- [12] Wagner Z.: *Čárový kód EAN13 s ISBN*. <http://mirror.ctan.org/macros/latex/contrib/ean13isbn.zip>
- [13] Wagner Z.: *Makra pro L^AT_EX*. <http://icebearsoft.euweb.cz/tex/>
- [14] Wagner Z.: *Ořezové značky*. <http://icebearsoft.euweb.cz/tex/#crop>

Summary: L^AT_EX in the Typographer's Profession

T_EX is known mainly in the academic world and is used for writing technical publications. Many people are aware of the possibility of creating high-quality typesetting with T_EX. However, these days when programs with graphical user interface hiding important information prevail, it is difficult to find instructions on how to prepare with T_EX a file for a phototypesetter or a digital printer. The lecture will demonstrate the methods of using L^AT_EX in practice. A few macro-packages that prepare leaflets and invitation cards will be discussed. Also the typesetting of books including their covers.

Zdeněk Wagner
<http://icebearsoft.euweb.cz>