



# OBSAH

Petr Olšák: Úvodníček . . . . .	45
Vít Zýka: Používáme pdf $\TeX$ III: video a zvuk v prezentaci . . . . .	47
Kamil Anis: $\LaTeX$ – Jak na to? . . . . .	55
David Antoš: Jaký byl EuroBach $\TeX$ 2002 . . . . .	98
Jiří Matyáš: Tutoriály pro $\TeX$ ovské začátečníky . . . . .	101
International $\TeX$ Users Meeting Euro $\TeX$ '2003, ENST de Bretagne, Brest, France . . . . .	103
$\TeX$ BITS & BYTES: A Conference Newsletter . . . . .	105

Toto číslo obsahuje CD-ROM  $\TeX$ Live 7

Zpravodaj Československého sdružení uživatelů  $\TeX$ u je vydáván v tištěné podobě a distribuován zdarma členům sdružení. Po uplynutí dvanácti měsíců od tištěného vydání je poskytován v elektronické podobě (PDF) ve veřejně přístupném archivu dostupném přes <http://www.cstug.cz>.

Své příspěvky do Zpravodaje můžete zasílat v elektronické podobě nejlépe jako jeden archivní soubor (.zip, .arj, .tar.gz). Postupujte podle instrukcí, které najdete na stránce <http://bulletin.cstug.cz>. Pokud nemáte přístup na Internet, můžete zaslat příspěvek na disketě na adresu:

Zdeněk Wagner  
Vínohradská 114  
130 00 Praha 3

Disketu formátujte nejlépe pro DOS, formáty Macintosh 1.44 MB a EXT2 jsou též přijatelné. Nezapomeňte přiložit všechny soubory, které dokument načítá (s výjimkou standardních součástí  $\LaTeX$ u), zejména v případě, kdy vás nelze kontaktovat e-mailem.

ISSN 1211-6661 (tištěná verze)  
ISSN 1213-8185 (online verze)

Vážení čtenáři,

v době, kdy píšu tyto řádky, řádí venku Medard se svým typicky deštivým počasím. Není ovšem zcela jasné, kdy se Vám dostane do rukou druhý časopis letošního roku. Zásoba příspěvků je, bohužel, až na jeden případ o LyXu, prázdná. Pokud tedy máte dojem, že není dobře, aby časopis přestal existovat, budete pro to asi muset také něco udělat. Například napsat příspěvek.

Předpokládám, že první číslo Zpravodaje dostali už všichni členové našeho sdružení a že jedinou výjimkou je předseda, který do schránky stále časopis nedostal. Pokud nedostali časopis další členové, znamenalo by to, že nám kromě neexistence příspěvků začíná skřípat administrativa. Zatím si ovšem nikdo nestěžoval. A pokud nemá časopis předseda, je to jeho problém a neměl by jej zatahovat sem do tohoto úvodníčku.

Když už je řeč o administrativě, pak před několika málo dny, 6. června, došlo k významné změně. Podle předchozí dohody předal Ondřej Vácha administrativní záležitosti CSTUGu do Prahy. Patří mu na tomto místě právem poděkování za dlouholetou službu CSTUGu, třebaže občas některé věci nešly hned tak, jak by měly (jako třeba naposled ten nedoručený časopis pro Olšáka). Administrativu převzala paní Helena Holovská z MÚ ČAV. Přeji ji co nejméně problémů a pevné nervy při této práci. A taky ji samozřejmě moc děkuji, že se nabídla tuto práci vůbec dělat.

29. května ukončil Sebastian Rahtz verzi 7 každoročně připravované distribuce  $\text{T}_{\text{E}}\text{X}$ u na CDROM nazvanou  $\text{T}_{\text{E}}\text{X}$ Live. Od této chvíle si ji začaly stahovat jednotlivé národní skupiny uživatelů  $\text{T}_{\text{E}}\text{X}$ u (LUG) s tím, že vytvoří matici a začnou ji lisovat pro své členy. Bohužel, v CSTUGu asi nejsou lidi, a proto se do této verze CSTUGu nepodařilo zapracovat ani aktualizovanou verzi dokumentace a nevím o ničem novém, čím by CSTUG do této verze přispěl. Výjimkou je snad jenom drobnost – program vlna v nové verzi. Pan kolega Demel mě totiž upozornil na chybu a požádal, ať do programu zapracuji novou vlastnost. Podivil jsem se, že tento program v době inteligentních textových editorů vybavených jistě mocnými nástroji pracujícími s regulární výrazy, vůbec někdo používá. Nicméně jsem program opravil a upravil a do  $\text{T}_{\text{E}}\text{X}$ Live i na svůj ftp archív jsem zařadil novou verzi. Naposledy jsem na zdrojáky toho programu sáhl před sedmi lety, ale přesto jsem dnes nemusel složitě luštit, co jsem to tenkrát vlastně myslel. Bylo to totiž napsáno v CWEBu.

Koncem května jsem upozornil na diskusní skupině T<sub>E</sub>X-implementors, že pokud se například v MikT<sub>E</sub>Xu nebo T<sub>E</sub>Xlive spustí binárka `tex` s parametrem `--src`, pak tato binárka o sobě tvrdí, že je T<sub>E</sub>X, ale přitom to není pravda. Parametr způsobí, že na úrovni expand processoru se začnou líhnout ve vnitřnostech programu navíc specialy, které sice informují dvi ovladač o tom, kterému řádku zdrojového souboru dané místo v dvi odpovídá (což může být užitečné), ale na druhé straně zcela staví na hlavu běžné programátorské praktiky s `\everypar`. Makro spouštěné z `\everypar` s jedním neseparovaným parametrem prostě nabere do tohoto parametru token `\special` a nikoli první písmeno odstavce. Když jsem se to dozvěděl, důrazně jsem upozornil, že je ohrožena licence, kterou pro T<sub>E</sub>X stanovil autor Donald Knuth. Jsem přesvědčen, že binárka `tex`, ve které se líhnou navíc takové specialy, není T<sub>E</sub>X. Přiznávám, že jsem na vlastnost `--src` nepřišel sám, ale upozornili na to uživatelé našeho diskusního listu `cstex@cs.felk.cvut.cz`.

Netušil jsem, že upozornění na problém s licencí T<sub>E</sub>Xu v souvislosti s přepínačem `--src` způsobí u autorů T<sub>E</sub>Xlive nebývalý rozruch a vyprovokuje je to k okamžité práci. Během hodiny stáhli (dosud definitivní) verzi obrazu svého CDčka a začali upravovat zdroják T<sub>E</sub>Xu tak, aby při použití parametru `--src` o sobě neříkal, že je to T<sub>E</sub>X. Pak se jali celou distribuci znovu překládat pro všechny platformy. Jednotlivé LUG to už málem začaly dávat na matrice a stáhli to na poslední chvíli.

Je vidět, že konzervativní licence T<sub>E</sub>Xu, která významnou měrou dělá T<sub>E</sub>X T<sub>E</sub>Xem, je pro většinu tvůrců T<sub>E</sub>Xového software natolik posvátná, že jsou ochotni kvůli ní i něco obětovat. A to je dobré znamení.

Jistě se ptáte, jak a kdy najdeme CDčko T<sub>E</sub>Xlive 7 jako přílohu do našeho časopisu. Na poslední chvíli jsme odvolali objednávku 700 kusů CDček od Němců a náš výbor se rozhodl, že se pokusí najít levnější cestu k lisování přímo u nás. Držím výboru palce.

... Tento úvodník jsem psal v červnu 2002. Od té doby uplynulo hodně času a také hodně vody v českých řekách. Právě povodně nepřímou způsobily, že časopis dostáváte s takovým zpožděním. V časopise je vloženo čerstvé Cédéčko TeXlive7, které jsme s aktuálními změnami nechali lisovat pro členy našeho sdružení v listopadu 2002. Přeji Vám, ať to Cédéčko dobře slouží.

(SLT 2002, 10. 11. 2002)



*V tomto článku si ukážeme příklady, jak pdf $\TeX$ em vytvářet dokumenty se zvukovou nahrávkou nebo videem.*

## Audio a video v PDF

Jedna z výhod elektronické publikace je, že nemusí obsahovat pouze statické prvky (text a obrázky), ale i prvky dynamické (video) a takové, přinášející jinou, než vizuální informaci (zvuk). Chceme-li například přiblížit zpěv slavíka, je mnohem efektivnější připojit zvukovou ukázkou, než jej popisovat slovy. Podobně, budeme-li chtít rozebrat herní algoritmy robotů v robotickém fotbale, bude užitečné nejprve promítnout videozáznam z vlastního zápasu. O užitečnosti začlenění audia a videa do dokumentů zvláště učených pro pedagogické a prezentační účely asi není pochyb.

Na začátek však musím čtenáře varovat. Abychom si mohli zvukovou nebo video ukázkou v PDF prohlídnout, musí být náš počítač náležitě hardwarově i softwarově vybaven. S hardwarem již dnes moc problémy nejsou, ale s programy je potíže. Ačkoliv PDF formát specifikuje odkaz na audio i video již od svých prvních verzí, málokterý prohlížeč je umí přehrát. Většina tyto odkazy ignoruje. Podle mých informací umí se zvukem a videem pracovat Acrobat (Reader) pro Windows a Mac; pro Linux zatím žádný takový program není.<sup>1</sup> Tento dokument byl testován v Acrobatu 4.0 a Readeru 5.0 CE pro Windows 98.

V minulém díle [3] jsme se věnovali vytvářením odkazů, záložek a náhledů. Tyto hypertextové objekty v žargonu PDF patří mezi *anotace*. Zvláštním typem anotace je i objekt obsahující odkaz na audio nebo video soubor. Tento soubor můžeme ponechat vně dokumentu nebo jej vložit „tak jak je“ dovnitř PDF. Přípravu takového souboru musíme svěřit externímu programu; pdf $\TeX$  zvukům ani videu nerozumí.

Tento text není úplnou referenční příručkou, spíše přehledkou základních postupů uložení, přehrávání a vizualizace zvukové a video anotace v PDF formátu. Věřím, že s pomocí PDF referenčního manuálu [1] si čtenář další varianty snadno odvodí.

---

<sup>1</sup>Existují dvě další možnosti, jak automaticky spustit video: a) ve webovém prohlížeči přes externí URL (funguje spolehlivě, jen nám prohlížeč ukončí prezentační mód a překryje PDF svým oknem), b) v jiném externím programu přehrávajícím video (spouštění cizích aplikací z PDF je však podobně systémově závislé jako spouštění videa interně, takže tudy cesta k přenositelnosti určitě nevede).

## Základní syntaxe PDF

Protože se nevyhneme zápisu některých parametrů přímo v kódu PDF, seznámíme se nejprve se syntaxí jeho datových struktur. Základním stavebním kamenem PDF je *objekt* (object). Každý objekt má své identifikační a modifikační číslo. Například

```
1 3 0 obj <<
2   /Producer (pdfTeX-1.0)
3 >> endobj
```

je objekt s identifikačním číslem 3 a modifikačním číslem 0. Všechny objekty, které vytvoří pdfTeX mají nulové modifikační číslo, protože pdfTeX dokumenty vytváří, ale nemodifikuje. Pokud byste tento objekt pozměnili, např. v Acrobatu, pak jeho původní varianta v dokumentu zůstane a na konec se přidá pozměněná verze se stejným identifikačním číslem 3, ale modifikačním číslem 1.

Mezi dvojicí slov `obj` a `endobj` je slovník (dictionary). Poznáme jej podle páru otevíracích `<<` a zavíracích `>>` závorek. Uvnitř slovníku jsou dvojice *klíč-hodnota*. Předdefinovaná jména klíčů či parametrů začínají lomítkem. V našem případě máme ve slovníku jednu dvojici klíč-hodnota, klíčem je slovo `Producer` a hodnotou `pdfTeX-1.0`.

Každý klíč dle kontextu očekává hodnotu určitého typu. Zde je seznam datových typů a jejich příklady:

- boolean (`true`, `false`),
- celé nebo reálné číslo (`2`, `-3.35`, `.03`),
- řetězec (`(Text s\n druhym radkem)`),
- hexadecimální číslo (`<781BD3>`),
- jméno (`/Movie`, `/Producer`),
- nehomogenní pole (`[/Win false (pdfTeX) 1]`),
- slovník (`<< /Subtype /Movie /Volume 1 >>`).

*Proud* je množina bytů s předřazeným slovníkem, který definuje její délku v bytech, způsob kompresního kódování ap. Typicky je v proudu text sazby s posuny referenčního bodu:

```
5 0 << /Length 85 >>
stream
1 0 0 1 91.925 759.927 cm
BT
/F1 9.963 Tf 0 0 Td[(Si)1(m)-1(p)1(1e)-334(text.)]TJ
ET
endstream
endobj
```

Obsah proudu se komprimuje podle nastavené úrovně `\pdfcompresslevel`.

## Anotace

Portable Document Format řadí mezi *anotace* následující objekty:

- odkaz (`/Link`),
- video, zvuk (`/Movie`, `/Sound`),
- soubor jako přílohu (`/FileAttachment`),
- text v poznámkovém okýnku i mimo něj (`/Text`, `/FreeText`),
- úsečku nebo šipku, obdelník, elipsu (`/Line`, `/Square`, `/Circle`),
- zvýraznění, podtržení a přeškrtnutí s poznámkou (`/Highlight`, `/Underline`, `/Squiggly`, `/StrikeOut`),
- razítko s poznámkou, čáru od ruky s poznámkou (`/Stamp`, `/Ink`),
- vzhled interaktivních objektů (`/Widget`),
- ořezové značky a značky pro barevné sesazení (`/PrinterMark`, `/TrapNet`).

Některé z těchto objektů jsou v pdfTeXu nepříliš užitečné. Zvýraznění, šipky a poznámky může využít korektor nebo školitel, sedí-li zrovna na opačné straně planety a potřebuje vám předat připomínky k vašemu textu. V takovém případě ale použije interaktivní Acrobat, pdfTeX sotva, to raději bude přímo editovat zdroják. Video a zvuk však patří mezi ty anotace, pro které nemáme ve standardním TeXu nástroje, a jejich vkládání pdfTeXem není o moc těžší než vkládání obrázků. A právě na tyto dva typy se zaměříme v tomto článku. Ostatní typy anotací lze vložit analogicky s pomocí referenčního manuálu [1, str. 489].

Pro vkládání anotací má pdfTeX primitiv  
`\pdfannot [rule spec] general text`

Ve složených závorkách zapsaný `<general text>` prochází expanzí a definuje parametry anotace. Obdelníková oblast, kterou anotace zaujímá, je určena pomocí `<rule spec>`. Poloha obdelníku je dána aktuální pozicí sazby, ale definovaná šířka, výška a hloubka anotace s touto aktuální pozicí nepohnou! Napíšeme-li

```
4 \pdfannot width2cm height2ex depth.5ex{...}Nápis
```

bude text „Nápis“ uvnitř anotace. Pokud bychom chtěli mít „Nápis“ až za anotací musíme bod sazby posunout, např.

```
5 \pdfannot width2cm height2ex depth.5ex{...}\hskip2cm Nápis
```

## Video anotace

*Note: The format of a self-describing movie file is left unspecified, and there is no guarantee of portability.* PDF Reference Manual [1]

Video anotace slouží k přehrání audio i video samoobsažného souboru. O tom, jaký přesně samoobsažný soubor Acrobat Movie Plug-In dokáže přehrát se ne-

dočteme ani v jeho dokumentaci. Fungoval mi audio wav a video mov s kodekem Cinepak Radius. Musíme se zatím smířit s tím, že video vkládáme a testujeme jen pro naši konkrétní verzi Acrobatu (Readru).

Vložme tedy externí zvukový soubor jako video anotaci (/Subtype /Movie)

```
6 \pdfannot width 11mm height 2ex depth .5ex {%
7 /Subtype /Movie
Slavík 8 /Movie << /F (birds/slavik.wav) >>}
9 Slavík
```

Řetězec klíče /F určuje název a cestu audio/video souboru. Oddělovačem adresářů je lomítko. Absolutní cesta začíná lomítkem, název disku píšeme bez dvojtečky. Například

```
10 \pdfannot width 15mm height 2ex depth .5ex {%
11 /Subtype /Movie
Káně 12 /Movie << /F (/d/buletin/anotace/birds/kane.wav) >>}
13 Káně
```

Přesunutím dokumentu na počítač s jinou adresářovou strukturou způsobí, že prohlížeč soubor s absolutní cestou nenajde. V takovém případě nabídne dialog s výzvou, aby jste soubor vyhledali ručně. To jste si zřejmě nyní vyzkoušeli. Další možností je zadat URL. Pak můžeme s dokumentem cestovat kam chceme, jen tam musí být funkční Internet. Pro URL musíme zavést samostatný objekt:

```
14 \pdfobj{<< /Type /FileSpec /FS /URL
15 /F (file:///d/buletin/anotace/birds/datel.wav) >>}
16 \pdfannot width 15mm height 2ex depth .5ex {%
Datel 17 /Subtype /Movie
18 /Movie << /F \the\pdflastobj\space 0 R >>}
19 Datel
```

Velikost boxu anotace určuje aktivní plochu pro spuštění zvuku kliknutím. Názorně to ukazuje anotace s rámečkem (/Border) a titulkem /T. Titulek se objeví při zapnutých Preferences→Annotation→Auto open other window...

```
20 \pdfannot width 15mm height 7mm depth 3mm {%
Dudek 21 /Subtype /Movie /Border [0 0 1] /T (Dudek)
22 /Movie << /F (birds/dudek.wav) >>} Dudek
```

Plocha video anotace může být vyplněna statickým obrázkem prvního video záběru nastavením parametru /Poster. I když to je bláhové, chtít po našich zvukových ukázkách první záběr, z didaktických důvodů si také ukážeme příklad:

```
23 \pdfannot width 15mm height 7mm depth 3mm {%
24 /Subtype /Movie /Border [0 0 1]
Sýkorka 25 /Movie << /F (birds/sykorka.wav) /Poster true >>}
26 Sýkorka
```

Pro zvuk je výhodnější nastavit /Poster jako obrázek. Obrázek načteme stan-



dardním způsobem pomocí `\pdfximage` a její referenci předáme parametru `/Poster`:

```
27 \immediate\pdfximage width 15mm height 18mm depth 0mm
28     {birdsfig/kos.jpg}
29 \pdfannot width 15mm height 18mm depth 0mm {%
30   /Subtype /Movie /Border [0 0 1]
31   /Movie << /F (birds/kos.wav)
Kos 32     /Poster \the\pdflastximage\space 0 R >>}
33 \par Kos
```

Video anotace může být doplněna řídicí lištou (`/ShowControls`) umožňující nastavit konkrétní místo přehrávání a video spustit a zastavit.

```
34 \pdfannot width 15mm height 18mm depth 0mm {%
35   /Subtype /Movie /T (Drozd)
36   /A << /ShowControls true /Mode /Once >>
37   /Movie << /F (birds/drozd.wav)
Drozd 38     /Poster \the\pdflastximage\space 0 R >>}
39 \par Drozd
```

Změnou parametru `/Mode` ovlivníme způsob přehrávání souboru (`/Palindrome` funguje jen pro formáty umožňující zpětné přehrávání).

```
/Mode /Once      /Mode /Open      /Mode /Repeat  /Mode /Palindrome
```

Kukačka

Bažant

Křepelka

Pěnkava

Někdy si přejeme, aby se anotace spustila automaticky s nějakou akcí, například s otevřením stránky. Musíme vytvořit objekt typu akce podtypu video (`/S /Movie`), přes titul `/T` ji svázat s danou anotací a vytvořit odkaz ve slovníku `/AA` atributů příslušné stránky.

```

40 \pdfannot width 15mm height 18mm depth 0mm {%
41 /Subtype /Movie /T (AutoMovie) /F 1
42 /Movie << /F (birds/racek.wav)
43 /Poster \the\pdflastximage\space 0 R >>}
44 \par Racek
45 \immediate\pdfobj
46 {<< /S /Movie /T (AutoMovie) /Operation /Play >>}
Racek 47 \edef\pdfpageattrbody{%
48 /AA << /O \the\pdflastobj\space 0 R >>}
49 \def\pdfpageattrbegin{\pdfpageattr\bgroup}
50 \expandafter\pdfpageattrbegin\pdfpageattrbody}

```

Cvičení na posledních čtyřech řádcích je nutné proto, že primitiv `\pdfpageattr` neexpanduje svůj parametr. My však potřebujeme získat referenci na poslední objekt z registru `\pdflastobj`.

## Zvuk

Zvuk můžeme také vložit zvukovou anotací (`/Sound`)<sup>2</sup>. Protože tato anotace vyžaduje surová audio data, je třeba nastavit několik parametrů.



```

51 \immediate\pdfobj stream
52 attr{/Type /Sound /R 22050 /C 1 /B 8 /E /muLow
53 /F (birds/straka.raw)}{ }
54 \pdfannot width 2em height 2em depth 0mm {%
55 /Subtype /Sound /Sound \the\pdflastobj\space 0 R }

```

Straka

Parametr `/R` značí vzorkovací frekvenci, `/C` počet kanálů (mono=1, stereo=2) a `/B` počet bitů na vzorek. Zvolíme-li šestnáctibitový formát (`/B 16`), je třeba pomocí parametru `/E` nastavit pořadí významnosti bytů.

Zvuková data můžeme přímo vložit do dokumentu v podobě datového proudu. Nemusíme se pak starat o externí soubor.



```

56 \immediate\pdfobj stream
57 attr{/Type /Sound /R 22050 /C 1 /B 8 }
58 file{birds/spacek.raw}
59 \pdfannot width 2em height 2em depth 0mm {%
60 /Subtype /Sound /Contents (Interní zvuk)
Špaček 61 /Sound \the\pdflastobj\space 0 R }

```

Chceme-li změnit ikonu zvukové anotace, použijeme kód nastavující vzhledový slovník `/AP`.

<sup>2</sup>Tento druh anotace se mi nepodařilo přehrát v Acrobat Readeru.

```

62 \immediate\pdfximage width 15mm height 18mm
63     {birdsfig/vrabec.jpg}%
64 \setbox0=\hbox{\pdfrefximage\pdflastximage}%
65 \immediate\pdfxform 0
66 \immediate\pdfobj stream
67   attr{/Type /Sound /R 22050 /C 1 /B 8 }
68   file{birds/vrabec.raw}
Vrabec 69 \pdfannot width 15mm height 18mm depth 0mm {%
70   /Subtype /Sound /Contents (Ikona)
71   /Sound \the\pdflastobj\space 0 R
72   /AP << /N \the\pdflastxform\space 0 R >> }
73 \par Vrabec%

```

Ikonu však zřejmě nelze změnit v době přehrávání ukázky. Protože tento vzhledový slovník požaduje typ form xobject, můžeme místo ikony vložit i sazbu  $\TeX$ u pomocí `\pdfxform`.

```

74 \setbox0=\hbox{Čáp a  $\pi$ }
75 \newdimen\wdz \newdimen\htz \newdimen\dpz
76 \wdz=\wd0 \htz=\ht0 \dpz=\dp0
77 \immediate\pdfxform 0
78 \immediate\pdfobj stream
79   attr{/Type /Sound /R 22050 /C 1 /B 8 }
80   file{birds/cap.raw}
81 \pdfannot width \wdz height \htz depth \dpz {%
82   /Subtype /Sound /Contents (TeX box)
83   /Sound \the\pdflastobj\space 0 R
84   /AP << /N \the\pdflastxform\space 0 R >> }

```

## Hotové řešení

Pro ty, kteří hledají hotové řešení nabízím následující makro.

```
\putmovie [(rule spec)]{(img file)}{(movie file)}
```

V místě jeho použití vytvoří video anotaci o velikosti `rule spec`. Uvedeme-li `img file`, zaujme plochu anotace specifikovaný obrázek, jinak se získá z prvního záběru videa. Obrázek musí jít načíst primitivem `\pdfximage`. Předefinováním makra `\attrmovie` můžeme změnit parametry video anotace. Přednastavené chování neukazuje řídicí listu ale nakreslí okolo anotace rámeček. Mód přehrávání (přednastavený `/Once`) je v makru `\modemovie`. Zde jsou definice maker:

```

85 \def\putmovie#1#{\putmoviee{#1}}
86 \def\putmoviee#1#2#3#{\edef\tmp{#2}%

```

```

87 \ifx\tmp\empty\def\tmpattrmovie{/Poster true }%
88 \else%
89   \immediate\pdfximage#1{#2}%
90   \def\tmpattrmovie{/Poster \the\pdflastximage\space 0 R }%
91 \fi%
92 \pdfannot#1{/Subtype /Movie /Title (#3) \attrmovie
93   /Movie << /F (#3) \tmpattrmovie \modemovie >>}%
94 \vrule#1 width0pt \vrule#1 height0pt depth0pt}}
95
96 \def\attrmovie{/Border [0 0 1]
97   /A << /ShowControls false /Mode /Once >> }
98 \def\modemovie{/Mode /Once }

```

a zde příklad jejich použití

```

99 \putmovie width15mm height18mm
100   {birdsfig/havran.jpg}{birds/havran.wav} a
101 \putmovie width5mm height6ex{}{birds/vrana.wav}
a 102 \par Havran a vrána

```

Havran a vrána

## Errata

V minulém díle jsem omylem uvedl dnes již neplatný primitiv `\pdfannotlink` místo nového `\pdfstartlink`. Prosím, opravte si syntaxi uvedenou na posledním řádku strany 15 ve Zpravodaji 1/2002. Čtenářům se omlouvám. Správný tvar má být:

```
\pdfstartlink [⟨rule spec⟩] [attr{⟨attributes⟩}] ⟨action spec⟩
```

## Reference

- [1] Adobe systems Incorporated. *PDF reference manual, v. 1.4*, second edition, 2000. <http://partners.adobe.com/asn/developer/acrosdk/DOCS/PDFRef.pdf>.
- [2] Hàn Thê Thành, Sebastian Rahtz, and Hans Hagen. *The pdfTEX user manual*. Pragma, 2001. <http://www.tug.org/applications/pdftex/pdftex-s.pdf>.
- [3] Vít Zýka. Používáme pdfTeX II: prezentace fotografií aneb jak na hypertext *Zpravodaj Československého sdružení uživatelů T<sub>E</sub>Xu*, 12(1):13–21, 2002.

## Summary: Using pdf<sub>T</sub>E<sub>X</sub> III: movie and sound for presentation

This article shows incorporating movie and sound annotations into document by pdf<sub>T</sub>E<sub>X</sub>. The text follows the last issue article *Using pdf<sub>T</sub>E<sub>X</sub>II: document for photographs presentation; how to create a hypertext*, CSTUG Zpravodaj 1/2002.

Vít Zýka, zyka@cmp.felk.cvut.cz

*Redakční poznámka: Článek o vkládání videa a zvuku do prezentací je psán v pdf<sub>T</sub>E<sub>X</sub>u a všechny popisované techniky ilustruje. Papír však ještě není tak živý, aby dokázal interagovat se čtenářem a přenášet zvuk či video. Navíc se ukázalo, že některé ikony a obrázky, určené k ovládní zvuku a videa, jsou sice na obrazovce viditelné, ale Acrobat je nevytiskne a ponechá jen bílé místo. Abyste se mohli podívat, jak bylo vše autorem zamýšleno, a mohli jste si při čtení článku přehrát všechny zvuky, je tento článek současně s tištěnou verzí uvolněn i ve formátu PDF. Najdete jej na <http://bulletin.cstug.cz/bul20022.shtml>*

---

---

## Ly<sub>X</sub> – Jak na to?

---

KAMIL ANIS

### Prolog

Obliba Ly<sub>X</sub>u<sup>1</sup> neustále roste. Šíře jeho možností, dostupnost a snadné ovládní činí z tohoto programu velmi účinný nástroj na tvorbu dokumentů rozličného charakteru. S tím, jak se obec uživatelů Ly<sub>X</sub>u stále rozrůstala, přirozeně došlo ke zvýšení frekvence (neustále se opakujících) dotazů, které se Ly<sub>X</sub>u přímo či nepřímo dotýkají. Proto jsem se rozhodl napsat tento článek, v dobré víře, abych uživateli práci s programem co nejvíce ulehčil. V lepším případě bych si přál, aby byl čtenář po takto nabytých zkušenostech schopen většinu problémů řešit svépomocí.

Osobně se domnívám, že aktuální verze Ly<sub>X</sub>u je pro uživatele, který nechce nic jiného, než snadno a rychle vytvořit konvenční dokument nesoucí určité známky profesionality, zcela postačující. O sazbě, nebo dokonce o mechanismech odehrávajících se za obrazovkou jeho monitoru toho v podstatě mnoho

---

<sup>1</sup>Čti [lyksu].

vědět nemusí. Uživatel, který však od sázecího systému očekává o něco více, musí ale počítat s tím, že bude muset vynaložit určité úsilí. Nemýlím-li se, tak pokus o vytvoření víceúčelového programu na sazbu textu, který by uměl ovládat člověk s velmi nízkou počítačovou gramotností, vždy skončil neúspěchem.

Pokud jste přešli k sofistikovanějším sázecím systémům právě proto, že konvenční textové procesory nebyly sto uspokojit vaše požadavky, nemělo by vás toto úsilí od práce s novým systémem odradit. Profesionální výsledek budiž odměnou za dobře a pečlivě vykonanou práci. Zatímco většina lidí se spokojí s průměrným výsledkem, existují i tací, kteří razí teorii „Na detailech záleží!“. Osobně jsem se vždy počítal k těm druhým. Precizní sazba s propracovanými detaily však zpravidla klade extrémně vysoké nároky na objem vynaložené práce. Proto drtivá většina dotazů související se sazbou dokumentů není pouze ve smyslu „Jak udělat to či ono?“, ale „Jak se u toho co nejméně zapotit?“.

Dále bych chtěl na tomto místě předeslat, že tento text nemá suplovat uživatelský manuál. Domnívám se, že dokumentace k LyXu je zpracována velmi kvalitně a její nastudování hravě zvládne i člověk, jehož znalost angličtiny nelze označit známkou *excelentní*. Za nejdůležitější ale považuji větu, že autoři helpu nepředpokládají, že čtenář je hlupák. V tomto textu se krom toho dále předpokládá, že uživatel je alepoň částečně seznámen s pravidly hladké sazby a základními typografickými poučkami a že má zcela jasnou představu o tom, čeho má být v konečném výsledku dosaženo. Proto se zde zmíním převážně o věcech, které v oficiální dokumentaci k LyXu nejsou, nebo z ní na první pohled nevyplývají.

Připouštím, že nadpis tohoto článku může u řady čtenářů evokovat představu onoho známého „howto“, ale zároveň bych chtěl zdůraznit, že se nejedná o kuchařku s jednořádkovým komentářem. Jednak proto, že to často ani není ze samé podstaty věci možné, ale také proto, že sazba textu je poměrně náročný proces, který nelze odbýt dvěma řádky v nastavení sázecího programu. Knihy zasvěcené typografii zpravidla pojednávají o určitých konvencích, kterými se sazba tiskovin řídí, ale jak toho docílit konkrétními prostředky, se z nich čtenář bohužel nedozví. Proto jsem se snažil text koncipovat tak, aby konkrétnímu úkonu nejprve předcházela diskuse ve smyslu typografického pravidla, podobně jako je tomu např. v [7].<sup>2</sup> Na konci se zmíním o hrstce rad a triků, které jsou výsledkem dlouhodobějších zkušeností a které mohou uživatelům LyXu ušetřit spoustu starostí. Věřím, že tento článek bude užitečným pomocníkem laikům i pokročilejším uživatelům. Pokročilejší uživatelé nechť laskavě přeskóčí následující sekci.

---

<sup>2</sup>Osobně mám k této publikaci jisté výhrady, ale je to pravděpodobně první počín svého druhu.

## Co je to L<sub>Y</sub>X?

Nejprve bych chtěl zdůraznit, že náplní této sekce není popis L<sub>Y</sub>Xu, jakožto programu samotného. Jedná se spíše o jakousi „berličku“, která by neznalému čtenáři měla pomoci vytvořit si velmi hrubou představu o tom, co se pod oněmi třemi písmeny vlastně skrývá. Místo toho, abych se snažil emulovat něco, co už výborně udělal někdo jiný [1] (sekce What is L<sub>Y</sub>X? a L<sub>Y</sub>X is What?!), chtěl bych L<sub>Y</sub>X představit jakožto velmi silný a účinný nástroj, který může být v mnoha ohledech zajímavý pro obě potencionální skupiny uživatelů. V prvé řadě jsou to ti, kteří o L<sup>A</sup>T<sub>E</sub>Xu nemají ani ponětí, ale především i ti, kteří s L<sup>A</sup>T<sub>E</sub>Xem už dlouhou dobu pracují a domnívají se, že na způsobu sazby, který používali až doposud, se toho moc vylepšit nedá.

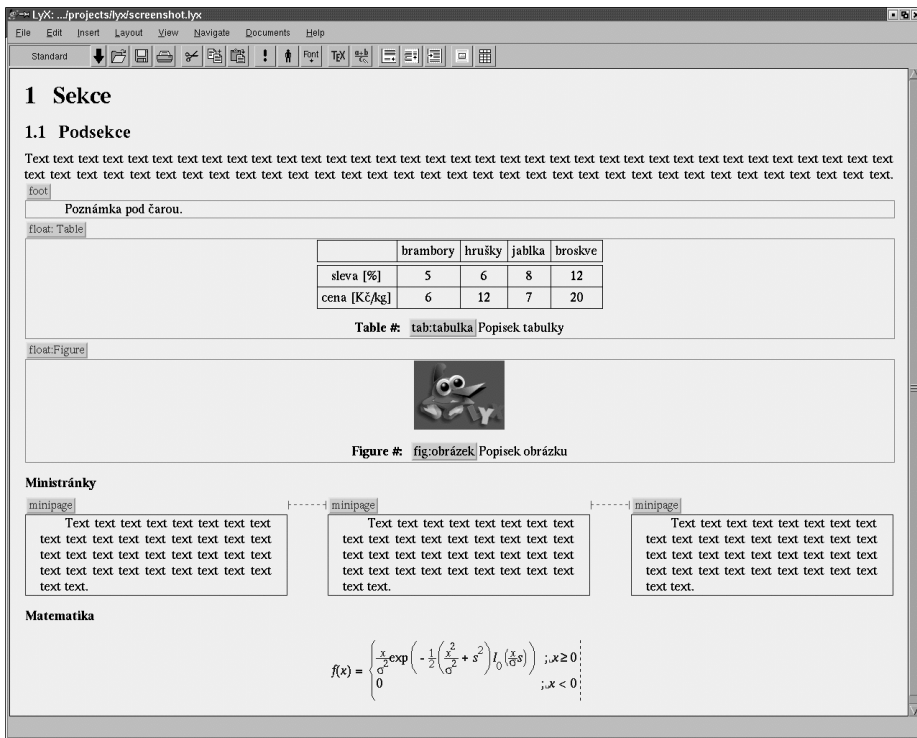
Co se týče vizuální stránky věci, nemám v úmyslu zdvojnásobit objem tohoto článku nespočetným množstvím screenshotů. Světlou výjimku tvoří obr. 1. Po dlouhém přemýšlení, jsem nakonec dospěl k názoru, že pro neznalého čtenáře (kterému je ostatně tato sekce určena) patrně platí známá poučka „jeden obrázek je lepší, než tisíc slov“. Každopádně chcete-li se s L<sub>Y</sub>Xem seznámit (aniž byste si ho instalovali), pak Vám vřele doporučuji nejprve navštívit jeho domovské stránky [1]. Věřím, že po shlédnutí tutoriálu «L<sub>Y</sub>X Graphical Tour» začnete velmi vážně uvažovat o jeho instalaci. Pokud jste s L<sub>Y</sub>Xem nikdy předtím nepochovali, pak by se Váš první soubor (otevřený v nově nainstalovaném L<sub>Y</sub>Xu) měl jmenovat `tutorial.lyx`.<sup>3</sup>

Ačkoli může prvních pár řádek na někoho působit trochu komicky, rozhodně tento text nepodceňujte! Jeho přečtení až do konce se Vám opravdu vyplatí. Text je psán (podle mého soudu) velmi neformální a zábavnou formou. Pro ty, kterým by snad úroveň tamní angličtiny připadala neúnosná, existuje i český překlad, který lze stáhnout např. ze stránek Ludka Bruknera [2]. Česká verze není úplně totožná s poslední anglickou verzí, ale jako úvodní seznámení (včetně demonstrací) bohatě postačí.

A nyní již k nadpisu této sekce. Myslím, že díky mnohostrannosti L<sub>Y</sub>Xu neexistuje na tuto otázku jednoznačná odpověď. Jedna z verzí by mohla znít asi takto: L<sub>Y</sub>X je vizuální (interaktivní) nadstavba nad sazební systém L<sup>A</sup>T<sub>E</sub>X. Interaktivní textové procesory bývají často označovány zkratkou WYSIWYG (What You See Is What You Get). Překlad této magické formule zcela ponechávám čtenáři. Já sám jsem se setkal s několika interpretacemi tohoto „sloganu“ a musím říci, že některé z nich byly vskutku zábavné. Naproti tomu L<sub>Y</sub>Xu dali jeho vývojáři do vínku nálepku WYSIWYM (What You See Is What You Mean). Toto úsloví v sobě skrývá cosi o tom, že část břemene bude svalena na počítač. Někteří uživatelé si však tento výrok mylně vysvětlují tak, že za ně bude počítač i přemýšlet.

---

<sup>3</sup>Soubor je standardní součástí instalace a dá se otevřít z menu Help/Tutorial.



Obr. 1: Pracovní plocha LyXu

Podstatou tohoto přístupu je ale „pouhý“ fakt, že autor by se měl soustředit výhradně na obsah svého textu a neměl by si lámat hlavu s tím, jak bude vlastně dokument ve své výsledné podobě vypadat. V jistém smyslu tedy opět dochází k dělbě práce, ovšem tentokrát je sazeč nahrazen počítačem. Asi tušíte, že pro tvůrce LyXu to nebyla úloha nijak snadná. Ačkoli je největší devizou LyXu tvorba textových dokumentů, za povšimnutí stojí to, že v souvislosti s LyXem se nikde nemluví o textovém procesoru, poněvadž tento nástroj není určen pouze ke kompozici textů, nýbrž obecně dokumentů založených na principu strukturovaného meta-jazyka (typu SG/X/HT/ML) tzv. (*structured authoring*).<sup>4</sup>

V zásadě nejhrubší zrno, podle kterého lze systémy určené k manipulaci s textem dělit, je na tzv. *interaktivní* a *dávkové* – význam obou výrazů je snad samovysvětlující. Naproti tomu diskuse, zdali takový program označovat za *tex-*

<sup>4</sup>Přestože LyX umožňuje export do formátu HTML (a dají se v něm tvořit i dokumenty SGML), následující text se bude týkat výhradně tvorby textových dokumentů.



*tovej processor* či *sázezí systém*, mnohdy neberou konce. Z podstaty věci je však toto označení zcela bezvýznamné (na funkci programu to nemá žádný vliv). Oba dva druhy však mají v počítačové typografii své místo.

Za zástupce interaktivních textových procesorů bývá zpravidla označován program MS Word, ovšem do stejné kategorie spadají i programy určené pro profesionální sazbu jako InDesign, QuarkXPress, PageMaker a další. Naproti tomu typickým zástupcem systémů pro dávkovou sazbu je dnes důvěrně známý systém  $\text{T}_{\text{E}}\text{X}$  (resp. jeho nadstavba  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ ). Ačkoli bývá  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  velmi často označován za zástupce interaktivních textových procesorů, při bližším ohledání snadno zjistíte, že se vlastně jedná o jakýsi hybrid, který nelze jednoznačně „zaškatulovat“. Právě tato jedinečnost činí  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  velmi atraktivním.

Obliba interaktivních sázezí systémů vzrostla převážně v poslední době, kdy malá DTP studia začala vyrůstat jako „houby po dešti“. Ačkoli dávkové sázezí systémy oplývají nesčítanou řadou možností (a to dokonce i takových, o kterých by se uživatelům komerčních programů mohlo jen zdát), pracovníci DTP studií mnohdy lamentují tím, že typograf potřebuje sednout a tvořit, namísto učení se programovacím technikám a že jeho myšlenky musí být zhmotněny na obrazovce monitoru bezprostředně pro provedení změn. Setkal jsem se i s názorem, že dávkové sázezí systémy typu  $\text{T}_{\text{E}}\text{X}$  zůstaly dnes atraktivní pouze pro vybranou skupinku „nadšenců“, zejména z řad akademických a univerzitních pracovníků. Osobně se domnívám, že každý z uvedených programů má něco do sebe a že jednomyslné zavržení dávkových systémů je přinejmenším krátkozraké, poněvadž to, co se na počátku zdá býti jako příliš složité, zákonitě vyústí v obrovskou efektivnost a úsporu práce. Ostatně složitý systém, který přináší složitá řešení, je automaticky předurčen k zániku.

Zde se tedy nabízí otázka, zdali by bylo možno výhody obou typů sloučit do jednoho a zároveň tak odstranit nedostatky obou. Asi cítíte, že takto formulovaný požadavek je z čistě praktického hlediska poněkud naivní, avšak jistá skupinka lidí, v čele s duchovním otcem  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u Matthiasem Ettrichem, se podobné myšlenky nemohla zbavit, a tak byl na světě  $\text{K}_{\text{L}}\text{T}_{\text{E}}\text{X}$ . Do jaké míry se tento záměr podařil, se nejlépe můžete přesvědčit sami. Já se pokusím o částečný výčet bodů, které by však neměly ujít vaší pozornosti.

Bohužel se mi nepodařilo vypátrat, jak velké nároky si  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  dělá na systémové prostředky. Straším verzím (1.1.\*) údajně stačila k životu 486/8MB RAM spolu s libovolným klonem operačního systému Unix a X-windows, nebo OS/2 s (XFree). Pro ty, kdo by snad chtěli provozovat tento program na platformách rodiny *Microsoft*, lze  $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  vcelku bezproblémově spustit pod emulátorem *cygwin* [15].

**Poznámka:** Před pár lety započal Matthias Ettrich práci na zbrusu novém projektu, původně určeném pouze pro KDE (postaveném na Qt), nesoucí jméno  $\text{K}_{\text{L}}\text{T}_{\text{E}}\text{X}$ . Byť  $\text{K}_{\text{L}}\text{T}_{\text{E}}\text{X}$  údajně předběhl svojí dobu, osobně jsem se bo-

hužel s tímto programem nestihl blíže seznámit, poněvadž jak rychle tento projekt započal, tak rychle zase skončil. Ostatní vývojáři LyX Teamu však s Ettrichem jeho nadšení pro KLyX příliš nesdíleli, a tak se Ettrich rozhodl tento „suchý pahýl“ odříznout sám. Pochybnosti o tom, že se KLyX bude ubírat správným směrem, podle všeho ještě více prohloubily (tehdy) prapodivné licenční podmínky knihovny Qt, díky nimž by LyX úzce provázaný s touto knihovnou mohl snadno zmizet v komerčním prostředí. Navíc další krok ve vývoji LyXu vedl k roztržení na jádro (*kernel*) a rozhraní (*front-end*), které mělo migrovat k nezávislosti na platformě (*GUI Independent*). Počínaje verzí 1.2.0 je front-end diverzifikován na tři varianty:

- xforms (návaznost na standardní LyX) – dokončeno
- Qt 2 (port pro rozhraní KDE) – téměř dokončeno
- Gtk (port pro rozhraní GNOME) – z větší části nedokončeno

Aktuální informace o stavu vývoje lze dohledat na vývojářských stránkách [1, Developers' Web Site].

Závěrem bych chtěl poznamenat, že se jedná o program s otevřeným zdrojovým kódem. Tato skutečnost zjevně nepředstavuje pro úplného začátečníka žádnou výhodu – nepřepokládá se, že si program před vlastní instalací zcela přetvoří ke svým potřebám, nicméně o této možnosti bude ještě v průběhu článku řeč. Pozitivní myšlenkou budiž to, že je tento program (podobně jako většina programů s otevřeným zdrojovým kódem) k dispozici zcela zdarma a legálně (konkrétní licenční podmínky viz [1]).

## Jak to funguje?

Vraťme se ještě k označení WYSIWYM. LyX sám o sobě se na samotné sazbě textu nijak nepodílí. Jeho výstupem je zdrojový kód, který je zpracován L<sup>A</sup>T<sub>E</sub>Xem do výsledné podoby. Mimochodem to je jeden z důvodů, proč lze výsledný dokument (původně napsaný v LyXu) považovat po typografické stránce za velmi zdařilý. Zpracování dalších, zejména odkládacích souborů, je taktéž výhradně v kompetenci L<sup>A</sup>T<sub>E</sub>Xu.

Dále je třeba zdůraznit, že podoba textu, tak jak ho autor vidí v editačním okně LyXu, nemá takřka nic společného s tím, jak bude výsledný dokument v konečné podobě vypadat. Tato vlastnost by měla začínajícímu uživateli pomoci oprostít se od tzv. *vizualizačního* způsobu tvorby dokumentů a nenásilně přejít ke *strukturovanému*.

Principem strukturování (někdy též značkování) je pojmenování určitých atributů textu (titulek, nadpis, podnadpis, apod.) pomocí maker a teprve až v konečné fázi přemýšlet nad tím, jakou (vizuální) podobu jednotlivým makrům vdechnout. Pokud není autor textu a sazeč jedna a tatáž osoba, je tedy z tohoto pohledu více než žádoucí, aby text určený k sazbě neobsahoval žádné vizuální značky.

Dalším překvapením, které uživatele konvenčních textových procesorů po napsání prvních pár řádek v  $\text{L}\text{Y}\text{X}$  přinejménším trochu zaskočí, je to, že odražení textu pouhým opakováním úderů do mezerníku se text ne a ne posunout, ačkoli je takto si počínající uživatel skrze stavovou řádku nabádán k přečtení tutoriálu. Obdobná situace nastane při odsazování odstavců, případně při vkládání vertikálních mezer opakovaným stiskem klávesy **Enter**.

Tato skutečnost je jednoduše zapříčiněna tím, že  $\text{T}\text{E}\text{X}$  jako jeden z mála sázecích systémů nezná znak „prázdná mezera“, neboli jednotku 1 úhoz. Vložení libovolného množství prázdných mezer, (které lze vložit pouze do zdrojového kódu) se navenek projeví tak, že do výsledného textu se vysází pouze jedna. Za povšimnutí také stojí to, že zatímco  $\text{T}\text{E}\text{X}$  určité operace ignoruje, nebo je zpracuje pomocí vlastních (vnitřních) pravidel,  $\text{L}\text{Y}\text{X}$  je uživateli vůbec nedovolí provést. Z tohoto kontextu vyplývají určitá omezení pro „běžného uživatele“, která jsou vlastní zejména interaktivním textovým procesorům.

## Co umí $\text{L}\text{Y}\text{X}$ ?

V této sekci bych rád vyzdvihl rysy, které jsou pro  $\text{L}\text{Y}\text{X}$  charakteristické. Dále je třeba zdůraznit, že ve srovnání s  $\text{L}\text{A}\text{T}\text{E}\text{X}$ em není uživatel  $\text{L}\text{Y}\text{X}$ u ve skutečnosti o nic ochuzen, poněvadž v případě potřeby může do komponovaného textu vložit přímo kus zdrojového kódu  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u a pokrýt tak zcela jeho možnosti. Ostatně tento text je především o  $\text{L}\text{Y}\text{X}$ u (ne o  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u), proto zde bude pojednáno pouze o tom, co je v možnostech samotného  $\text{L}\text{Y}\text{X}$ u, případně dalších programů s ním souvisejících. Na uživatele  $\text{L}\text{Y}\text{X}$ u nejsou z hlediska znalosti programovacího jazyka  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u resp.  $\text{T}\text{E}\text{X}$ u kladeny žádné požadavky, takže v  $\text{L}\text{Y}\text{X}$ u mohou bez problémů psát i lidé s nízkou počítačovou gramotností. Znalost  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u není ovšem při psaní v  $\text{L}\text{Y}\text{X}$ u překážkou – spíše naopak. Mnohdy je velmi užitečné vědět, co se pod dialogy a menu vlastně skrývá, což často pomůže uživateli pochopit programátorský záměr, který může být laikovi zcela ukryt. Takový uživatel pak není odkázán výhradně na uživatelský manuál, ale mnohdy si umí při řešení problému poradit sám.

## Sazba matematiky

O  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u resp.  $\text{T}\text{E}\text{X}$ u se tvrdí, že právě sazba matematiky je jeho nejsilnější stránkou a že tento mocný nástroj nebyl doposud žádným systémem v tomto směru překonán. Konečně to bylo i jeho původní poslání. Ano, na vzorec vysázený z  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u je radost pohledět, ovšem ti z vás, kteří v  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u již nějakou tu řádku napsali, vědí, že za tím vězí hromada práce. Obzvlášť, jedná-li se o vzorce složitějšího rázu.

Osobně se domnívám, že právě zde je to místo, kde mají interaktivní editory navrch, neboť autor má tvoření matematické sazby přímo pod kontrolou,

namísto aby sledoval, jestli někde udělal o jednu závorku méně nebo více. To je také důvod, proč je psaní matematického textu v  $\text{L}\text{Y}\text{X}$ u nesrovnatelně rychlejší než v samotném  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u. Výsledná podoba je ovšem naprosto stejná. Zatímco pro laiky představuje tato možnost velký přínos, někteří ortodoxní zastánci  $\text{PLAIN}\text{L}\text{A}\text{T}\text{E}\text{X}$ u tento názor však nesdílí. Jako příklad uvádím vzorec pro hustotu Rayleigh-Riceova rozdělení. Srovnání výše uvedených postupů ilustruje Tab. 1. Úsudek zcela ponechám na čtenáři.

Začátečníci pravděpodobně s chutí využijí psaní matematických výrazů za pomoci grafické paletky. Pokročilejším uživatelům  $\text{L}\text{Y}\text{X}$  nabízí velké množství „horkých kláves“, takže není třeba sundávat ruce z klávesnice a složitě se „proklikávat“ k požadovanému výrazu pomocí myši.

Vzmemme-li v úvahu všechny matematické příkazy  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u, může pomýšlení na fakt, pamatovat si tak obrovské množství kombinací, u některých jedinců vyvolat hrůzné představy, čímž se pro ně záhy stává takový požadavek neakceptovatelný. Zde bych rád vyzdvihl možnost  $\text{L}\text{Y}\text{X}$ u psaní matematických výrazů (v tzv.  $\text{T}\text{E}\text{X}$ -módu) tak, jakoby se jednalo o zápis zdrojového kódu  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u. Během tohoto procesu ale  $\text{L}\text{Y}\text{X}$  příkazy automaticky přetváří do vizuální podoby, takže např. po napsání příkazu `\frac` se v editačním okénku vytvoří zlomková čára se dvěma poli, do nichž je možno vepisovat další výrazy, které lze pochopitelně dle libosti dále do sebe vnořovat.

Pokud jste tedy byli doposud zvyklí psát texty přímo pomocí zdrojového kódu  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u, můžete nabyté vědomosti s výhodou používat i nadále bez nutnosti učení se novým příkazům, čímž se přechod k  $\text{L}\text{Y}\text{X}$ u stane zcela bezbolestným.

Jelikož jsou definice horkých kláves  $\text{L}\text{Y}\text{X}$ u uloženy v textovém souboru (který lze běžně editovat), může se tento soubor verze od verze a uživatel od uživatele lišit, zatímco makra  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u zůstávají za všech okolností stejná.

Zmíňme se ještě o vizualizaci při sazbě matematiky v porovnání se standardním způsobem. Píše-li autor přímo zdrojový kód  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u v běžném textovém editoru, je často nucen představit si výslednou sazbu v hlavě. Otázka tedy zní, proč si s takovýmto úkonem lámat hlavu, když tuto úlohu hravě zvládne počítač. V  $\text{L}\text{Y}\text{X}$ u je pochopitelně uživatel této námahy zbaven.

Další úlevou je psaní rovnic v prostředí `equation`. Není nic snazšího než stisknout kombinaci kláves `Alt-m-d` – vytvoří se prostředí pro samostatně vysázenou rovnici (*displayed formula*) a `Ctrl-Enter` (tím se vytvoří šablona typu: levá strana – střed – pravá strana). Rozšíření soustavy o další rovnici snadno provedeme opětovným stiskem `Ctrl-Enter`. Stejně jednoduché je i číslování rovnic. Stiskem `Alt-m-n` se automaticky očíslovuje celá soustava rovnic.

Pokud je u některé rovnice číslování nežádoucí nebo nemá smysl (např. u dlouhých rovnic, kde jedna ze stran přeteče a musí být rozepsána na více řádek), je třeba nastavit kurzor na příslušný řádek a do stavové řádky  $\text{L}\text{Y}\text{X}$ u napsat příkaz `math-nonumber`. Pro tuto operaci lze také použít sekvenci kláves `Alt-m-Shift-n`. Naopak, chceme-li se odkazovat na nějakou rovnici soustavy,

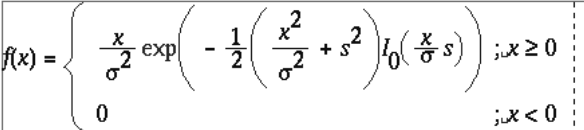
Zápis v L<sup>A</sup>T<sub>E</sub>Xu

```
\[
  f(x)=\left\{
    \begin{array}{lr}
      \frac{x}{\sigma^2} \exp
      \left( \frac{1}{2}
        \left( \frac{x^2}{\sigma^2} + s^2 \right)
        I_0\left(\frac{x}{\sigma} s\right)
      \right) & ; x \geq 0 \\
      0 & ; x < 0
    \end{array}
  \right.
```

Zápis v PLAINU

```
$$
f(x)=\cases{
  {x \over \sigma^2} \exp
  \left( -{1 \over 2}
    \left( {x^2 \over \sigma^2} + s^2 \right)
    I_0 \left( {x \over \sigma} s \right)
  \right)
  & ; x \geq 0 \\
  0
  & ; x < 0}
$$
```

Screenshot pořízený z editačního okna L<sup>A</sup>T<sub>E</sub>Xu


$$f(x) = \begin{cases} \frac{x}{\sigma^2} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma^2} + s^2\right) I_0\left(\frac{x}{\sigma} s\right)\right) & ; x \geq 0 \\ 0 & ; x < 0 \end{cases}$$

Vysázený vzorec

$$f(x) = \begin{cases} \frac{x}{\sigma^2} \exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma^2} + s^2\right) I_0\left(\frac{x}{\sigma} s\right)\right) & ; x \geq 0 \\ 0 & ; x < 0 \end{cases}$$

Tab. 1: Odlišná pojetí tvorby matematické sazby

je třeba na příslušném řádku vložit návěští (*label*). Toho lze jednoduše docílit kombinací kláves Alt-i-l. Do editačního okénka vepíšeme jméno návěští.

Některé starší verze  $\text{L}\text{Y}\text{X}$  si neuměly poradit s mezerami v návěští. Nechcete-li se vystavovat zbytečnému riziku, prázdné mezery do návěští nepište. Pro lepší orientaci mezi návěštími (viz dále) je dobré před jméno návěští vepsat např. třípísmenný kód typu eqn: – rovnice, fig: – obrázek, tab: – tabulka, sec: – sekce apod.  $\text{L}\text{Y}\text{X}$  ve verzi 1.2.0 nabídne některé předpony automaticky.

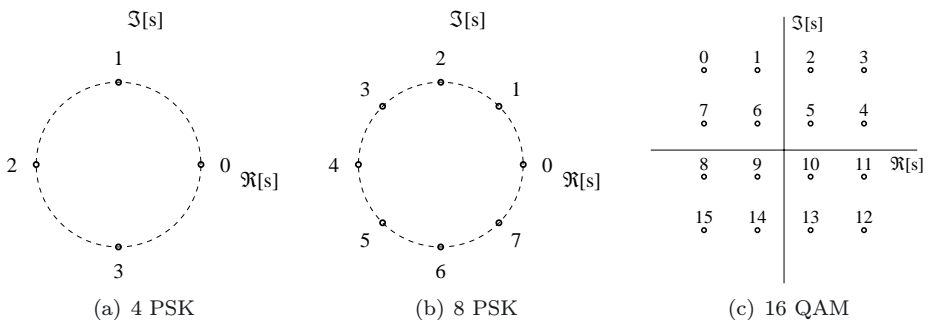
Poněkud těžkopádnější je tvorba matic a vybraných typů velkých závorek, které se musí zadávat přímo z paletky. Tento úkon lze ale zmechanizovat vytvořením vhodného makra. O tvorbě maker bude stručně pojednáno v sekci „ $\text{L}\text{Y}\text{X}$  pro pokročilé“.

### Vložení obrázku

Vložení obrázku v  $\text{L}\text{Y}\text{X}$  hravě zvládne i uživatel, který o mechanismu začlenění obrázku nemá ani potuchy. Verze 1.2.0 již disponuje multifunkčním dialogem, který dovoluje vkládat obrázky nejrozličnějších grafických formátů.

Pokud obrázek není ve formátu, kterému rozumí  $\text{L}\text{A}\text{T}\text{E}\text{X}$ , tj. EPS – zapouzdřený PostScript (*encapsulated PostScript*),  $\text{L}\text{Y}\text{X}$  si zavolá na pomoc program ImageMagick, který export do tohoto formátu obstará. Ostatně exportem do EPS dnes disponují snad všechny programy, které mají co do činění s grafikou, včetně mnoha počítačových algebraických systémů (MATLAB, MATHEMATICA, MAPLE a další).

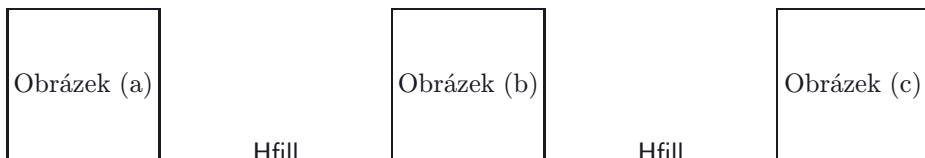
$\text{L}\text{Y}\text{X}$  kromě vkládání samostatných obrázků nabízí taktéž vcelku pohodlné rozhraní pro umísťování obrázků v prostředí *subfigure*, čehož s výhodou využijeme tam, kde se obrázek vyskytuje v několika variantách.



**Obr. 2:** Příklad umístění obrázků ve více variantách – konstelační diagram digitálních modulací

Je-li v dokumentu vyžadován seznam obrázků (*list of figures*), nejsou popisky

dílčích podobrázků (*subcaptions*) do tohoto seznamu zahrnuty. Ostatně tento seznam má pomoci čtenáři rychle se zorientovat v textu. Úplný výpis všech popisek zde představuje až přemíru informací – často působí i rušivě. Obrázky lze umisťovat mnoha způsoby a práce s nimi je obdobná jako práce s boxy (viz dále). Příkladně na obr. 2 byla použita konstrukce



Popisky obrázků jsou v L<sup>A</sup>T<sub>E</sub>Xu centrovány a jejich implicitní umístění je zpravidla pod, ale mohou být i nad obrázkem. V odborné literatuře se ale často můžeme setkat s umístěním popisek vedle obrázků, což je po estetické stránce velmi pěkné, ale bezesporu hlavním důvodem je úspora místa. L<sup>A</sup>T<sub>E</sub>X interaktivně možnost vložení popisky vedle obrázku bohužel nenabízí, ale tento nedostatek se dá snadno obejít konstrukcí



Podotýkám, že ačkoli tato konstrukce působí poněkud kostrbatě (začátečník vůbec nemusí vědět, oč tady běží) jedná se o doporučení převzaté z uživatelského manuálu. Méně zkušené uživatele odkazují právě tam. Na obhajobu L<sup>A</sup>T<sub>E</sub>Xu však musím dodat, že jedním z bodů, které si vývojáři L<sup>A</sup>T<sub>E</sub>Xu do budoucna vytyčili, je mimo jiné zdokonalení rozhraní pro vkládání atypických popisek a rozšíření na spolupráci s balíčkem `caption2`.

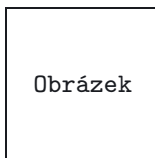
Mezi další často kladené otázky patří obtékání textu kolem obrázků. Textové procesory (WYSIWYG) umožňují nenásilně umístit obrázek vcelku kamkoli pouhým uchopením a tažením pomocí myši. Tomu se však zákonitě musí podřídit také obtékání textu. Dokonce i v některých tiskovinách se tak můžeme setkat s jistým fenoménem, kde sloupec mezi obrázkem a okrajem textu je tak úzký, že se do něj vejde pouze jedno nebo dvě slova. Vzhledem k tomu, že text má být zarovnán do bloku, je textový procesor nucen spolu s mezislovními mezerami taktéž zvětšit i odsazení mezi jednotlivými písmeny. Takový text, který vypadá, jako by se jednalo o tzv. *p r o s t r k á n í*, pak působí zvýrazňovacím dojmem, ačkoli autor textu zjevně tento úmysl neměl.

Častým výskytem těchto hrubých chyb jsou také skripta a různé texty polo-profesionálního charakteru.  $\text{\LaTeX}$  byste k podobným „zvěrstvům“ přinutili leda násilím. Možná právě to je ten pravý důvod, proč je začátečník díky (skryté) složitosti od svého záměru odrazen. Z výše uvedených skutečností je zcela patrné, že tento úkon nemohu v žádném případě doporučit úplným začátečníkům. I zkušení typografové se shodnou na tom, že při umísťování obrázků obtékaných textem je třeba počínat si nanejvýše opatrně. Obrázek může být totiž snadno umístěn velmi nevhodně v kontextu s okolním textem. Dost ale teorie. Pojdme se podívat na to, jak obrázek obtékaný textem vložit do  $\text{\LaTeX}$ u.

Ve skutečnosti existuje hned několik možností, jak toho docílit. Uživatelsky nejpohodlnější je užití volby z menu pro styl odstavce: `Layout/Paragraph`. Kurzor umístíme na řádek s obrázkem (uvnitř plovoucího prostředí) a na záložce `Extra` zvolíme volbu `Floatflt`. V poli `Width` musí být specifikována šířka neviditelného rámečku (v absolutních nebo relativních jednotkách), podle kterého se bude obtékaný text zarovnávat. Doporučená hodnota je 100 % velikosti obrázku. Zarovnání obrázku je implicitně nastaveno na horní okraj, ale podle potřeby jej lze změnit buď na střed, nebo spodní okraj.

Začátečníka patrně zaskočí to, že text v editačním okně  $\text{\LaTeX}$ u zůstane nezměněn. O to větší je pak jeho úleva, když zjistí, že po překladu kódu  $\text{\LaTeX}$ em k obtečení textu ve vysázeném dokumentu skutečně došlo. Je-li dokument definován jako dvoustranný, umístí se obtékaný obrázek buď k pravému nebo levému okraji podle toho, zdali se jedná o lichou resp. sudou stranu. Pokud z nějakých důvodů chceme toto implicitní nastavení změnit, je třeba explicitně použít některý z balíčků `floatflt` nebo `wrapfig`, které by měly být zahrnuty ve standardní distribuci  $\text{\LaTeX}$ u. Obrázek pak musíme „obalit“ příkazy prostředí `floatingfigure`.

```
\begin{floatingfigure} [⟨pozice⟩] {⟨šířka⟩}
```



```
\end{floatingfigure}
```

Volitelný parametr `⟨pozice⟩` řídí umístění obrázku a může nabývat hodnot `r` – vpravo, `l` – vlevo, `p` – vpravo nebo vlevo, je-li číslo stránky liché resp. sudé a `v` – obsahuje-li deklarace balíčku doplňkové volby (detaily viz dokumentace k `floatflt`). Obdobně je možno použít prostředí `floatingtable` pro obtékání textu kolem tabulek. Ve verzi 1.2.0 byla (z mně neznámých důvodů) ponechána pouze možnost obalení obrázku příkazy prostředí `floatingfigure`.



## L<sup>A</sup>T<sub>E</sub>X a Xfig

Historicky pravděpodobně nejčastějším kamenem úrazu byla tvorba obrázků. Dnes již díky nepřehlednému množství programů na tvorbu grafiky patří tyto doby nenávratně minulosti. Název podsektory prozrazuje cosi o možnostech spolupráce L<sup>A</sup>T<sub>E</sub>Xu s jedním z těchto programů. Ačkoli si L<sup>A</sup>T<sub>E</sub>X neklade žádné nároky na volbu programu, nejvíce mu podle mého soudu „sluší“ programy s otevřeným zdrojovým kódem. Velmi oblíbený (a pravděpodobně asi nejpoužívanější) je program Xfig, který byl ještě donedávna asi jediným interaktivním programem na tvorbu vektorové grafiky pro Linux.

Tento program, který na první pohled působí, řekl bych, až primitivním dojmem, se asi těžko může rovnat všemocným nástrojům typu PhotoShop, CorelDraw apod., ale jeho síla je vskutku veliká. Patrně proto se pozornost tvůrců L<sup>A</sup>T<sub>E</sub>Xu obrátila právě k němu. Jeho snadné ovládání, dostupnost, nízké hardwarové nároky a právě práce s vektorovou grafikou ho předurčují zejména ke tvorbě obrázků technické literatury, ale umí se celkem obstojně vypořádat i s obrázky složitějšího rázu. Zájemci nechtě nahlédnou do příslušné dokumentace.

Jedna z fundamentálních pouček typografie praví, že musí být nekompromisně dodržena absolutní jednota v textu. To znamená, že zvolím-li určitou velikost a řez písma pro nadpis úrovně jedna, pak tuto volbu musím použít pro všechny nadpisy stejné úrovně. Málokoho už ale napadne, že obdobné pravidlo také platí i v tvorbě obrázků. Obsahuje-li dokument např. bloková schémata, pak všechny bloky (ve všech obrázcích) musí být kresleny stejnou tloušťkou čáry a pokud je to jen trochu možné, měly by mít i stejnou velikost. Obdobně při tvorbě grafů je třeba dodržet tloušťku čar a jejich poměr vůči tloušťce os, případně rastrů (*grids*) apod.

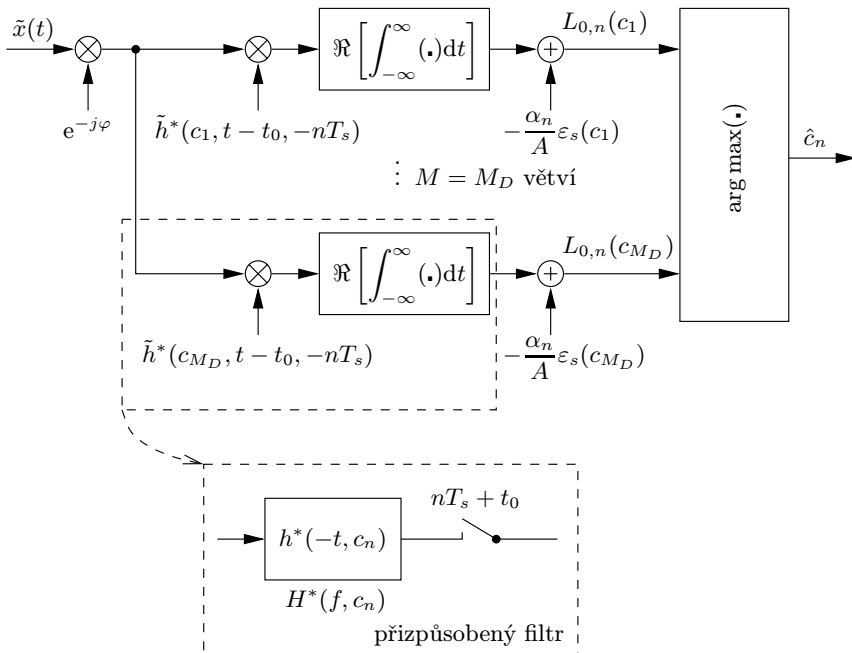
Otevřeným problémem, který byl po dlouhou dobu trnem v oku řadě typografů, je volba písma textu vyskytujících se v obrázcích. Zkušený typograf (nebo alespoň ten, který dělá svoji práci poctivě) se snaží vždy zvolit písmo, které harmonuje s písmem těla dokumentu a s kontextem příslušného díla – např. technická literatura.

Sofistikované sazecí systémy, jako je T<sub>E</sub>X, však umožňují pracovat s textem v obrázcích stejně tak, jako by se jednalo o text těla dokumentu. Tímto způsobem je možno do obrázků vkládat třeba i matematické formule a vzorce, zapsané pomocí vnitřního jazyka T<sub>E</sub>Xu. Aby T<sub>E</sub>X ale v místě výskytu tento kód interpretoval správně, musí text s příkazy nést příznak *special*.<sup>5</sup> Jinak by se např. místo značky integrálu vysázel text `\int`.

Vytvoření obrázku obsahujícího text s příznakem *special* je v Xfigu více než snadné. Pokud takové obrázky tvoříte často, můžete tuto volbu nastavit i implicitně. Poté je zapotřebí obrázek vyexportovat do dvou částí – jednou jako vlastní obrázek (bez textu) s příponou `pstex` a podruhé jako soubor se samotným

---

<sup>5</sup>Nezaměňovat s T<sub>E</sub>Xovým primitivem `\special`.



**Obr. 3:** Obrázek s textem typu *special* – korelační detektor a jeho varianta s přizpůsobenými filtry

(Obrázek byl vytvořen podle [3, str. 294])

textem s příponou `pstex.t`. Ten se pomocí nástrojů pro začlenění externích souborů (nezaměňovat s importem) vloží do `LyXu`.

Novější verze `LyXu` ale umějí pracovat přímo se soubory `Xfig` (s příponou `fig`). Přesněji řečeno `LyX` si obě části před začleněním obrázku do dokumentu vytvoří sám, takže proces sazby se zase o jeden krůček urychlí. Postup: `Insert/External Material`. Jako `Template` zvolit `Xfig`. `Template` ale nelze použít tehdy, pokud obrázek obsahuje některý další (externí) objekt (obrázek, graf); více informací v sekci „`LyX` – nedostatky“.

Pokud je obrázek do `LyXu` začleněn skrze šablonu (*template*) nebo z dialogu pro vložení obrázku, může být soubor s obrázkem umístěn i mimo běžný adresář. `LyX` zároveň do zdrojového kódu uloží informaci o jeho umístění, a tak překlad kódu zpravidla neprovázejí žádné potíže. Je-li ale použita varianta začlenění pouze skrze soubor `pstex.t`, máme na výběr ze dvou možností:

- 1) Oba soubory (`pstex` a `pstex.t`) musí být umístěny v běžném adresáři. V případě, že dokument obsahuje velké množství takových obrázků, dojde

k jejich promíchání s odkládacími soubory, které generuje  $\LaTeX$  během překladu zdrojového kódu. Tímto způsobem se zřejmě celkový počet souborů v adresáři zvýší na trojnásobek. Orientace v takové „džungli“ pak vyžaduje pevné nervy a notnou dávku trpělivosti.

- 2) Tomuto jevu se dá v celku snadno zabránit tím, že soubory s obrázky přesuneme do samostatného adresáře.  $\text{LyX}$  ale uloží do zdrojového kódu pouze informaci o umístění `pstex.t`. Proto musíme říci  $\text{LaTeXu}$ , aby se při překladu zdrojového kódu nezapomněl podívat i do adresáře s obrázky (se soubory s příponou `pstex`). Prakticky to znamená, že do systémové proměnné `TEXINPUTS` musíme připojit cestu k nově vytvořenému adresáři. Pro terminály třídy `sh` např. takto:

```
export TEXINPUTS=../fig/:
```

V tomto případě se bude  $\text{LaTeX}$  dívat do běžného adresáře a do podřazeného adresáře `fig`, kde jsou umístěny obrázky. Toto řešení je velmi generické a univerzální, poněvadž při psaní nového dokumentu stačí v běžném adresáři vytvořit adresář s názvem `fig` a do něj umisťovat pouze obrázky. Dále je třeba podotknout, že oddělení jednotlivých cest dvojtečkami je třeba dodržet i na konci. Pokud nechcete, aby se odkládací soubory ukládaly do běžného adresáře, vypněte volbu `Temp dir` v menu `Edit/Preferences/Inputs` (nebo do pole vepište jinou cestu, např. `/tmp`). U starších verzí  $\text{LyX}$  je zapotřebí provést změnu cesty k odkládacím souborům v konfiguračním souboru  $\text{LyX}$  `lyxrc`.<sup>6</sup>

```
\tempdir_path ""  
\use_tempdir false
```

Lhostejno, který způsob začlenění obrázků používáte, vždy je lepší přesunout obrázky do samostatného adresáře a v běžném adresáři ponechat pouze soubory, které bezprostředně souvisejí se samotným dokumentem.

## Křížové odkazy

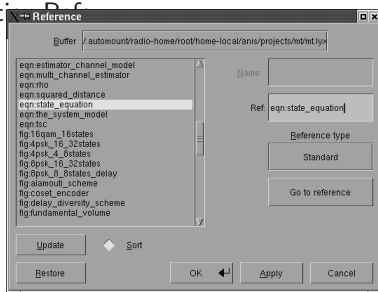
Mezi další nesporné výhody  $\text{LyX}$ u patří správa návěští (*labels*) při tvorbě křížových odkazů. Píšete-li rozsáhlou technickou publikaci, kde se to těmito objekty jenom hemží, pak takovou pomoc jistě velmi uvítáte.  $\text{LyX}$  umí návěští abecedně seřadit, takže použijete-li již dříve zmíněné doporučení s třípísmennými předponami, dostane tato hromádka nepřeborných slovíček pevný řád. Seznam je pak velmi přehledný a orientace v něm je snazší. Pokud byste přece jen zapomněli, co se pod daným návěstím skrývá, můžete si pomocí tlačítka `Goto reference` odskočit na ono místo a tlačítkem `Go back` se vrátit zpět, aniž byste se mezi tím museli k požadovanému návěstí prodírat únavným „scrollováním“ obrazovek s textem. Menu `Insert/Cross Reference` pochopitelně umožňuje kromě vložení samotných

---

<sup>6</sup>Na platformách Unix se tento soubor nachází v adresáři `~/lyx/`.

referencí i čísla jejich stránek, případně textové reference, kde číslo strany je nahrazeno textem typu „následující“, „předchozí“ apod. Skrze tento dialog se  $\text{L}\text{Y}\text{X}$  umí také odkazovat na reference z jiného dokumentu (viz dále).

Obdobné možnosti skýtá i menu **Insert/Citate Reference** pro vkládání odkazů na citované prameny, jejichž doslovný název se zobrazí ve stavovém okénku. Dialog pro vkládání citovaných pramenů zároveň plní funkci rozhraní mezi  $\text{L}\text{Y}\text{X}$ em a programem  $\text{Bib}\text{T}\text{E}\text{X}$ , takže nemusíte mít obavy, že vámi pracně nashromážděné položky v databázi přijdou nazmar. V případě většího počtu referencí lze citace sestavit do výčtu, kde posloupnost [1], [3], [4] je nahrazena výrazem [1, 3, 4]. Pořadí citovaných pramenů lze libovolně měnit. Citace lze specifikovat doplňkovým textem typu [3, str. 156].



## Sazba ministránek

V  $\text{L}\text{Y}\text{X}$ u lze také celkem pohodlně provést sazbu tzv. „ministránek“ (*minipages*).<sup>a</sup> Jakým způsobem toho docílit, je velmi ilustrativně popsáno v helpu, proto se jím zde zabývat nebudu. Šířku ministránky lze zadat buď v absolutních, nebo relativních jednotkách podle toho, co je pro danou situaci výhodnější.

<sup>a</sup>Nezaměňovat s prostředím pro sazbu více sloupců `multicols`.

Jako příklad jsem použil rozdělení v poměru 1:3 a 2:3 šířky sloupce. K ministránkám se lze částečně chovat jako k boxům, proto  $\text{L}\text{Y}\text{X}$  také nabízí volbu oddělení ministránek výplní `Hfill` (srovnej s obr. 2). Ministránka se ale dá také krom samotného textu „zneužít“ ke zcela jiným účelům. Příkladně, potřebujeme-li přinutit  $\text{L}\text{A}\text{T}\text{E}\text{X}$ , aby kapitolu vysázel na sudou stranu, nebo tehdy, chceme-li např. pod prostředí `part` uvést stručný výtah z toho, co může čtenář v následujícím oddílu (části) očekávat. Závěrem této sekce bych chtěl zdůraznit, že sazba ministránek vyžaduje určitý cit. Jelikož má ministránka charakter boxu, nedojde při „přetečení“ k zalomení textu, ale k přesunu na další stranu.

## $\text{L}\text{Y}\text{X}$ a barvy

Další možností, kterou  $\text{L}\text{Y}\text{X}$  nabízí, je práce s barvami. Pravda je spíše taková, že v tomto směru nelze od  $\text{L}\text{Y}\text{X}$ u očekávat žádné zázraky, ale i tak bych se chtěl o této možnosti velmi letmo zmínit. Vzhledem k tomu, že práce samotného  $\text{T}\text{E}\text{X}$ u (a tím pádem i jeho libovolné nadstavby) spočívá pouze ve skládání

tzv. *boxů*, které mají v reálném světě připodobnění v tiskařských literách, je mu z tohoto pohledu naprosto lhostejné, co se v oněch „krabičkách“ nachází. Tím spíše pak neřeší práci s barvami.<sup>7</sup> Barevnou informaci lze přidat teprve až na úrovni PostScriptu. L<sup>A</sup>T<sub>E</sub>X dokáže obarvit text pouze standardními typy barev podle modelů RGB a CMYK. Barva se volí jako atribut textu z menu Layout/Character/Color. Zde si ale neodpustím jisté varování, poněvadž vložení barevného textu může na celkovém vzhledu napáchat více škody než užitku. Snad proto, že se ve standardních dokumentech barva vyskytuje velmi zřídka, není asi příliš žádoucí tento „nedostatek“ odstraňovat. Pokud požadujete text ve speciálním barevném provedení, můžete si namíchat barvu podle vlastního přání z výše uvedených modelů, případně odstínů šedi na úrovni příkazů L<sup>A</sup>T<sub>E</sub>Xu nebo PostScriptu.

## L<sup>A</sup>T<sub>E</sub>X a PDF

Obliba formátu PDF (*Portable Document Format*) v poslední době vzrostla natolik, že i tvůrci L<sup>A</sup>T<sub>E</sub>Xu museli k této skutečnosti přihlédnout. Export do formátu PDF nabízely již starší verze L<sup>A</sup>T<sub>E</sub>Xu – jednak pomocí pdfL<sup>A</sup>T<sub>E</sub>Xu (jehož výstupem, jak sám název napovídá, je přímo PDF), ale také skrze volně dostupné programy *dvips* a *ps2pdf*. Program *dvips* implicitně používal bitmapové fonty, které byly generovány v rozlišení určeném pro tiskárnu (typicky 300 dpi a 600 dpi). Změna rozlišení fontů však není příliš triviální operací. Prohlížeč PDF pak ve snaze zmenšit rozlišení fontů, tak aby je bylo možno zobrazit na obrazovce monitoru, produkoval výsledek nevalné kvality. Ostatně i kvalita tisku může být v tomto smyslu poněkud zavádějící, není-li rozlišení tiskárny totožné s rozlišením fontu.

Měl-li tedy dokument vyhlížet kvalitně, bylo zapotřebí programu *dvips* vnutit tzv. „obrysové“ (*outline*) fonty.<sup>8</sup> Standardní distribuce L<sup>A</sup>T<sub>E</sub>Xu od jisté doby obsahují kromě klasických CM (Computer Modern) fontů také jejich postscriptovou verzi ve formátu Type1, nazvanou *Blue Sky*. Jelikož pdfL<sup>A</sup>T<sub>E</sub>X implicitně používá právě tyto (Type1) fonty a *dvips* ne, bylo zapotřebí *dvips* k tomuto úkonu donutit modifikací konfiguračního souboru *config.ps*<sup>9</sup> (globálně), případně *~/dvipsrc* (lokálně):

```
p +psfonts.cmz
p +psfonts.amz
```

Tyto mapy bývají obsaženy ve standardní distribuci T<sub>E</sub>Xu, nebo je lze stáhnout z archívu CTAN. Pokud je použito implicitní kódování L<sup>A</sup>T<sub>E</sub>Xu (OT1), není zapotřebí podnikat další kroky. Ovšem jsou-li použity fonty v kódování T1, pak L<sup>A</sup>T<sub>E</sub>X sáhne po fontech s rozšířeným kódováním (tzv. EC fontech), které jsou

---

<sup>7</sup> Ano, T<sub>E</sub>X „pouze“ skládá za sebe boxy, ale tak mistrně, že dodnes v této rutinní práci nenašel konkurenci. Kéž by i v jiných oblastech panovala stejně přísná dělba práce.

<sup>8</sup> Nezaměňovat se slovem *outline*, jakožto vzhledem fontu.

<sup>9</sup> Cesta je odvislá od distribuce, zpravidla však \$TEXMF/dvips/config/.

v podstatě shodné s CM, ale obsahují navíc ještě některé speciální znaky pro jiné než anglické abecedy. Jelikož EC fonty dlouho neexistovaly ve verzi Type1, dal se tento nedostatek částečně obejít pomocí balíčků `ae` a `aecompl`, které emulují fonty v kódování T1 pomocí Type1 fontů v kódování OT1. Ve verzi LyXu nižší než 1.1.6fix4 je třeba zavést tyto balíčky v preambuli příkazem

```
\usepackage{ae,aecompl}
```

Počínaje verzí 1.1.6fix4 však lze tuto volbu aktivovat z dialogu `Layout/Document/Document/Fonts`. Piktogramy, které v CM fontech chybí, jsou nahrazeny z EC fontů a proto nezbyvá, než je zobrazit jako bitmapy. LyX implicitně používá fonty v kódování T1. Postscriptové fonty sice všechny T1 piktogramy obsahují, ale na druhou stranu neobsahují fonty tučných symbolů. Proto je zapotřebí při překladu pdfL<sup>A</sup>T<sub>E</sub>Xem nahradit příkaz `\boldsymbol` příkazem `\pmb` (*poor man's bold*) například takto:

```
\renewcommand{\boldsymbol}[1]{\pmb{#1}}
```

Až na malé výjimky, Computer Modern a postscriptové fonty dávají v obou případech dobré výsledky a je čistě věcí vkusu, kterému z nich dáte přednost.

Pravděpodobně nejpohodlnější způsob práce s postscriptovými fonty nabízí balíčky obsažené ve standardní distribuci L<sup>A</sup>T<sub>E</sub>Xu, které většinu otročké práce provedou za vás. Tyto balíčky nahradí CM případně EC fonty odkazy na fonty, které jsou implicitně obsaženy v každém postscriptovém interpretu. Balíčků, skrze které lze provést tuto substituci, existuje hned několik. Mezi hlavní patří `times`, `palatino`, `newcent` a `bookman`. Balíček `times` používá font Times-Roman pro serifové písmo, font Helvetica pro bezserifové písmo a font Courier pro strojpisné písmo. Tyto (postscriptové) fonty se však od CM a EC fontů podstatně liší. Proto při jejich použití dojde ke změně vzhledu dokumentu. Pokud není z nějakých důvodů explicitně vyžadována exaktní podobnost s CM fonty, nečiní jejich použití v běžném textu žádný problém. Potíže nastanou teprve při sazbě matematiky.

Nevýhoda balíčku `times` totiž spočívá v tom, že obstarává substituci pouze textových fontů. Matematické symboly a alfanumerické znaky zůstanou nedotčeny, a proto budou vysázeny z CM fontů. Jelikož rozdíl mezi fonty Times a Computer Modern je poměrně očividný (rozdíl pozná snadno i laik), může být tato neuniformita v řadě případů velkou překážkou. Odhlédneme-li od komerčního MathTimesu, pak tento nedostatek (částečně) odstraňuje až balíček `pslatex`, který používá stejné substituce jako `times`, ale i v matematickém prostředí.

Dále je třeba poznamenat, že základní sada Type1 fontů je součástí každé postscriptové tiskárny a PS/PDF překladače. Proto je při použití fontů z této sady velikost PS (potažmo PDF) souboru o poznání menší, neboť není potřeba s dokumentem zároveň ukládat i fonty. Ať už se rozhodnete pro jakoukoli volbu, LyX obstará potřebná nastavení za vás. Roletka `Fonts` z menu `Lay-`

out/Document/Document obsahuje volby: default (CM), ae, pslatex, times, palatino, helvet, avant, newcent a bookman.

Mezi nejpalčivější problémy při tvorbě PDF dokumentů však patří vložení obrázků. Program pdfL<sup>A</sup>T<sub>E</sub>X totiž neumí pracovat s obrázky ve formátu EPS, ale pouze s formáty PDF, JPEG, případně rastrovými PNG. Pokud je mi známo, soubory EPS lze sice snadno převést pomocí volně šiřitelného programu `eps-topdf` do formátu PDF, ovšem pěkně jeden po druhém. Dokonce ani L<sup>A</sup>X neobsahuje algoritmy, kterými by se tento otrocký postup dal zmechanizovat.<sup>10</sup>

Uživatelé, kteří trochu vládli bashovými nebo perlovými skripty, se snažili tuto mezeru vyplnit po svém. Část těchto „výkřiků do tmy“ pak migrovala ke společnému projektu, který nese název `tex2pdf` (dříve `lyx2pdf`) [10]. Tento skript „parsuje“ soubor L<sup>A</sup>Xu a postupně na něj aplikuje příslušné transformace. Upravený kód je pak přeložen pdfL<sup>A</sup>T<sub>E</sub>Xem. Starší verze L<sup>A</sup>Xu vlastně ani s importem jiných než EPS obrázků ještě zcela nepočítají (tato možnost není přístupná z menu). Okno s tímto dialogem ale lze vyvolat kombinací kláves Alt-x. Poté je zapotřebí do stavové řádky napsat „magickou formuli“ `graphics-insert`. Teprve až s příchodem verze 1.2.0 se objevilo vylepšení v podobě multifunkčního dialogu, skrze který lze importovat nejrozmanitější grafické formáty včetně příslušných transformací (viz konverze pomocí ImageMagick).

Slabina programu `tex2pdf` spočívala v tom, že pro konverzi obrázků používal program `epstopdf`. Ten však buď sám o sobě neprodukoval výsledky valné kvality, nebo si překonvertovaný obrázek příliš nerozuměl s pdfL<sup>A</sup>T<sub>E</sub>Xem. Chyby se často projevovaly v případě, že obrázek obsahoval bitmapu.

Pro tyto nedostatky jsem v programu `tex2pdf` zpočátku nenašel příliš velké zalíbení. Od dob, kdy jsou v teL<sup>A</sup>T<sub>E</sub>Xu (od verze 1.0.6) obsaženy CM fonty ve verzi Type1 (v kódování T1), však program `tex2pdf` ztrácí trochu lesk, poněvadž po aplikaci `ps2pdf` je kvalita výstupu zcela souměřitelná s komerčním Distillerem. Jejich nectností je to, že (zatím) neobsahují ligatury. Řešení tohoto a spousty dalších problémů lze nalézt např. v [11], odkud jsem ostatně čerpal některé z výše uvedených informací. Pokud nejste spokojeni s kvalitou vašeho PDF, pak by tyto stránky rozhodně neměly ujít vaší pozornosti.

Zprvu se zdálo, že jediná výhoda mluvící ve prospěch `tex2pdf` spočívala v tom, že tento skript automaticky obstaral v PDF dokumentu vytvoření lišty s „bookmarky“, aniž by uživatel musel cokoli dělat. Stejného efektu lze ovšem docílit i bez použití pdfL<sup>A</sup>T<sub>E</sub>Xu, přidáním kousku zdrojového kódu do preamble L<sup>A</sup>Xu. Další důkaz toho, že se čas od času vyplatí nahlédnout do dokumentace.

```
\newif \ifpdf
\ifx \pdfoutput \undefined
```

---

<sup>10</sup>Ve starých verzích L<sup>A</sup>Xu se dal tento nedostatek obejít tak, že se u obrázku nezadala přípona. Balíček `graphics` pak zpracoval obrázek jako EPS nebo jako PDF podle toho, byl-li volán L<sup>A</sup>T<sub>E</sub>Xem resp. pdfL<sup>A</sup>T<sub>E</sub>Xem.

```

\pdffalse
\else
\pdftrue
\fi
\ifpdf
\pdfinfo {/Author (<jméno autora>)
          /Title (<titulek>)
          /Subject (<předmět>)}
\pdfcatalog {
            /PageMode (/UseNone)
            /OpenAction (fitbh)}
\usepackage[pdftex]{hyperref}
% pro příznivce METAPOSTu
\DeclareGraphicsRule{*}{mps}{*}{}
\else
\usepackage[ps2pdf]{hyperref}
\fi

```

Předpoklady o tom, že se `tex2pdf` stane s příchodem verze 1.2.0 obsoletní, se bohužel nepotvrdily. Proto bude tento program ještě nějakou dobu přežívat.

## Kontrola pravopisu

Stejně tak jako `LATEX` používá pro sazbu textu jiný program (`LATEX`), obdobná analogie platí i při kontrole pravopisu. `LATEX` sám v sobě totiž žádný program na kontrolu pravopisu implementován nemá, ale nabízí k němu vcelku přívětivé uživatelské rozhraní. Z webu lze stáhnout celou škálu spellcheckerů<sup>11</sup> s otevřenou licencí. Mezi nejznámější (a pravděpodobně asi nejpoužívanější) patří programy `ispell` a `aspell`. `LATEX` poskytuje rozhraní k oběma, takže je čistě věcí uživatele, kterému z nich dá přednost. Vždy je ale zapotřebí nastavit správný jazyk, jinak úloha spellcheckeru zcela pozbývá smysl.

Veškerá nastavení spellcheckeru se provádí skrze menu `Edit/Preferences/Lang Opts/Spell checker`. Pro češtinu je nezbytné vložit do pole `Use escape characters` všechna akcentovaná písmena, jinak spellchecker chápe slovo obsahující akcentovaný znak jako složeninu z více slov. Pokud není tento jev ošetřen, je příkladně slovo »přeložit«<sup>11</sup> chápáno jako »p-řelo-žit«<sup>11</sup>. Potíže se projevovaly zejména ve starších verzích `window-manageru KDE` a to díky nefunkčnosti tzv. „mrtvých“ kláves (*dead keys*). Proto bylo zapotřebí tento nedostatek obejít přímou editací konfiguračního souboru `preferences` (případně `lyxrc`). Ten by pak měl obsahovat následující řádky:

---

<sup>11</sup>Omlouvám se za nedodržení jazykové čistoty, ale termín „kontrolor pravopisu“ se mi pro tento účel nezdá příliš vhodný.



```

\use_escape_chars true
\escape_chars "ěščřžýáíéúůďťňĚŘŽÍÓĀŠĎÝČŇ"
\use_input_encoding true

```

Při novém spuštění LyXu se pak v poli Use escape characters objeví všechna akcentovaná písmena a spellchecker se začne chovat kultivovaně. Dalším krokem ke správné kontrole pravopisu (s výjimkou angličtiny) je to, aby kódování slovníku bylo shodné s kódováním dokumentu. Jelikož je český slovník pro `ispell` v kódování ISO8859-2, je při kontrole češtiny zapotřebí zapnout volbu Use input encoding. Ta zařídí, že se `ispell` bude spouštět s volbou `-T` (kódování dokumentu).

## LyX – hezky česky

Možností, které LyX nabízí, sice není málo, ale našince asi nejprve napadne, zdali lze tento nástroj „zneužít“ i k sazbě českých textů a zdali s národním prostředím umí pracovat i samotný LyX. Samozřejmě že ano – tento text budiž toho důkazem. Zde bych se rád zmínil o úskalích, které počestění LyXu provází. Tento proces se zpravidla sestává z následující posloupnosti kroků:

- 1) lokalizace L<sup>A</sup>T<sub>E</sub>Xu;
- 2) lokalizace LyXu;
  - (a) nastavení fontů pro zobrazování;
  - (b) volba mapování klávesnice;
  - (c) lokalizace nabídky.

**ad. 1)** Ačkoli je tento text převážně o LyXu, počestění L<sup>A</sup>T<sub>E</sub>Xu je s ním úzce spjata, proto se nejprve zabýváme právě jím. Precizní řešení tohoto problému jsou předmětem obsáhlých diskusí na webových fórech na nejrůznější úrovni, proto bych se zde do podobných záležitostí velmi nerad pouštěl. Tím spíše, že je tento text určen především začátečníkům. Pokusme se alespoň nastínit souvislosti, které bezprostředně souvisí s LyXem. Existují v podstatě tři způsoby počestění L<sup>A</sup>T<sub>E</sub>Xu:

- makra T<sub>E</sub>Xu pro sazbu akcentovaných znaků;
- mezinárodní distribuce L<sup>A</sup>T<sub>E</sub>Xu s lokalizačním balíčkem `babel`;
- L<sup>A</sup>T<sub>E</sub>X včetně L<sup>A</sup>T<sub>E</sub>X fontů.

Výhoda první varianty spočívá v tom, že pro zápis akcentovaných znaků se používá T<sub>E</sub>Xových maker, a tudíž je tento mechanismus principiálně dostupný z kterékoli jeho nadstavby. V případě, že zdrojový kód obsahuje příliš mnoho diakritických znamének, vyžaduje psaní v tomto formátu notnou dávku zručnosti, o čitelnosti ani nemluvě. Jako odstrašující příklad uvádím všem dobře známou větu „Příliš žluťoučký kůň úpěl ďábelské ódy“, která by ve výše zmíněné notaci vypadala následovně:

P\{r}\{\i}li\{s} \{z}lu\{t}ou\{c}k\{y} k\{r}u\{n}
 \{u}p\{e}l \{d}\{a}belsk\{e} \{o}dy.

Je zřejmé, že tudy cesta pravděpodobně nevede.

Výhoda druhé varianty spočívá v její nezávislosti na lokalizaci, neboť balíček `babel` je součástí standardní distribuce  $\text{\LaTeX}$ u. Omezení „babelizovaného“  $\text{\LaTeX}$ u dříve spočívalo údajně v částečně nedokonalém slovníku dělení slov. Od dob, kdy `babel` obsahuje slovník Pavla Ševečka, by ale tomuto problému mělo být učiněno zadost. Mezi slabiny mezinárodní distribuce  $\text{\LaTeX}$ u (z hlediska počestění) patří program na tvorbu rejstříků `makeindex`, který zjevně neřadí hesla správně podle české abecedy (řešení nalezne čtenář na str. 92). Dále pak fonty, které nemají ošetřeny kerningové páry pro písmena »d« a »t«.

Naproti tomu  $\text{\LaTeX}$  s  $\text{\LaTeX}$ fonty tyto nedostatky sice řeší, ovšem na úkor kompatibility vzhledem ke standardnímu  $\text{\LaTeX}$ u. Tento nedostatek je trnem v oku zejména lidem, kteří dbají na absolutní uniformitu  $\text{\LaTeX}$ u případně nadstaveb z něho odvozených. Osobně takovým problémům však nepřikládám větší váhu. Krom již zmiňovaných kerningů je předností  $\text{\LaTeX}$ fontů správné posazení naboděnic nad českými písmeny. Příkladem těchto úprav je např. posazení háčku nad písmenem »c« na střed obloučku namísto střed písmene atd. Zatímco typografové a návrháři písma vždy nekompromisně trvají na svém a v takových věcech jsou zcela neoblomní, oku „průměrného“ čtenáře jsou tyto nuance při standardní velikosti písma 10 bodů naštěstí ukryty (s výjimkou oněch nešťastných kerningů).

Problém počestění výsledné sazby padá výhradně na vrub  $\text{\LaTeX}$ u. Ostatně  $\text{\LaTeX}$ u je z tohoto pohledu upřímně jedno, jakým způsobem bude zdrojový kód zpracován. Jeho úkolem je pouze tvorba zdrojového kódu, nikoli však jeho překlad.  $\text{\LaTeX}$  implicitně používá babelizovaný  $\text{\LaTeX}$ . Pokud dáváte přednost jinému formátu (zejména  $\text{\LaTeX}$ u), můžete jeho jméno specifikovat v menu `Edit/Preferences/Converters`.

**ad. 2a)** Dalším krokem k české lokalizaci je změna obrazkových fontů.  $\text{\LaTeX}$  implicitně používá pro zobrazování fonty v kódování ISO-8859-1. Pokud chcete, aby se v editačním okně  $\text{\LaTeX}$ u české znaky zobrazovaly správně, je třeba typ kódování změnit na ISO-8859-2. Nastavení se provádí na záložce `Screen Fonts` (podle potřeby taktéž na záložce `Interface`)<sup>12</sup> v menu `Edit/Preferences/Look & Feel`.

Taktéž je třeba změnit formát stránky z `USletter` na `A4` a vybrat správné uvozovky. Implicitní formát papíru (`Default paper size`) lze také nastavit z menu `Edit/Preferences/Outputs/Misc`. V češtině se přednostně používají dvojité uvozovky typu 99-66, které začínají dole a končí nahoře. Veškeré tyto změny lze provést interaktivně na záložce `Language` z menu, `Layout/Document`. Podle potřeby

<sup>12</sup>I když preferujete anglické rozhraní, je nutné kvůli správnému zobrazení znaků spell-checkeru, pro `Interface` taktéž nastavit ISO-8859-2.

můžete toto nastavení uložit jako implicitní příkazem `Layout/Save Layout as Default`.

**ad. 2b)** Dále je třeba zavést českou klávesnici. Podle unixové filozofie by ani aplikace (LyX) ani window-manger neměl vůbec do tohoto procesu zasahovat. Jelikož většina window-managerů přebírá mapování klávesnice z X-serveru, stačí mít správně nastavenou klávesnici právě zde, a pak lze v LyXu psát bez problémů i hezky česky. Výjimku snad tvoří window-manager KDE spolu s některými klony Linuxu, který díky své koncepci mezinárodní klávesnice může procesem – stisk klávesy / zjevení se písmene na obrazovce monitoru, ještě pěkně zamíchat.

Nabízí se tedy dvojí řešení: buď KDE definitivně odeberat moc nad mapováním klávesnice (a ponechat ho výhradně v kompetenci X-serveru), nebo změnu mapování provést na úrovni aplikace (LyXu). Příkladně v konfiguraci Linux Red-Hat 7.3/KDE EN (kterou používám) se podařilo definitivně odstranit problémy spojené s mapováním klávesnice teprve až s příchodem KDE verze 3. V LyXu postaveném na xforms však (při změně klávesnice z window-manageru) potíže (na některých místech) přetrvávají, díky nefunkčnosti tzv. „mrtvých kláves“.

Krom toho LyX zde (částečně) doplňuje funkci X-serveru pro psaní abeced, které nejsou postavené na latině. Na základě těchto zkušeností spíše doporučuji provádět změnu mapování klávesnice na úrovni aplikace, poněvadž to by mělo podle všeho fungovat všude stejně. Proto se následující komentář bude týkat právě jeho.

Mimo to, mezi neoddiskutovatelné výhody mapování klávesnice na úrovni LyXu patří také to, že v menu a matematickém prostředí LyX automaticky přepne klávesnici na anglickou, což ocení zejména ti, kteří používají pro psaní matematiky příkazy  $\LaTeX$ .

Nastavení mapování se provádí v dialogovém okně `Edit/Preferences/Lang Opts/Language`; volba mapování, pak kombinací kláves `Alt-k-1` – primární, `Alt-k-2` – sekundární, případně `Alt-k-t` – prohození. Při psaní anglických textů však může toto mapování působit potíže, proto je lepší ho implicitně nechat vypnuté a podle potřeby ho zapínat kombinací kláves `Alt-k-x` resp. `Alt-k-o`.

Upřímně řečeno, problematika kolem mapování klávesnice v LyXu je pro mě zastřena trochou tajemství, proto je třeba odzkoušet, co je pro danou variantu nejlepší. Sami vývojáři v tomto směru cítí vůči uživatelům LyXu jistý závazek, a tak je vcelku strohý popis v dokumentaci zakončen příslibem, že budoucí verze manuálu bude v tomto ohledu více sdílná.

Obecně lze říci, že novější verze LyXu umožňují změnit většinu nastavení v dialogu `Edit/Preferences`. V případě, že je nějaká volba z toho menu neproveditelná, nebo není dostupná (ve starších verzích LyXu – viz nastavení spellcheckeru), lze provést změnu přímo v souboru `~/lyx/preferences` (dříve `~/lyx/lyxrc`), kde jsou uloženy informace o aktuálním nastavení. Některé změny v souboru `preferences` se ale po novém spuštění LyXu nemusí projevit, poněvadž pokud

L<sup>A</sup>T<sub>E</sub>X tento soubor nenajde, vytvoří si ho sám na základě nastavení v menu Edit/Preferences. Podle potřeby taktéž provede automaticky konverzi z `lyxrc` na `preferences`. Chcete-li ale jednoduše ověřit, zdali kroky vedoucí k počestění byly uloženy, měly by se v něm objevit následující řádky:<sup>13</sup>

- změna formátu na  $\text{\LaTeX}$   
`\latex_command cslatex`
- nastavení obrazkových fontů  
`\screen_font_encoding iso8859-2`  
`\screen_font_encoding_menu iso8859-2`
- nastavení formátu papíru  
`\default_papersize "a4"`  
po exportu se ve zdrojovém kódu objeví  
`\usepackage{a4}`  
a v případě použití volby Use Geometry Package  
`\geometry{verbose,a4paper}`
- nastavení mapování klávesnice  
`\kbmap true`  
`\kbmap_primary "czech"14`  
`\kbmap_secondary "american"`

**ad. 2c)** Ti, kdo nevládnou příliš anglickým jazykem nebo je jim zkratka české prostředí bližší, mohou L<sup>A</sup>T<sub>E</sub>X naučit bezzbytku česky. Dokumentace (která je určena spíše vývojářům, nežli uživatelům) taktéž obsahuje instrukce pro lokalizaci L<sup>A</sup>T<sub>E</sub>Xu do mnoha dalších jazyků. Zatím poslední distribuce L<sup>A</sup>T<sub>E</sub>Xu zahrnuje bezmála 14 jazyků, včetně čínštiny, japonštiny, korejštiny, hebrejštiny a arabštiny. Lokalizaci je možno provést pouze tehdy, nebyl-li L<sup>A</sup>T<sub>E</sub>X před kompilací zkonfigurován volbou "`--disable-nls`" (*national language support*). Pak lze jeho prostředí počestit nastavením globální proměnné `LANG` na příslušnou hodnotu. Pro terminály třídy `sh` například příkazem

```
export LANG=cs_CZ
```

V tomto případě musí být nastaveny fonty ISO-8859-2 i pro rozhraní a menu, Edit/Preferences/Look & Feel/Interface/Popup Encoding. Počestění skrze nastavení proměnné `LANG` ale není stoprocentní – dojde pouze ke změně dialogů, případně jejich vybraných položek. Položky menu zůstanou nedotčeny. Bohužel ani verze 1.2.0 nedoznala v tomto směru přílišných změn. Jelikož nepatřím mezi příznivce počestřování prostředí softwaru, s touto možností příliš velké zkušenosti

---

<sup>13</sup>Soubor nutně nemusí obsahovat vše. Záleží na rozsahu provedených změn a verzi L<sup>A</sup>T<sub>E</sub>Xu.

<sup>14</sup>Podle potřeby `\kbmap_primary "czech-qwerty"`, který lze vytvořit modifikací souboru `/usr/share/lyx/kbd/czech.kmap`.

nemám. Tuto volbu jsem zkusil provést pouze pro účely psaní tohoto článku. Tuším, že z webu se kdysi dalo stáhnout i lokalizované menu. Prehistorická verze údajně 100% počestěného LyXu je k mání na [13]. Detaily lze dohledat v Help/Customization a dále pak na webu.

## LyXtriky

V této sekci bych rád zmínil některé užitečné rady a tipy, které jsou výsledkem dlouhodobějších zkušeností a které mohou nejednomu uživateli LyXu ušetřit pár šedých vlasů. Tyto rady sice rámcově zapadají do předchozích sekcí, ale na druhou stranu mají řadu společných rysů, což mne motivovalo k tomu, abych je zahrnul do samostatné sekce. Obdobnou filozofii sleduje i koncepce dokumentace k LyXu (a k mnoha jiným programům). Ta bývá zpravidla rozdělena na část *základní* a *rozšířenou*. Základní část je zaměřena především na obecný popis systému a jeho vlastností, zatímco rozšířená část pojednává o specifických možnostech, vyplývajících z aplikace programu na konkrétní účel.

V souvislosti s LyXtriky je nezbytné zmínit odkaz na nevyčerpatelnou pokladnici rad a tipů Herberta Vosse [1, Tips and Tricks] a dále pak na obrovské diskusní fórum `lyx-users@lists.lyx.org`. Myslím, že nebudu přehánět, když řeknu, že 99 % problémů lze překlenout s pomocí těchto informačních zdrojů.

## Preambule

Charakteristikou každého autora je určitý styl sazby. Tento styl se zpravidla dokument od dokumentu nemění. Jednak proto, že k tomu není důvod, ale také proto, že se tím autor identifikuje (často bohužel i ve špatném světle). Z tohoto pohledu může být velmi žádoucí vytvořit si k již předdefinovaným stylům (book, article, ...) své vlastní, které obsahují určité odchylky a specifika vyplývající z možností a definic příslušného stylu.

Kupříkladu jednotná preambule, která obsahuje čítače pro číslování rovnic odvislé od kapitoly, zcela pozbývá význam v kontextu se stylem `article`.<sup>15</sup> Vytvoření univerzální preambule, kde bude zapotřebí při změně třídy dokumentu zakomentovat resp. odkomentovat několik řádek, se nezdá býti příliš vhodným řešením a navíc působí značně nepřehledně. Proto je dobré vytvořit několik souborů s preambulemi a ty pak skrze příkaz `\input` zavést do okna pro vložení preambule: `Layout/LaTeX Preamble`.

V dílčích souborech (preambulích) jsou zahrnuta specifika, případně rozšíření, které se pojí s danou třídou dokumentu. Odkazy na preambule lze pak podle potřeby jednoduše odkomentovat resp. zakomentovat. Výhodné je soubory

---

<sup>15</sup>Ve stylu `article` kapitola totiž definována není.

s preambulemi zahrnout do samostatného adresáře, který je společný všem dokumentům. Následující příklad snad nepotřebuje komentář.

```
\input{/home/user/preambles/my_article.tex}
%\input{/home/user/preambles/my_book.tex}
%\input{/home/user/preambles/my_slides.tex}
%\input{/home/user/preambles/my_seminar.tex}
:
:
```

Pokud nechcete preambuli  $\text{L}\text{X}\text{e}\text{L}\text{A}\text{T}\text{E}\text{X}$  (a další nastavení) vytvářet s novým dokumentem znovu, nezapomeňte nastavení uložit příkazem `Layout/Save Layout as Default`.

**Poznámka:** Pokročilejší uživatelé  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u jistě z důvodu zachování programátorské kultury (dá-li se to tak nazvat) vytvoří preambule jako stylové balíčky, které se zavedou pomocí příkazu `\usepackage`.

## $\text{L}\text{X}\text{e}\text{L}\text{A}\text{T}\text{E}\text{X}$ a $\text{Bib}\text{T}\text{E}\text{X}$

Práce s  $\text{Bib}\text{T}\text{E}\text{X}$ em není žádný trik. Jelikož si řada uživatelů odvykla číst uživatelské manuály, stala se ze samozřejmosti často kladená otázka. V místě výčtu citací nevolte styl `Bibliography`, ale vložte odkaz na soubor s příponou `bib` skrze menu `Insert/Lists & TOC/BibTeX Reference` (příponu do editačního pole nepiště). Do pole `Style` lze zadat specifický styl, přesně řečeno jméno souboru s příponou `bst` (bez přípony).

**Poznámka:** Standardní  $\text{Bib}\text{T}\text{E}\text{X}$  neumí pracovat s „tagy“, které obsahují akcentované znaky, proto nelze použít reference jako `[Kočička]`, ale pouze formy typu `[Kocicka]`, `[Koc00']`, případně číselný tvar `[7]`. Tento nedostatek nelze bohužel obejít ani trikem s  $\text{T}\text{E}\text{X}$ ovými makry: `Ko\{v\{c\}i\{v\{c\}ka}`. V samotných citacích se diakritika pochopitelně objevit může. Alternativou budiž program  $\text{C}\text{S}\text{Bib}\text{T}\text{E}\text{X}$ , který je zapotřebí šikovně  $\text{L}\text{X}\text{e}\text{L}\text{A}\text{T}\text{E}\text{X}$ u podstrčit (podobně jako program `makeindex`, viz řešení ze str. 92).

## Velké závorky

Pro standardní typy závorek, (tj. ty, které začínají a končí stejným ohraničením) lze použít horké klávesy v podobě `Alt-m- $\{$` . Příkladně kombinace `Alt-m- $\{$`  vytvoří ohraničení typu `{ }`. Obdobně pro `[ ]`, `( )`, `| |`. V ostatních případech je zapotřebí využít paletku pro sazbu matematiky. Zde je velmi vtipné pro nastavení otevírací a uzavírací závorky použít levého resp. pravého tlačítka myši.

## Individuální doladění umístění plovoucích prostředí

Pokud jste donuceni řešit i takové věci, jako je vzhled sazby, a zároveň patříte mezi sazeče – začátečníky, tak tuto podsekcí raději přeskočte a umístění

plovoucích prostředí (*floats*) ponechte výhradně v kompetenci L<sup>A</sup>T<sub>E</sub>Xu. Pokud se považujete za čtenáře s estetickým cítěním, tak směle pokračujte ve čtení následujícího textu.

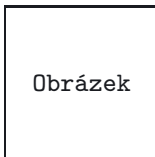
Cílem každého kvalitního sazecího programu je bezesporu snaha o jistou uniformitu a dodržení jednotných pravidel při sazbě dokumentu. Příkladem budiž umisťování obrázků. Pokud není specifikováno jinak, L<sup>A</sup>T<sub>E</sub>X přebírá implicitní nastavení L<sup>A</sup>T<sub>E</sub>Xu v pořadí t, b, p. To znamená, že L<sup>A</sup>T<sub>E</sub>X se nejprve pokusí umístit obrázek na horní okraj stránky, nepodaří-li se mu to, tak obrázek zkusí umístit na spodní okraj a v případě, že se mu nepodaří ani to, tak obrázek umístí na samostatnou stránku. Pořadí těchto voleb lze v L<sup>A</sup>T<sub>E</sub>Xu dle libosti zaměňovat skrze menu Layout/Document/Extra – okénko Float Placement. Sem lze samozřejmě umístit i volbu h. V tom případě se L<sup>A</sup>T<sub>E</sub>X pokusí umístit obrázek v místě, kde byl skutečně vložen. Každá z výše uvedených voleb může být navíc doplněna o symbol !, kterým lze u daného parametru zvýšit prioritu. Příkladně volba h! téměř s jistotou způsobí, že obrázek bude umístěn v místě vložení, a to bez ohledu na styl stránky, čemuž se zákonitě musí podřídít lámání okolního textu.

Pokud je obrázek vložen v polovině odstavce nebo stránky, pak taková volba způsobí při čtení jistou diskontinuitu, která mnohdy působí více než rušivě. Proto je zapotřebí vždy podobný krok velmi pečlivě zvážit. Je-li použito pouze globální pravidlo, může se stát, že určitý obrázek se díky tomuto nastavení umístí na místo, které k tomu není příliš vhodné, nebo které neúměrným způsobem naruší formátování textu.

V L<sup>A</sup>T<sub>E</sub>Xu se tento problém dá vyřešit vcelku snadno tím, že u konkrétního prostředí parametr umístění změním na požadovanou hodnotu. Toto dílčí (lokální) nastavení má vyšší prioritu než nastavení globální, a proto se umístění plovoucího objektu projeví pouze na tomto místě, přičemž pravidlo pro ostatní objekty zůstane zachováno.

Starší verze L<sup>A</sup>T<sub>E</sub>Xu tuto možnost individuálního doladění bohužel nenabízejí, takže změna globálních pravidel pochopitelně způsobí kompletní reorganizaci textu. V případě dílčích nastavení je tedy třeba objekt (obrázek, tabulku) obalit příkazy prostředí figure resp. table s modifikovaným parametrem například takto:

```
\begin{figure}[b!]
```



```
\end{figure}
```

Analogicky lze samozřejmě provést tuto změnu pro tabulky. V LyXu 1.2.0 lze již provést individuální doladění floatu z kontextového dialogu.

## Tvorba průsvitných fólií

Jednou z mnoha možností LyXu je také tvorba průsvitných fólií (*slides*). K tomu účelu lze použít buď styl `slides`, nebo bohatší `seminar` z nabídky `Layout/Document/Document/Class`. Pokud je však použit formát papíru A4, LyXu se to moc líbit nebude. Proto mezi velmi frekventované dotazy patří otázky typu „Proč tato volba LyXu vadí?“ a „Co s tím?“. Odpověď na první otázku bohužel neznám. Řešení té druhé spočívá v těchto krocích: V menu `Layout/Document/Paper` změňte `Papersize` z A4 na `default`. Klepněte na záložku `Document` a do okénka `Extra options` vložte parametr `a4`. Více parametrů se odděluje čárkou. Pokud jste LyXu dovolili upravit třídu dokumentu na `seminar`, tak by v onom okénku mělo být `slidesec`, `a4`. Je-li zároveň nastavena orientace naležato (*landscape*), promítnou se výše zmíněná nastavení do argumentu příkazu `documentclass` následujícím způsobem

```
\documentclass[landscape,slidesec,a4]{seminar}
```

V menu `Edit/Preferences/Outputs/Misc` ale musí být pro `Default paper size` taktéž nastaveno A4.

**Poznámka:** Některé starší verze GhostScriptu parametrům LyXu nerozumí, proto se může stát, že je výsledný dokument o 90° otočený, nebo nemá náležitý formát papíru.

Často ale chceme hotové fólie převést do formátu PDF, proto se někdy stane, že konvertor `dvips`, případně `ps2pdf` parametry formátu papíru nerozporná a okraje fólií jsou úplně jinde, než bychom si představovali. Obzvláště při použití rámečků to nevypadá příliš pěkně. Nejlepších výsledků se mi podařilo docílit dle výše uvedeného nastavení spolu s doladěním velikosti orámování

```
\addtolength\slidewidth{\šířka rámečku}  
\addtolength\slideheight{\výška rámečku}
```

případně

```
\renewcommand{\slidetopmargin}{(horní okraj)}  
\renewcommand{\slidebottommargin}{(spodní okraj)}  
\renewcommand{\slideleftmargin}{(levý okraj)}  
\renewcommand{\sliderightmargin}{(pravý okraj)}
```

Jelikož jsou rozměry stránky shodné s formátem A4, nečiní pak tisk fólií žádné potíže. Výše uvedený popis považuji z hlediska LyXu za dostačující. Pokud chcete vaši prezentaci vylepšit o další „vymoženosti“, nezbude než nahlédnout do dokumentace k balíčku `seminar`. Mohu-li mluvit za sebe, pak za nevyčerpatelnou studnici kouzel a triků považuji např. [12].



Ačkoli demonstrační soubory obsahují „kousky“, o kterých se vám doposud možná ani nesnilo, je třeba zdůraznit, že se jedná opravdu pouze o ukázky, které se na co nejmenším prostoru snaží zkombinovat co nejvíce možností. Při tvorbě standardní prezentace doporučuji spíše střídmost. Úsloví, že někdy „méně znamená více“, zde platí dvojnásob. Každopádně máte-li přebytek volného času a bohatou fantazii, můžete takto vaše představy relativně snadno zhmotnit. Pokud jste v opačné situaci, přejděte rovnou do sekce Denise Giroua – Seminar FAQ.<sup>16</sup>

Díky cenové dostupnosti datových projektorů jsou v poslední době kromě klasických fólií velmi žádané nástroje pro tvorbu prezentací promítaných přímo z obrazovky počítače. Takto vytvořená prezentace nejen že oplývá řadou prostředků a vymožeností, jako jsou barvy, animace, hyperlinky apod., ale také se snadno modifikují za cenu takřka zanedbatelných nákladů. Tím pochopitelně odpadá složitý a nákladný proces výroby průsvitných fólií, který byl dříve při přípravě nebo změně obsahu prezentace nevyhnutelný.

Příkladně pořizovací cena barevných fólií byla tak vysoká, že někteří tvůrci se „neštítili“ provádět aktualizaci ručně pomocí fixů – jiní tak bezostyšně činili i bez ohledu na peníze. Do problematiky a zejména pak estetiky tvorby počítačových prezentací se zde raději pouštět nebudu. Ostatně nevhodné a necitlivé kombinace barev a dalších prostředků nechtě jsou vizitkou jejich tvůrce. Myslím, že žlutý obrázek na bílém pozadí, případně černý text na tmavě modrém pozadí mluví za vše.

Nepočítáme-li velmi omezenou skupinku nadšenců, hrdě se hlásící k myšlence „udělej si sám“, pocházela většina programů na tvorbu počítačových prezentací výhradně z komerční sféry. S tímto faktem se ale rozhodně odmítli smířit tvůrci programů s otevřeným zdrojovým kódem. Zejména pak s nástupem pdfLaTeXu se balíčky k tomuto účelu začaly objevovat jako „houby po dešti“.

Mezi nejznámější patrně patří `pdfscreen`. Pro LyX existuje i modifikovaná verze `pdfscreen2`, včetně layoutu [4] (o layoutech bude pojednáno dále). Jeho zakomponování do LyXu se však neobejde úplně bez problémů, obzvláště tehdy, je-li dokument psán jiným jazykem, než anglicky.

Chcete-li přesto tento nástroj pro tvorbu prezentací používat, je třeba postupovat dle následujících kroků: zkopírujte soubor `pdfscreen.layout` do adresáře s layouty.<sup>17</sup> Proveďte rekonfiguraci LyXu – Edit/Reconfigure. Známkou úspěšného začlenění layoutu by mělo být zjevení se položky `slides (pdfscreen)` v menu `Layout/Document/Document/Class`. Je-li zároveň použit `babel`, je třeba ho úplně vypnout – do souboru `preferences` je zapotřebí přidat řádek

```
\language_use_babel "false"
```

Pokud tak neučiníte, bude `pdfscreen2` v místě výskytu proměnné `english` (kterou `babel` používá) hlásit chybu. Chcete-li navíc svoji prezentaci vylepšit o ovlá-

---

<sup>16</sup>Úplně nejlepší je postscriptová verze s ukázkami.

<sup>17</sup>Na unixových platformách obvykle `~/lyx/layouts`.

dací panel s tlačítky, případně zavést speciální volby balíčku `pdfscreen`, je třeba toto nastavení zahrnout do preambule layoutu a znovu rekonfigurovat `LyX`. Příklad těchto úprav může vypadat následovně (implicitní volba je zakomentována).

```

:
%\usepackage[screen,nopanel]{pdfscreen2}
\usepackage{xspace,colortbl}
\usepackage[screen,panellleft,gray,paneltoc]{pdfscreen2}
\margins{.75in}{.75in}{.75in}{.75in}
\screensize{6.25in}{8in}
:
```

Pro tvorbu počítačových prezentací (à la PowerPoint) lze také s výhodou použít třídu `prosper` (která je údajně založena na `pdfscreen`) a pro kterou existuje i layout [6]. Jelikož třída `prosper` není součástí standardní distribuce `LATEX`u, je zapotřebí ji doinstalovat [5]. Proces začlenění layoutu je velmi kvalitně popsán v [1, Tips and Tricks]. Některých speciálních efektů (jako např. prolínání obrazů), je zde docíleno pomocí javascriptových maker. Z volně dostupných prohlížečů podporuje tento formát snad jen Acrobat (Reader) fy Adobe. Každopádně PDF musí být vytvořeno způsobem `LATEX`→`dvips`→`ps2pdf`.

## Problémy s kontrolou pravopisu

Pokud píšete více jazyky, může při použití (vlastního) uživatelského slovníku dojít ke kolizi s jazykem dokumentu a tudíž i ke „zmatení“ spellcheckeru. Proto je třeba dát pozor na to, aby byl jazyk uživatelského slovníku shodný s jazykem dokumentu. V opačném případě vám totiž `LyX` kontrolu pravopisu vůbec neumožní. Ačkoli jste `LyX`em upozorněni na to, že vámi zvolený slovník (nalézající se v `/usr/lib/ispell`) vůbec neexistuje, je obvykle tato chyba zapříčiněna právě kolizí (existujícího) slovníku s jazykem dokumentu.

Tento rozpor lze jednoduše odstranit v menu `Edit/Preferences/Lang Opts/Spell checker`. Zkontrolujte, zdali pole `Use personal dictionary` skutečně obsahuje správný slovník, případně tuto volbu zcela vypněte. Rozumným řešením je vytvoření více uživatelských slovníků – pro každý jazyk jeden – a ten pak zvolit skrze volbu `Use personal dictionary`. Implicitním jazykem pro spellchecker je jazyk dokumentu (zvolený skrze menu `Layout/Document/Language`). Pokud se volba jazyka liší od jména slovníku (např. `german` vs `deutsch`), můžete tento rozpor odstranit skrze pole `Use alternative language` (viz menu `Edit/Preferences/Lang Opts/Spell checker`).

## Jak na vakáty?

Mezi jisté nešvary L<sup>A</sup>T<sub>E</sub>Xu patří sazba záhlaví (*headings*) na zcela prázdnou stránku (tzv. vakát), která mnohdy formálně k předchozí kapitole či sekci vůbec nenáleží, nebo zde záhlaví působí rušivým dojmem. Momentálně si ani nedokáží vybavit makro, které by uvedený nedostatek jednoduchým způsobem řešilo. Vsa-  
dím se, že programátor takového makra by zde byl s největší pravděpodobností  
přinucen využít možností všemocného T<sub>E</sub>Xu, což se od regulérního uživatele  
L<sup>A</sup>Xu dá očekávat jen stěží.

Abych se inspiroval tím, jak je tento nedostatek řešen v technické literatuře  
renomovaných nakladatelství, s překvapením jsem zjistil, že v žádné knize se ni-  
kde taková stránka nenachází. Jednak proto, že kapitola je často sázena i na su-  
dou stranu (a musím říci, že to vůbec nevypadá špatně), nebo jednoduše proto, že  
práci provedl profesionální sazeč, kterému „zázrakem“ vyjde vždy text na celou  
stranu. V L<sup>A</sup>Xu se dá výše uvedený problém řešit pouze metodou „hrubé síly“,  
tzn. před stránkou provést stránkový zlom a na stránku pak umístit L<sup>A</sup>T<sub>E</sub>Xový  
příkaz

```
\thispagestyle{<styl>}
```

Pokud je takových stránek více (např. v úvodní části dokumentu), je dobré  
styl záhlaví vypnout: `Layout/Document/Document`, volbu `Pagestyle` nastavit na  
`empty` a teprve na požadované straně tuto volbu zapnout příkazem

```
\pagestyle{<styl>}
```

Proměnná `<styl>` může nabývat hodnot `empty` (bez hlavičky/patičky), `plain`  
(pouze číslo strany), `headings` (zkrácené záhlaví) a `fancy` (úplné záhlaví s lin-  
kou).

## Obrázek jako template z Xfigu – změna velikosti

Změna velikosti EPS obrázků zpravidla nečiní žádné potíže. Velikost lze zadat  
jak v absolutních, tak v relativních jednotkách. Mezi často kladené dotazy však  
také patří otázka, jak tohoto efektu docílit u obrázků s vnořeným textem typu  
*special*. Tento úkon ale podobně jako v prvním případě nepředstavuje nic ob-  
tížného. Většina lidí však o této možnosti neví, poněvadž se neobtěžují čtením  
doprovodných instrukcí L<sup>A</sup>Xu.

Změnu velikosti externího objektu lze provést již při vkládání objektu, nebo  
dodatečně, klepnutím na referenci s obrázkem (zpravidla `[XFig: obrázek]`) a při-  
dáním parametru `-m` (*magnification*) do okénka `Parameters`. Konkrétně parametr  
`-m .75` způsobí zmenšení na 75 %.

Mnohdy ale bývá lepší se této volbě se zcela vyhnout a obrázek vytvořit  
v měřítku vhodném k začlenění do dokumentu, neboť s tímto parametrem dojde  
pochopitelně i ke změně velikosti vloženého textu, čímž se naruší již dříve zmí-  
něná uniformita. Této volby je globálně použito např. ve stylu `seminar`, kde ze

zcela zřejmých důvodů dojde ke zvětšení veškerého obsahu, včetně písma dokumentu. Zde je naopak tento efekt žádoucí, poněvadž takto vytvořené materiály (zejména k prezentačním účelům) vykazují nesrovnatelně větší čitelnost.

## LyX pro pokročilé

Než se pustím do velmi zevrubného popisu možností, které využijí zejména pokročilejší uživatelé, chtěl bych nejprve uvést pár věcí na pravou míru.

Bez nadsázky lze říci, že tato sekce by vydala na samostatný článek, minimálně stejného rozsahu jako tento. Přiznám se, že i já jevim neskrývaný zájem o takové informace, poněvadž ani dokumentace k LyXu v tomto směru příliš sdílná není. Zároveň bych rád podotknul, že v oblasti programování příliš velké zkušenosti nemám, proto bych se zde nerad pouštěl na „tenký led”.

Smyslem této sekce není poskytnout čtenáři takové množství informací, na základě kterých by byl schopen LyX přetvořit k obrazu svému, ale pouze přiblížit tento nástroj ve světle úprav, které jsou uživatelům komerčních produktů zcela zapovězeny. Čtenáři s nadprůměrnou počítačovou gramotností nechtě posílí řady vývojářů LyX Teamu. Myslím, že „pouhé” zpracování dokumentace by v tomto směru bylo více než záslužné. Nejprve ale něco málo i pro „obyčejné smrtelníky”.

## Tvorba L<sup>A</sup>T<sub>E</sub>Xových maker

O matematické sazbě bylo zevrubně pojednáno v předchozích sekcích. Jedním z nástrojů, který při tomto procesu přispívá velmi výraznou měrou ke zefektivnění práce, jsou makra. Popis tvorby maker je celkem obstojně popsán v helpu, proto se jím zde zabývat nebudu. Makra mohou (ale nemusejí) obsahovat argumenty.

Při exportu do L<sup>A</sup>T<sub>E</sub>Xu LyX produkuje příkaz `\newcommand`, proto maximální počet argumentů je odvozen právě od tohoto příkazu (tuším, že devět). Makra je užitečné vytvářet zejména pro funkce resp. jejich kombinace, pro které neexistuje klávesová zkratka. Tímto způsobem lze obejít někdy až zbytečně zdlouhavé „proklikání“ se k požadovanému výrazu skrze menu, nebo z paletky pro zadávání nestandardních znaků.

Podotýkám, že se jedná o L<sup>A</sup>T<sub>E</sub>Xová makra v matematickém prostředí. Samotný LyX bohužel (zatím) žádným makrojazykem (v pravém slova smyslu) nedisponuje. S touto možností se ale počítá u budoucích verzí.

## Multipart dokument

Pokud jste z nějakého důvodu donuceni k psaní díla obzvláště velkého rozsahu, možná byste měli zvážit možnost vytvoření tzv. „složeného“ (*multipart*) dokumentu. Méně výkonným počítačům se tak výrazně odlehčí a navíc dokument

se stane mnohem přehlednější, poněvadž při psaní „operujete“ pouze v aktuální části a tudíž není zapotřebí při každé změně (dílní části) překládat celý několikasetstránkový dokument neustále dokola.

Jednotlivé dokumenty se pak sloučí pomocí jediného (*master*), který obsahuje odkazy na všechny dílní soubory včetně dalších komponent. Podmínkou je, aby všechny dokumenty měly stejnou třídu (*document class*). Začlenění podřadných souborů do „masteru“ se provádí skrze menu **Insert/Include File**. Před překladem masteru  $\text{L}\text{Y}\text{X}$  nejprve provede export všech dílních dokumentů (součástí). Ve zdrojovém kódu masteru se pak v místě začlenění podřadného dokumentu objeví  $\text{L}\text{A}\text{T}\text{E}\text{X}$ ový příkaz

```
\input{<jméno části.tex>}
```

V dialogu zvolte jeden ze způsobů **Use include**, nebo **Use input**, podle toho, má-li se dokument začlenit na novou stránku resp. přímo v místě vložení.

Menu **Insert/Cross Reference** umožňuje odkazovat se na návěští a reference podřízených dokumentů a naopak. Více informací lze nalézt v **Help/Extended Features**.

## Tvorba stylových souborů $\text{L}\text{Y}\text{X}$ u

Velmi lákavou možností, kterou  $\text{L}\text{Y}\text{X}$  nabízí pokročilejším uživatelům, je tvorba a definice nových stylových souborů (*layouts*).<sup>18</sup> Tvorba layoutu se skládá ze dvou kroků:

- 1) definice stylu (zarovnání, odsazení, řez písma atd.) v prostředí  $\text{L}\text{Y}\text{X}$ u (pro zobrazení);
- 2) definice maker  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u (pro sazbu).

Jelikož definice zobrazovacího stylu nemá přímý dopad na vysázený text, je spíše na uvážení autora, jak bude styl v editačním okně  $\text{L}\text{Y}\text{X}$ u koncipován. Každopádně se jedná spíše o pohodlí při práci s  $\text{L}\text{Y}\text{X}$ em. Obecně může být totiž použit styl, který v layoutu již existuje. Např. vzhled prostředí **Description** může být (vizuálně) shodný s nově definovaným stylem **Remark**. Byť je pozadí obou prostředí (z hlediska definice  $\text{L}\text{A}\text{T}\text{E}\text{X}$ ových maker) zcela odlišné, rozdíl mezi těmito dvěma styly pak uživatel  $\text{L}\text{Y}\text{X}$ u pozná na první pohled jen těžší.

Předtím, než se pustíte do tvorby vlastního layoutu, je dobré se přesvědčit, zdali něco podobného už někdo neudělal před vámi. Z webu lze totiž stáhnout celou řadu specifických layoutů, charakteristické určitými specifiky, případně upravené pro potřeby různých organizací (např. AMS, AAS, IEEE a mnoha dalších). Kupříkladu pro psaní tohoto textu jsem použil layout Matěje Cepla pro publikování článků ve zpravodaji  $\mathcal{C}\mathcal{S}\mathcal{T}\mathcal{U}\mathcal{G}$ .

---

<sup>18</sup>Kvůli kolizi se stylovými soubory  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u (s příponou *sty*), budu v dalším textu používat termín *layout*.

Ačkoli tvorba vlastního layoutu může navenek působit složitě, ve skutečnosti to zas tak složitě není. Jednak je to velmi obstojně popsáno v dokumentaci, ale velkou inspirací jsou především hotové layouty, které případným zájemcům doporučuji ke studiu. Často bývá velmi užitečné vytvořit nový layout mírnou modifikací již stávajícího. Příkladně, je-li váš článek ryze technického charakteru, nepředpokládá se, že v něm někde uplatníte prostředí pro psaní veršů. Nevyužité prostředí můžeme předefinovat a využít ho k mnohem užitečnějším účelům – např. jako prostředí pro sazbu očíslovaných definic, příkladů apod.

Vizuální podobu dokumentu lze také vysázenému textu vdechnout pouze pomocí příkazů a maker  $\text{\LaTeX}$ u. Tímto způsobem je možno přetvořit základní styly  $\text{\LaTeX}$ u (kapitola, sekce, atd.) tak, že nezůstane kámen na kameni.  $\text{\LaTeX}$  tím samozřejmě své kouzlo nijak neztrácí, poněvadž zde hraje výhradně roli vizualizačního nástroje určeného ke značkování.

Tento elegantní způsob jsem využil např. při psaní diplomové práce, aniž bych se tím připravil o výhody, které  $\text{\LaTeX}$  nabízí, přičemž výsledný vzhled byl souměřitelný s některými publikacemi renomovaných nakladatelství. Mezi tyto úpravy patří např. sazba iniciál, esteticky řešené oddělení jednotlivých částí (*parts*) včetně nadpisů, piktogramy a grafické emblémy, jimiž je uvozen text nadměrné důležitosti, prostředí pro tvorbu helpu jazyka `MATLAB` a spousta dalších.

Způsob redefinice  $\text{\LaTeX}$ ových maker je (narozdíl od změny layoutu) méně pracný, ale pro nová prostředí je nepoužitelný. Vždy je tedy zapotřebí zvážit, zdali se zásah na úrovni layoutu vyplatí. Byť lze skrze menu stylů procházet kombinací kláves `Alt-P-Space`, osobně se domnívám, že stoprocentní účinnosti se docílí pouze tehdy, existují-li k nově definovaným prostředím i samostatné horké klávesy.

## **$\text{\LaTeX}$ a hladká sazba?!**

Ne! Nadpis této sekce není v žádném případě pokus o zmatení čtenáře (který se již v úvodu dočetl, že  $\text{\LaTeX}$  se žádným způsobem na sazbě textu nepodílí). Význam těchto slov (nebo spíše slovní hříčky) nespočívá v „hladkosti“ sazby jako takové, nýbrž v hladkosti při práci s  $\text{\LaTeX}$ em. V této sekci bych rád upozornil na některé často se vyskytující chyby, převážně zapříčiněné uživatelskou nevědomostí. Tyto chyby totiž velmi často pramení z toho, že se uživatelé – začátečníci snaží  $\text{\LaTeX}$  „znásilnit“ k návykům, které si osvojili během práce s konvenčními textovými editory (WYSIWYG).

Podle sebe vím, že člověk se novým věcem učí vcelku rychle, zato však s odvykáním je to už horší. A tak nezbyvá než doufat, že následující řádky budou pro čtenáře náležitou „odvykací kúrou“. Ostatní nechť okusí, jak chutná sazečské řemeslo na vlastní kůži.

## Oddělování odstavců

Mezi typické prvky laické sazby patří oddělování odstavců „umělým“ vložením jednoho nebo více prázdných řádek. Tento zlozvyk patrně ještě přežívá z éry psacích strojů. V počítačové sazbě ale žádné opodstatnění nemá – spíše naopak. Autor textu by měl pouze označkovat začátek a konec odstavce. O tom jak, budou odstavce odděleny rozhodne sazeč skrze nastavení sázecího programu. Opravami, případně dalším zásahem do textu může totiž odstavec s prázdným řádkem „přetéci“ na další stranu. Situace, kdy se autor textu snažil sazeči ušetřit práci, pak paradoxně vyústí v opak. Odstranění takových chyb se bohužel nedá zautomatizovat, proto je třeba provést ho ručně. Ať se budete snažit sebevíc, L<sup>A</sup>T<sub>E</sub>X vám vložení prázdného řádku opětovným stiskem klávesy Enter neumožní. Někteří uživatelé si zde však vypomáhají vložením znaku pro řádkový zlom tj. stiskem kláves Ctrl-Enter.

To, že L<sup>A</sup>T<sub>E</sub>X nereaguje na opakovaný úhoz do klávesy Enter, by mělo uživatele od tohoto způsobu „lámání přes koleno“ odradit. V lepším případě by to měl být signál, že něco není v pořádku. Jednoduše, autoři L<sup>A</sup>T<sub>E</sub>Xu tímto krokem sledovali jistý záměr, proto se ho raději držte – v opačném případě by se vám to rozhodně nemuselo vyplatit.

Způsobů tvorby odstavců zná typografie hned několik [7, 8]. L<sup>A</sup>T<sub>E</sub>X standardně nabízí pouze dva. První varianta je typičtější pro českou literaturu a používá odstavcové zarážky. S druhou se lze častěji setkat v anglosaské literatuře – místo odstavcové zarážky odděluje odstavce prázdným řádkem. Rozhodnete-li se pro první variantu, doporučuji vám nahlédnout do některé z knih o typografii, kde najdete pravidla pro tvorbu odstavců, řešení jejich východových řádků apod.

Je zřejmé, že pokud budete odstavce oddělovat předepsaným způsobem, můžete mezi oběma volbami libovolně přepínat, aniž by tím vznikly nějaké potíže. Přepnutí se provádí skrze menu Layout/Document/Document. V oblasti Separation zvolte Indent (s odrážkami), nebo Skip (s prázdnými řádky).

## Tvrdé mezery

Podobným prohrěškem je i opomenutí tzv. „tvrdých“, neboli nezlomitelných mezer, na které je čeština opravdu bohatá, zejména pak díky jednoslabičným spojkám a předložkám I, i, K, k, O, o, S, s, U, u, V, v, z, Z, které se nesmí vyskytnout na konci řádky. Další pravidla související s vkládáním tvrdých mezer je třeba nastudovat z příslušné literatury. V L<sup>A</sup>T<sub>E</sub>Xu tento úkon představuje pouhý stisk kláves Ctrl-Space.

Automatizované ošetření tvrdých mezer (včetně těch českých) lze provést např. aplikací programu Petra Olšáka v1na [14]. Tento program bohužel nelze aplikovat přímo na dokument L<sup>A</sup>T<sub>E</sub>Xu (i když jeho rozšíření by asi bylo přímočaré), nýbrž až na výsledný (vyexportovaný) zdrojový kód. Tento otrocký postup se dá vcelku snadno zmechanizovat pomocí „chytrého“ skriptu. Vzor takového skriptu

(za který vřele děkuji Matěji Ceplovi) vypadá následovně:

```
#!/bin/sh
LATEX='latex'
ZAKLAD=$(basename $1 .tex)
grep -q '^\\usepackage{czech}' < $ZAKLAD.tex
IS_ON=$? # vrací 0 pokud je to cslatex
if [ $IS_ON != '0' ]
  then LATEX='cslatex'
fi
if [[ $(basename $0) == "pdfvlnlatex" ]]
  then PROG="pdf$LATEX"
else
  PROG=$LATEX
fi
if [ -f $ZAKLAD.tex ]; then
  # LyX je trochu moc aktivní s přidáváním závorek
  sed -e 's/\(^|[\^?!]\)\{\}' '/\1' /g' $ZAKLAD.tex \
  | vlna -s -v KkSsVvZzOoUuA§ -f > $ZAKLAD.ctex
  $PROG $ZAKLAD.ctex
fi
```

Spolu se skriptem (nesoucím jméno `vlnlatex`) je třeba zároveň vytvořit symbolický link `pdfvlnlatex`, který na tento skript ukazuje. Určitou nadějí v tomto směru je i příslib, že budoucí verze `LyX`u budou obsahovat skutečný makrojazyk, díky němuž bude možno podobné úkony vcelku snadno řešit přímo.

### Prostředí `array` vs `equation`

Mezi časté chyby při sazbě matematiky patří použití šablony z paletky pro tvorbu matic místo šablony pro soustavy rovnic, ačkoli podtext obou prostředí je zcela odlišný (`array` pro matice a `equation` pro rovnice). Pokud rovnice obsahuje velké symboly (integrace, sumace, ...), vysázejí se tyto symboly v prostředí `array` jako v psaném textu (ne jako samostatně vysázená rovnice) včetně integračních resp. sumačních mezí.

## Má `LyX` také nedostatky?

Samozřejmě. Každá věc má svoje dobré a špatné stránky. Ani `LyX` není v tomto ohledu výjimkou. Zde se ve stručnosti zmíním o těch, které mi často zneprůjemňují život. Jistě přijdete i na řadu dalších.



## Dlouhé rovnice

Při psaní obzvláště dlouhých rovnic se může stát, že rovnice „přeteče“ přes editační prostředí LyXu. Zpravidla platí, že rovnice, která přesahuje rámec (maximalizovaného) editačního okna, přesáhne taktéž zrcadlo sazby a tudíž je třeba rovnici vhodně rozdělit. Tím je uživatel k rozdělení rovnice explicitně donucen. Jiná situace ale nastane tehdy, obsahuje-li rovnice výrazy, pro které LyX nemá vlastní znaky (fonty) např.  $\triangleleft$  a které nelze zapsat jinak, než pomocí L<sup>A</sup>T<sub>E</sub>Xového příkazu. Obsahuje-li daná rovnice těchto výrazů hned několik, může se stát, že šířka prostředí pro psaní matematiky přesáhne (maximalizované) okno samotného LyXu. To by tak zdaleka ještě nevadilo, pokud by bylo možné „dosáhnout“ i na text, který odcestoval kamsi za rámeček monitoru.

Například v Xfigu je to ošetřeno tak, že pokud text přesáhne šíři editačního okénka, objeví se pod ním „scrollbar“, kterým lze textem v okně pohybovat. V LyXu bohužel nic podobného není. S tímto fenoménem se zpravidla potýkají majitelé monitorů s nízkým rozlišením. Jste-li tedy postaveni před podobný problém, nezbude než přepsat celý výraz pomocí zdrojového kódu, nebo pro něj vytvořit substituci (zkratku) pomocí příkazu `\newcommand`. Tím se ale samozřejmě celé kouzlo LyXu naráz vytratí.

Naštěstí takových rovnic nebývá mnoho. Možná právě proto, tato chyba nikoho tolik netrápí. Zároveň je třeba dodat, že problematika lámání rovnic je velmi složitá a nedá se ošetřit nějakým uživatelským makrem. Její (částečné) řešení vede na zásahy do nejhlubších „vnitřností“ T<sub>E</sub>Xu, což vyžaduje excelentní znalosti tohoto systému. Ze známých zdrojů zabývající se touto problematikou lze doporučit např. [9].

## Tabulky s pevnou šíří buňky

Další chyba souvisí s tvorbou tabulek. Uživatel může při tvorbě tabulky volit z několika možností: ohraničení a sloučení buněk, zarovnání a rotace textu, ubírat a přidávat řádky/sloupce atd. Vše pouze z kontextového dialogu. Obstojnou tabulku tedy hravě zvládne i člověk, který o prostředí `table` nemá nejmenší ponětí. Problémy ovšem nastanou tehdy, obsahuje-li tabulka buňky s pevně definovanou šířkou. Pravda, tento postup se příliš neslučuje s myšlenkou „structured authoring“, ale je dobré na něj upozornit.

Potíž je v tom, že při novém otevření dokumentu se tabulka obsahující pevně definovanou šířku sloupce, „smrskne“ na šíři dvou znaků, čímž zákonitě musí dojít k jejímu „natažení“ na výšku. Podle objemu textu může délka (natažené) tabulky přesahovat i několik obrazovek. Kliknutím do prostoru příslušné buňky se tabulka opět poslušně přeformátuje do správného rozměru, takže zabírá pouze nezbytně nutné místo. Je tedy jasné, že pokud dokument obsahuje větší počet tabulek (s pevně definovanou šíří sloupce) je třeba „proklikat“ všechny takové tabulky, jinak se délka dokumentu až několikanásobně zvětší. Tento úkon jistě

k efektivitě práce nijak nepřispívá – spíše naopak. Verze 1.2.0 je naštěstí této chyby zbavena.

### Aktualizace obrázků s textem typu special

Jisté problémy také provází vložení obrázků s textem typu *special*, resp. jejich aktualizace. Máte-li zároveň otevřeno okno s Xfigem, ve kterém provádíte úpravy a prohlížeč `xdvi`, ve kterém by se po rekompilaci  $\LaTeX$ em měly provedené změny projevit, zjistíte s největší pravděpodobností, že se tak nestalo. Přesně řečeno vlastní obrázek se aktualizuje, ale vložený text nikoli. Proto je třeba před kompilací vložit (nejlépe před obrázek) kus textu – stačí i jedno písmeno.

### Omezení šablony pro začlenění obrázků Xfigu

Možnost začlenění obrázku s textem typu *special* pomocí šablony (*template*) bohužel nelze využít tehdy, obsahuje-li dokument Xfigu nějaký další (externí) objekt. Nejčastěji tímto objektem zpravidla bývá graf, který se díky možnostem Xfigu snažíme obohatit o nejrůznější popisky apod.

I přesto, že je grafický postprocessing, kterým dnes disponuje většina počítačových algebraických systémů na relativně vysoké úrovni,<sup>19</sup> jsme často postaveni před úkol tento úkon provést dodatečně za pomoci některého vhodnějšího nástroje (Xfig). Pak nezbude, než export z Xfigu do obou částí provést ručně – Combined PS/LaTeX (both parts) a do  $\text{LyX}$ u pak skrze menu Insert/Include file vložit část s příponou `pstex.t`. V dialogu je zapotřebí zvolit Use input. Z tohoto pohledu není zcela jasné, jestli je na vině  $\text{LyX}$  (resp. template psaný v jazyce python), nebo program `transfig`, který obstarává export z Xfigu.

### Netriviální změna programu na tvorbu rejstříku

Mezi velmi palčivé problémy související s počestěním  $\text{LyX}$ u (přesněji řečeno jeho výstupu) patří tvorba rejstříku (*index*). Tento nedostatek sice není chyba v pravém slova smyslu, ale problémů s ním spojených je více než dost.  $\text{LyX}$  si při tvorbě rejstříku automaticky zavolá na pomoc program `makeindex`, který je zahrnut ve standardní distribuci  $\LaTeX$ u. Celý proces tvorby rejstříku je plně zmechanizován, takže uživateli  $\text{LyX}$ u stačí pouze „ukázat prstem“ na místo, kde se má rejstřík vysázet a  $\text{LyX}$  vše ostatní obstará sám. Někdy může ale být toto zmechanizování na závadu, poněvadž uživateli není dovoleno do tohoto procesu zasáhnout.

Opět narážíme na omezení interaktivních systémů. Hlavní potíž je ale v tom, že `makeindex` neumí dobře pracovat s akcentovanými znaky, ale hlavně nezná

---

<sup>19</sup>Příkladně `MATLAB` umožňuje vkládat speciální znaky přímo do obrázku pomocí  $\text{TeX}$ ových maker.

písmeno »ch«, které je pravděpodobně českou specialitou. Všechny tyto nedostatky řeší program `csindex`, který je obdobou již zmíněného `makeindex` a který je součástí distribuce `CSLATEX`. Úkol tedy spočívá v tom, jak k používání `csindex` přinutit `LYX`.

Při použití nestandardního formátu lze některé problémy obejít využitím `\latex_command`. Ovšem obdoba tohoto příkazu ve stylu `\index_command` bohužel neexistuje. Podobné potíže nastanou tehdy, potřebujeme-li vysázet rejstřík v nestandardním stylu např. s tučnými verzálkami uvozující skupinu hesel stejného počátečního písmene. V [1, Tips and Tricks] jsou uvedena dvě řešení – bohužel ani jedno z nich se mi nezdá příliš elegantní. Tak tedy velmi stručně.

První varianta je v souladu s metodou „hrubé síly“. Vstupní soubor pro `makeindex` resp. `csindex` vygenerovat s pomocí `LYX` a vlastní setřídění provést z příkazové řádky se zahrnutím všech voleb (stylových souborů). Pak znovu přeložit zdroj z příkazové řádky. Je zřejmé, že tento otrocký postup se musí provést zvlášť pro každý dokument a konečně i tehdy, dojde-li v již hotovém rejstříku k nějakým změnám.

Druhá varianta je o poznání generičtější, ale vyžaduje více úsilí. Její podstata spočívá ve vytvoření jakéhosi „bypass“ souboru, který je třeba `LYX` vhodně podstrčit. V adresáři `~/bin` je zapotřebí vytvořit symbolický odkaz s názvem `makeindex` na „bypass“ skript,<sup>20</sup> ve kterém lze provádět změny dle libosti. Odkaz na skript musí být pochopitelně předřazen v cestě všem systémovým souborům (standardně v `/usr/bin/`), aby se provedl přednostně. Podobně jako při vkládání tvrdých mezer, i zde existuje špatné a ještě horší řešení.

Za ideální bych považoval stav, kdy pouhá ikona v místě vložení rejstříku s nápisem `Index` na sebe převezme formu tlačítka, po jehož stisknutí na uživatele vyskočí dialog (obdobný `BibTeX`), ve kterém bude možno specifikovat styl rejstříku, případně další volby. Výše popsáný problém nemusí být problémem pouze česky píšících uživatelů, neboť krom již zmíněného `makeindex` existují i další programy jako např. `xindy`.

### Vícejazyčný spellchecking

Následující problém není ani tak chyba, jako spíše nedostatek. `LYX` umožňuje (individuálně) nastavit u označeného bloku textu jiný jazyk, než je jazyk dokumentu. Tím je zaručeno, že se pro konkrétní jazyk použije správný slovník dělení slov. Stranou tohoto nastavení však zůstane kontrola pravopisu, poněvadž spellchecker neumí přepínat slovník v závislosti na tomto atributu. Nemám nejmenší tušení, jak pracné by bylo tohoto efektu docílit, ale prosby uživatelů (zatím) vývojáři nevyslyšeli.

---

<sup>20</sup>Template skriptu lze stáhnout např. z [1, Tips and Tricks].

Nefunkčnost „mrtvých kláves“

Závěrem této sekce bych dále poukázal na již dříve zmiňovaný problém související s nastavením dialogu spellcheckeru, poněvadž se svým způsobem také jedná o chybu, a proto formálně náleží do této sekce. Tentokrát ale není na vině  $\text{L}_\text{Y}\text{X}$ , nýbrž „konstelace“ softwarového prostředí, které někdy neumožňuje vepisovat do dialogů  $\text{L}_\text{Y}\text{X}$ u české znaky (např. v RedHat 7.2/KDE 2.2 EN).

Psaní češtiny v menu leckdy ani nemá opodstatnění (např. při tvorbě návštěví), ale je-li třeba takto zapsaný text i vysázet, pak není zbytí. S tímto případem se můžeme setkat například v bibliografických citacích. Často se místo čísla reference uvádí jméno autora např. [Kočička]. Pokud jméno obsahuje akcentované znaky, mělo by se tak skutečně vysázet – je zřejmé že Kočička a Kocicka nemusí být jeden a ten samý autor. Tento nedostatek lze sice obejít užitím  $\text{T}_\text{E}\text{X}$ ové notace pro zápis akcentovaných znaků (zde je díky krátkému slovu akceptovatelný), ale ve skutečnosti toto řešení není příliš šťastné.

## Epilog

Byť je rozsah tohoto článku na první pohled větší, než bývá zpravidla zvykem, neklade si v žádném případě nároky na úplnost. Na jednu stranu si uvědomuji, že řada lidí je mnohdy v časové tísní a nemá čas pročitat se rozsáhlými materiály (sám jsem často do takové situace uvržen), ale na druhou stranu povrchní čtení zpravidla nepřináší žádné výsledky, proto je mnohdy takový přístup spíše ztrátou času.

Jak bylo demonstrováno na předchozích příkladech (viz sazba matematiky), psaní textu v samotném  $\text{T}_\text{E}\text{X}$ u se v podstatě rovná programování ve vyšším programovacím jazyku, jehož syntaxe velmi vzdáleně připomíná HTML kód – ortodoxní uživatelé  $\text{T}_\text{E}\text{X}$ u nechť prominou toto macešské přirovnání. Pokud byste se chtěli naučit dovést ovládání tohoto unikátního systému k absolutní dokonalosti a využívat tak všech jeho možností, tzn. aby  $\text{T}_\text{E}\text{X}$  sloužil vám a nikoli naopak, budete se muset pravděpodobně smířit s tím, že tomuto procesu zasvětíte několik let života a ani tak vám asi nikdo nezaručí, že se vám to skutečně podaří.

Nechci, aby tyto řádky vyzněly až příliš pesimisticky, ostatně nejsnazší bude, když si práci s  $\text{T}_\text{E}\text{X}$ em vyzkoušíte sami, ale na druhou stranu si myslím, že problematiku sazby nelze zúžit pouze na znalost  $\text{T}_\text{E}\text{X}$ u. Sazeč, který jako prostředek ke své práci používá počítač, musí být také dokonale seznámen s písmovými a grafickými formáty.

O objemu těchto znalostí mimo jiné vypovídá fakt, že tento obor obnáší několik let studia na vysoké škole. Nechám tedy na uvážení každého z vás, jaká míra znalostí je pro konkrétní záměr ještě únosná. Těžko můžeme po laikovi vyžadovat výtvar srovnatelný s dílem renomovaného nakladatelství či reprografického studia.

Obávám se, že na volání po návratu sazečských nástrojů do ruku profesionálů je dnes už pozdě. Laicizace těchto nástrojů je trend, který lze asi jen stěží zvrátit. Proto je zapotřebí, aby tyto nástroje splňovaly alespoň jisté (přijatelné) minimum. Zkrátka a dobře, potřebujete-li sednout a psát namísto učení se programovacím technikám a dosud jste s žádným podobným systémem nepracovali, myslím, že budete vděční, že něco takového jako je  $\text{L}\text{Y}\text{X}$  vůbec existuje. V případě, že dáváte přednost interaktivnímu přístupu před dávkovým způsobem sazby, pak  $\text{L}\text{Y}\text{X}$  je v tomto směru asi jediný způsob, jak využívat možností  $\text{T}\text{E}\text{X}$ u bez nutnosti znalosti jeho programovacího jazyka (pochopitelně jen do určité omezené míry). Pokud jsem pro  $\text{T}\text{E}\text{X}$  zvolil přirovnání ke kódu HTML, pak  $\text{L}\text{Y}\text{X}$ u by nejspíše náleželo přirovnání k programům typu Netscape Composer.<sup>21</sup>

Možná jste se někdy setkali s názorem, že webové stránky tvoří v Composeru jen hlupáci, zatímco profesionálům se jejich výtvar rodí přímo z příkazové řádky. Řada programátorů, jejichž koníček je (bohužel nebo bohudík) i jejich živobytím, však tvrdí, že se vizualizačních a interaktivních nástrojů při práci nijak neštítí. Ba naopak. Ukázalo se, že za účelem zrychlení a zefektivnění práce je velmi výhodné připravit si hrubé rozvržení dokumentu včetně základních prvků za pomoci některého z těchto nástrojů. Teprve ve druhé fázi, ladění detailů a implementování speciálních požadavků zákazníka nastupuje na scénu individuální programátorské umění.

V podstatě stejnou analogii lze vysledovat i z  $\text{L}\text{Y}\text{X}$ u. Jak jsem se snažil ukázat na vybraných příkladech, může být práce v  $\text{L}\text{Y}\text{X}$ u nesrovnatelně efektivnější než při psaní standardním způsobem. V prvním přiblížení je  $\text{L}\text{Y}\text{X}$  určen především začátečníkům, ale ob stojí i v profesionální sazbě. Jeho všestrannost a variabilita je dost široká na to, aby uspokojila požadavky i náročnějších uživatelů. Některé z nich jsem v tomto článku ani nezmínil. Určitou pozornost by si pravděpodobně zasloužil popis modulu  $\text{reL}\text{Y}\text{X}$  určený k importu  $\text{L}\text{A}\text{T}\text{E}\text{X}$ ových zdrojů do  $\text{L}\text{Y}\text{X}$ u. Zpočátku jsem byl k tomuto nástroji poněkud skeptický, ale záhy jsem zjistil, jak překvapivě dává výsledky.

Z hlediska poskytovaných možností je  $\text{L}\text{Y}\text{X}$  zcela souměřitelný s komerčními programy tohoto typu a v některých vlastnostech je dokonce i předčí. Možná právě proto, že se jeho vývojáři rekrutují převážně z řad dobrovolníků, pro které je  $\text{L}\text{Y}\text{X}$  zejména koníčkem a ne zaměstnáním. Bohužel ani tento program nezůstal ušetřen chyb – ostatně komerční programy také ne. Na některé z nich jsem se zde snažil poukázat, případně navrhnout jejich řešení. Každopádně o šíři záběru  $\text{L}\text{Y}\text{X}$ u se již přesvědčil ne jeden uživatel.

Stejně bohatý je i výčet vlastností a vylepšení, které si vývojáři do budoucna vytyčili. Na druhou stranu je třeba dodat, že implementace všech funkcí  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u

---

<sup>21</sup> Rozumí se ve smyslu odstínění koncového uživatele od prezentační stránky věci tzv. „separation of structure and content“. V žádném případě tím nechci stavět do stejného světla kvalitu  $\text{L}\text{A}\text{T}\text{E}\text{X}$ u vytvořeného  $\text{L}\text{Y}\text{X}$ em, s kvalitou kódu HTML, který produkuje Composer.

Program	Verze
LyX	1.2.0
teTeX	1.0.7
TeX	(Web2C 7.3.1) 3.14159
LaTeX	LaTeX 2 <sub>ε</sub>
pdfLaTeX	(Web2C 7.3.3.1) 3.14159-0.14h-released-20010417
GhostScript	6.52
Xfig	3.2 patchlevel 3d (Protocol 3.2)

**Tab. 2:** Verze testovaných programů

do vizuální podoby nemá reálný smysl. Zásadní inovace pravděpodobně nastanou s příchodem již dlouho avizovaného LaTeXu3. Ačkoli LyX již dnes dokáže bez problémů pokrýt většinu požadavků, bude i nadále předmětem dalšího vývoje. Jak jsem se již přesvědčil, verze 1.2.0 opravdu nepředstavuje pouhou změnu verze „na druhém desetinném místě“, ale oproti svému předchůdci 1.1.6fix4 doznala velmi zásadních změn (věřím, že k lepšímu).

Daň, kterou za tento „komfort“ bude muset uživatel zaplatit, se jmenuje *kompatibilita*. Myslím, že tímto magickým slůvkem se může pochlubit snad jen samotný TeX, ovšem obavy některých autorů o reedici svých děl v horizontu padesáti a více let, mi připadají až příliš přehnané. Vývoj v této oblasti jde natolik rychle kupředu, že jsem byl nucen změnit obsah některých částí již během psaní tohoto textu (text byl původně psán pro 1.1.6fix4, ale jelikož věci kolem 1.2.0 dostaly poměrně rychlý spád, bylo zapotřebí poopravit řadu věcí). Samozřejmě není vyloučeno, že v momentě, kdy čtete tyto řádky, bude některým problémům učiněno za dost a některé další přirozeně přibudou. Pro úplnost jsou v Tab. 2 uvedeny verze testovaných programů.

Obrovský dík také patří Matěji Ceplovi a Petrovi Aubrechtovi za pedantickou revizi tohoto textu a věcné připomínky, které (doufám) přispěly ke zkvalitnění tohoto textu. Věřím tedy, že vám tento článek pomohl osvětlit některé problémy, ať už používáte LyX anebo jiný sázecí program, a že alespoň částečně obohatil vaše dosavadní znalosti.

## Odkazy

- [1] Oficiální stránky projektu LyX, <http://www.lyx.org>.
- [2] Český tutoriál k LyXu, <http://lubr.regionet.cz/index-en.html>.
- [3] SÝKORA J., Digitální rádiová komunikace II, *Vydavatelství ČVUT*, 1998 – použito pro demonstraci příkladů.
- [4] Balíček pdfscreen2 včetně layoutu, <http://www.math.tau.ac.il/~dekelts/lyx/pdfscreen.tar.gz>.

- [5] Balíček prosper, <http://prosper.sourceforge.net/>.
- [6] Layout prosper, <http://www.math.tau.ac.il/~dekelts/slides/>.
- [7] KOČIČKA P., BLAŽEK F., Praktická typografie, *Computer Press*, 1. vyd., 2000.
- [8] RYBIČKA J.,  $\LaTeX$  pro začátečníky, *Konvoj*, ISBN 80-85615-74-6, 1999.
- [9] DOWENS M., „Breaking Equations,“ *TUGboat*, vol. 18, no. 3, Proceedings of the Annual Meeting, 1997.
- [10] Stránky projektu tex2pdf, <http://tex2pdf.berlios.de/>.
- [11] High quality PDF output from  $\LaTeX$  and  $\TeX$ , <http://ltswww.epfl.ch/~dsanta/resources/type1>.
- [12] Stránky o balíčku seminar, <http://www.tug.org/applications/Seminar/>.
- [13] Česká lokalizace  $\text{LyX}$ , <ftp://ftp.fi.muni.cz/pub/linux/localization/>.
- [14] Program vlna, <ftp://math.feld.cvut.cz/pub/olsak/vlna>.
- [15] Stránky projektu *cygwin*, <http://sources.redhat.com/cygwin/>.

## Summary: $\text{LyX}$ – how to?

Although the exact title translation is “ $\text{LyX}$  – how to?”, I really discourage all readers who wish to skim this paper as a prompt cook-book. Actually I have never been thinkin’ of making either  $\text{LyX}$  Documentation or Tips & Tricks carbon-copy. There’s no doubt that the  $\text{LyX}$  Team, and many other contributors did an excelent job. The documentation to  $\text{LyX}$  itself is done exceedingly good, however I felt there are some gaps in the typographic knowledge remain to be filled. In fact this is the reason why I tried to develop the bridges between typographic art and particular typesetting system. By the way I just wanted to introduce  $\text{LyX}$  in a bright light and make its description interesting for both – the very beginners and regular  $\LaTeX$  users.

The content of the paper may be simply characterized by the message: “ $\text{Lyx}$  is not a toy for kids, but the powerfull tepesetting engine front-end”. Hope that this message is clear for all readers. The power of  $\text{LyX}$  has been demonstrated on many examples. Most of them raised from my own requirements which involve in technical typesetting and presentation tools creation. When using the  $\text{LyX}$  – this can be simply done at the same place. It was also shown that  $\text{LyX}$  can easily cover many of advanced users requirements and wide range of customizations. There is also a single section dedicated to the Czech localization. Even thought this paper was originally tailored to the  $\text{LyX}$  version 1.1.6fix4, some parts had been rewritten due to the version 1.2.0 quick release.

Ve dnech 29. 4. až 3. 5. 2002 se konala v Bachotku v polském vojvodství Brodnica konference EuroBach<sub>T</sub>E<sub>X</sub> 2002. Byla to konference tradiční a zároveň poněkud netradiční. Bach<sub>T</sub>E<sub>X</sub> se tu na samém začátku máje konal již podesáté. Euro<sub>T</sub>E<sub>X</sub>, navštěvující nejrůznější zákoutí Evropy, se zde dopracoval již do svého třináctého ročníku. Výsledkem tohoto zkřížení byla konference o něco bohatší, než bývá pro obě tyto akce obvyklé.

Většinu z triadevadesáti účastníků tvořili uživatelé <sub>T</sub>E<sub>X</sub>u z Polska, početné zastoupení měla také většina států Evropy. Nechyběli ani účastníci z Austrálie a Ameriky. Členové polského sdružení GUST zároveň na konferenci oslavili desetiletí existence této organizace.

Po příjezdu v neděli 29. 4. večer nás přivítala chatová osada na břehu jezera Bachotek. Dřevěné, skromně, leč útulně zařízené chatky skvěle zapadly do okolního lesa. Protože už se setmělo příliš, na pohled na obrovské jezero jsme si museli počkat do rána. Po dvanácti hodinách strávených v autě, z valné části na nepříliš udržovaných polských silnicích, nás stejně nejvíce ze všeho zajímalo, kde dostaneme večeri. Mimochodem, pokud si stěžujete na stav našich vozovek, zajedte si k našim severním sousedům. Nejen, že hlavní tah od Cieszyna na Gdaňsk vede většinou přes města, dokonce se na něm vyskytují i přechody pro chodce chráněné nejvyšší značkou snížené rychlosti. Zato dopravní policie je bdělá, po cestě jsme potkali více než deset hlídek sledujících provoz dalekohledy.

Ale zpět ke konferenci. Po tři dny nás čekaly přednášky v angličtině. V pondělí po oficiálním zahájení jako první přednášel Kees van der Laan o amatérech, profesionálech a BLUe. BLUe je makrobalík umožňující vkládání poznámek do souborů PDF pro usnadnění komunikace mezi autorem a redakcí a v neposlední řadě pro poznámky čtenářů. Následoval Péter Szabó s problematikou vkládání obrázků do <sub>T</sub>E<sub>X</sub>ových dokumentů. Péter diskutoval o vhodnosti použití různých formátů a demonstroval použití programu **sam2p**. Tento program umožňuje konvertovat obrázky různých formátů a optimalizovat je na velikost či rozlišení podle výstupního zařízení.

Włodzimierz Bzyl popsal své zkušenosti s vytvářením fontů pod operačním systémem Linux. Jako se již na evropských konferencích stalo tradicí, Jonathan Fine, typický konzervativní Angličan, předvedl novinky svého projektu interaktivního <sub>T</sub>E<sub>X</sub>u. Základní myšlenka spočívá v tom, že sázeční stroj běží jako démon a podle pokynů textového editoru přesazuje text třeba i po každém stisku klávesy. To přináší mnoho problémů k řešení, například jak se vypořádat s chybami ve zdrojovém kódu nebo jak zajistit, aby se textu nemuselo přesazovat příliš mnoho naráz.



Pokud se vám při programování maker stává, že byste se opravdu rádi podívali, co se vlastně uvnitř  $\text{T}_{\text{E}}\text{X}$ u děje, určitě by vás zaujal program Marcela Takače  $\text{DebugT}_{\text{E}}\text{X}$ . Ten to umožňuje velmi komfortně. Určitě by ho využil i pozorný posluchač přednášky Davida Kastrupa o pokročilých metodách programování  $\text{T}_{\text{E}}\text{X}$ u. A pokud potřebujete sázet bibliografie ve více jazycích, je tu Jean-Michel Hufflen se svým  $\text{MLBIBT}_{\text{E}}\text{X}$ em, který ovládá i XML. A pokud ani to není dost, tak si určitě — pokud zrovna nepotřebujete něco naprogramovat nebo dokončit materiály k přednášce — přijdete na své u večerního táboráku. Tedy sice počasí příliš nepřálo a docela intenzivní sprška nás předčasně rozehnala, ale zato jsme se záhy dozvěděli, že se táborák bude ještě opakovat.

Úterní program zahájil Petr Sojka tutoriálem na téma dělení slov. Podělil se o své dlouholeté zkušenosti s touto problematikou, zejména s generováním vzorů dělení pro komplikované speciální aplikace. O výrazně emotivnějším tématu promluvil Jerzy Ludwichowski v přednášce nazvané „Open Source: a lesson from history“ a samozřejmě vzbudil svým sugestivním projevem vášnivě diskuse. Zpět do světa sazby nás vrátila Blanca Mancilla a Omega nasazená v prostředí webového serveru s mapami se zvláštním důrazem na vícejazyčná uživatelská rozhraní.

Krzysztof Trojanowski informoval o 6. rámcovém programu v oblasti informačních technologiích. Techniky srovnávání různých typů fontů představil Karel Píška. Petr Sojka vystoupil již podruhé, tentokrát s návrhem na pečlivější zacházení se zdrojovými soubory pro generování vzorů dělení slov. V archivu CTAN nalezneme mnoho souborů se vzory, ale bohužel jen zřídka také data, ze kterých byly tyto vzory generovány. Cílem je vytvoření archivu těchto zdrojů, které by zvláště usnadnily práci v situacích, kdy se setkáme s problémem sazby lingvisticky komplikovaných textů.

O své metodě zpracování ruských textů promluvil Kees van der Laan. Kaveh Bazargan nás pozval na TUG 2002 v Indii a David Kastrup zahraničním účastníkům tento maraton zakončil představením preview- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u. V Emacsu vytvořil systém, který zobrazuje na žádost části textu (převážně matematické vzorce) vysázené mezi ostatním zdrojovým kódem, možná názorněji řečeno jako obrázky. Členům GUSTu však den ještě neskončil, čekala je valná hromada.

Středeční ráno zahájil David Antoš povídáním o generování vzorů pro segmentaci thajštiny. Spolu s Petrem Sojkou se snaží ukázat, že technika rozpoznávání vzorů je dostatečně obecná a že si zaslouží větší uplatnění než jen pro prosté dělení slov. Marcin Woliński připravil skvělý příspěvek se slibným názvem Poor Man's PowerPoint. To zajisté vyžaduje vysvětlení. Použil výstup  $\text{T}_{\text{E}}\text{X}$ u na konzolu při překladu jako prezentaci. Prezentace jsou sice textové, zato však na Linuxové konzole i barevné. Samozřejmě to nebylo míněno vážně, nicméně tato metoda ovládání konzoly vznikla z potřeby demonstrovat činnost makroexpanze. K vážnější problematice nás vrátili Gyöngyi Bujdosó a Ferenc Wettl, kteří popsali poměrně značné problémy s podporou maďarštiny.

Karel Skoupý představil návrh na unifikaci zpracování textu a grafiky pomocí integrace  $\text{T}_{\text{E}}\text{X}$ u a  $\text{METAPOST}$ u. Ukázal obecnější model zlomu odstavce a předvedl výsledek svého snažení z několika posledních dnů, nový primitiv do NTS (politicky korektní název je XTS, NTS s přejmenovanými několika soubory v souladu s platnou licenci), který umožňuje zadat tvar odstavce pomocí specifikace omezujících obdélníků, do kterých pak odstavec nateče. Kaveh Bazargan se podělil o zkušenosti z používání  $\text{T}_{\text{E}}\text{X}$ u pro akademické publikování a Karel Skoupý ještě přidal povídání o .NET, což je sice platforma, která se nám nemusí zcela líbit, ale jistě se nelze tvářit, že neexistuje.

Středa odpoledne se nesla ve znamení výletu do města Grudziądz. Prohlédli jsme si centrum města a zúčastnili se vernisáže výstavy „Twórczość fotograficzna i typograficzna Janiny i Zygfryda Gardzielewskich“. Večer jsme strávili v krčmě Czarci Młyn za vydatné podpory skvělé, v Polsku zaslouženě populární, kapely Blues Nightshift Band.

Ve čtvrtek a v pátek už přednáškový sál ovládla téměř zcela polština. Adam Kolany prezentoval zkušenosti s používáním Perlu při zpracování textů, Bogusław Lichoński porovnal  $\text{T}_{\text{E}}\text{X}$  se systémem MathMagic.

Hans Hagen opět přinesl novinky ze své praxe s použitím  $\text{T}_{\text{E}}\text{X}$ u ve velkých a komplikovaných projektech, dynamickém vytváření dokumentů a integraci grafiky. Předvedl několik velmi pěkných prací, u kterých se nechtělo příliš věřit, že je vytvořil  $\text{T}_{\text{E}}\text{X}$ em. Marcin Woliński, tentokrát už zcela seriózně, předvedl svou techniku pro optimalizaci velkých dokumentů uniformní struktury, konkrétně slovníku o několika stech stranách. Základní princip je v tom, že z logu z prvního běhu  $\text{T}_{\text{E}}\text{X}$ u se spočte potřeba změny odstavců pomocí `\looseness` a tyto úpravy se zavedou do zdrojového kódu.

Maciej Jan Chojnacki pohovořil o zásadách typografie ve vědeckých publikacích, Tomasz E. Kowalski o použití  $\text{T}_{\text{E}}\text{X}$ u v technologii klient-server a o výhodách, které tato metoda přináší z hlediska správy systému. Maciej Głowacki prezentoval nástroje pro práci s fonty v Linuxu a Włodzimierz J. Martin příspěvek s názvem „ $\text{T}_{\text{E}}\text{X}$  sklada słowa“, o kterém nemohu naprosto nic říci, protože na to moje (nevelká) schopnost rozumět polštině vůbec nestačila.

Na závěr Stanisław Wawrykiewicz a Martin Schröder informovali o stavu projektu  $\text{T}_{\text{E}}\text{X}$ Live7. Předpokládám, že v době, kdy budete tento článek číst, bude už toto vydání k dispozici. Přináší propracovanější koncepci popisu balíků a mnoho dalších vylepšení.

Čtvrteční večer zakončil (pro nepřízeň počasí opakovaný) táborák, tentokrát spojený i s tradičním pliváním ohně. Pohled na  $\text{T}_{\text{E}}\text{X}$ isty s dvoumetrovým plamenem u úst je jedním slovem fascinující.

Pátek zahájili Yannis Haralambous a John Plaice referátem o posledním vývoji systému Omega. Představili také svou koncepci dalšího pokroku tohoto konzervativního následníka  $\text{T}_{\text{E}}\text{X}$ u. Následoval Andrzej Tomaszewski, grafik, s přednáškou o návrhu loga. Probral stručně základní principy kompozice grafických

objektů a předvedl návrhy nového loga sdružení GUST, o kterém se pak také bouřlivě hlasovalo. Norbert Jankowski vyrábí z  $\LaTeX$ u HTML Help a Janusz Bień píše v  $\LaTeX$ u a Emacsu učební texty, nicméně o tyto příspěvky jsme se připravili, směřující zpět k rodným lánům.

Rád bych jménem všech účastníků z Čech, Moravy a Slovenska (mimochoodem, všichni přednášeli) poděkoval sdružení  $\zeta$ TUG za vydatnou finanční podporu jejich účasti na této konferenci. A kdy se znovu můžeme sejít? Pokud to nebude na TUGu v Indii, tak jistě na Euro $\TeX$ u 2003 v Brestu. Už se na to těším.

*David Antoš*  
*Fakulta informatiky Masarykovy university, Brno*  
*xantos@informatics.muni.cz*

---

---

## Tutorialy pro $\TeX$ ovské začátečníky

JIŘÍ MATYÁŠ

Není tomu tak dlouho, co jsem některé z Vás poprvé viděl. — 8. prosince 2001 se konalo valné shromáždění  $\zeta$ TUGu a mé jméno se objevilo i na kandidátce do nového výboru, který byl při této příležitosti po třech letech znovu volen. Hlavním záměrem mé kandidatury bylo zlepšení informovanosti „začátečníků“. Když jsem diskutoval s lidmi, kteří se věnují práci s výpočetní technikou (s programátory, uživateli, účetními, studenty i laiky), nabýval jsem dojmu, že by se měli informace o  $\TeX$ u rozšířit i za hranice univerzit a vydavatelství.

Jako student gymnázia se mimo to „všeobecně“ věnuji správě WWW-stránek a operačnímu systému Linux. Při této příležitosti jsem se před lety setkal i s  $\TeX$ em a byl jsem zcela očarován. Kabinet informatiky už tradičně podporuje snahu studentů zdokonalovat se ve svém „volném“ čase a díky tomu mají možnost tvořit a spravovat školní servery. Jako druhý, služebně nejstarší jsem se rozhodl vyhlásit seminář nazvaný *Typografický systém  $\TeX$  a OS Linux*. Při realizaci mi pomáhaly zkušenosti nabyté organizací přednášek přírodovědného charakteru. Velký zájem studentů o něco jiného než je Pascal mi dodal důvod dále studovat a řádně zpracovat pomocné výukové texty.

Diskuse uzavírající valné shromáždění řešila právě problém zaměření se na začátečníky. Byly projednány možnosti přístupu laiků-začátečníků k informacím a k samotnému  $\TeX$ u.

Když jsem se vracel domů, v klidu vlaku EuroCity jsem v polospánku přemýšlel nad harmonogramem a formou vedení přednášek: Jednalo by se o jednodenní

„tutorialky“ či o jednorázovou akci typu S<sub>L</sub>T 2001? Druhá varianta je velmi náročná finančně i časově. A co problém s výběrem místa či města? Je možné počítat s Plzní? Vždyť pan ing. Brejcha, ředitel vydavatelství ZCU, nám nabízí technické zázemí. Nebo snad okolí Prahy? Vždyť sem jezdí spoje z celé republiky (i ze Slovenska). A co Skalský Dvůr u Bystrice nad Pernštejnem v oblasti krásné Českomoravské vysočiny? Tam se už přece jeden ročník konal. Uvidíme . . . !

Při řešení prvního problému je nutné rozhodnout, zda využít zázemí univerzit, či se zkusit vnutit do podvědomí knihoven a středních škol. Jak jsem již prezentoval, mělo by se mi podařit zajistit učebnu pro deset lidí na Gymnáziu Vídeňská v Brně, kde působím a mám zde pod kontrolou technické zázemí. Vyplatilo by se také využít sálu na Fakultě informatiky MU v Brně, ZCU v Plzni nebo JCU v Českých Budějovicích – jsou přece kolektivními členy  $\zeta$ STUGu. U této „začátečnické“ varianty jsem schopen z 99 % reprezentovat  $\zeta$ STUG v souladu s bodem č. 4 odstavce *Činnost sdružení stanov*.

V mém záměru je tedy dát dohromady sborníček s tématy přednášek na jeden den. Aby se účastníci mohli seznámit s historií, typografickými poučkami, vlastnostmi distribuce T<sub>E</sub>X Live, s makry plainT<sub>E</sub>X a L<sup>A</sup>T<sub>E</sub>X.

Předběžný plán přednášek s časovým rozvrhem je uveden níže. K tomuto plánu je možné vznášet připomínky.

### Pracovní plán:

1. Úvod přednášek, seznámení s $\zeta$ STUGem . . . . .	20 min
2. Možné OS (lehké seznámení s OS Linux, Windows) . . . . .	50 min
3. Základní charakteristika T <sub>E</sub> Xu (verze, distribuce, CTAN, T <sub>E</sub> X Live) . . . . .	20 min
4. Užití plainT <sub>E</sub> Xu a L <sup>A</sup> T <sub>E</sub> Xu . . . . .	15 min
5. Úvod do typografie (užití písma v praxi, hladká sazba, okraje, . . . ) . . . . .	30 min
6. Jednoduchý dokument v plainT <sub>E</sub> Xu (nadpisy, odstavce, výčty, . . . ) . . . . .	90 min
7. Jednoduchý dokument v L <sup>A</sup> T <sub>E</sub> Xu (textová prostředí) . . . . .	90 min
8. Sazba obrázků . . . . .	40 min
9. Matematická sazba . . . . .	30 min
10. Tabulky . . . . .	30 min
11. Speciality, informace, zdroje, diskuse, shrnutí . . . . .	30 min
<b>Celkem:</b> . . . . .	<b>445 min</b>
	tj. cca. 7,5 hodin à 60 minut
	tj. cca. 10 hodin à 45 minut
	Časové údaje jsou předběžné.

*Jiří Matyáš*  
jmatyas@gvid.cz

---

---

# International T<sub>E</sub>X Users Meeting EuroT<sub>E</sub>X'2003, ENST de Bretagne, Brest, France

---

## 14th European T<sub>E</sub>X Conference

### Brest, June 24–27, 2003

EuroT<sub>E</sub>X Conferences are held every year (since 1987) in a different European country: previous conferences took place in Karlsruhe (Germany), Cork (Ireland), Paris, Praha, Birmingham (UK), Gdansk (Poland), Arnhem (The Netherlands), Moscow, Saint-Malo (France), Heidelberg (Germany), Oxford (UK), Kerkrade (The Netherlands), Bachotek (Poland).

In 2003, the EuroT<sub>E</sub>X will be organized in Brest (Brittany, France), by ENST de Bretagne.

The theme of this conference covers the electronic document, digital typography, typographical heritage and its survival in the computer era, new technologies applied to education, book production process, fonts, document internationalization and exchange, etc.

Talks will be held in English, but articles for the Proceedings can be in any European language. The proceedings of the conference will be published in the TUGboat journal, before the fall of 2003.

### Important Dates

<b>Jan, 7 2003:</b>	send in abstract for paper
<b>Jan, 31 2003:</b>	notification of acceptance of paper
<b>March, 15 2003:</b>	send first version of full paper
<b>May, 15 2003:</b>	send final version of full paper
<b>June, 24-27 2003:</b>	EuroT <sub>E</sub> X conference in Brest

### Programme Committee

Jacques André (IRISA, Rennes)  
Hans Hagen (Pragma ADE)  
Yannis Haralambous (ENST de Bretagne)  
Alan Hoenig (City University of New York)  
Ronan Keryell (ENST de Bretagne)  
Frank Mittelbach (EDS, Mainz)

John Plaice (UNSW, Sydney)  
Sebastian Rahtz (Oxford University CS)  
David Salomon (California State University)  
Petr Sojka (Masaryk University, Brno)

### **Organizing Committee**

Odile Ely (ENST de Bretagne)  
Yannis Haralambous (ENST de Bretagne)  
Ioannis Kanellos (ENST de Bretagne)  
Ronan Keryell (ENST de Bretagne)  
Anne-Marie L'Hostis (ENST de Bretagne)  
Jean-François Mercel (ENST de Bretagne)  
Maria-Tereza Segarra (ENST de Bretagne)

### **E-mail Newsletter**

To be kept informed on the EuroT<sub>E</sub>X'2003 conference, please subscribe to our (low-traffic) newsletter by sending the following message:  
SUBSCRIBE eurotex2003 *Your name and affiliation*  
to the address: [sympa@enst-bretagne.fr](mailto:sympa@enst-bretagne.fr).

### **For further information, please contact**

Secrétariat de la conférence EuroT<sub>E</sub>X'2003  
Département Informatique - ENST Bretagne  
tél. : +33 (0)2 29 00 14 27  
fax : +33 (0)2 29 00 12 82  
<mailto:Yannis.Haralambous@enst-bretagne.fr>

*Yannis Haralambous, [Yannis.Haralambous@enst-bretagne.fr](mailto:Yannis.Haralambous@enst-bretagne.fr)*

---

---

# **T<sub>E</sub>X BITS & BYTES: A Conference Newsletter**

---

**Monday, September 9, 2002, Number 1/2002-09-09**

This is a brief newsletter on past, present, and future T<sub>E</sub>X Conference happenings by T<sub>E</sub>X User Groups Worldwide.

## **Contents**

1. TUG 2001 Post Conference Bulletin Online
2. EuroBachoT<sub>E</sub>X 2002, Bachotek, Poland
3. TUG 2002, Thiruvananthapuram, India
4. EuroT<sub>E</sub>X 2003, Brest, France
5. TUG 2003 Outrigger Waikoloa Beach Resort, Big Island, Hawaii

## **2001 August 12–16: Newark, Delaware USA**

TUG 2001 POST CONFERENCE BULLETIN ONLINE

<http://www.tug.org/tug2001/bulletin/>  
[tug2001@tug.org](mailto:tug2001@tug.org)

The TUG 2001 Post-Conference Bulletin and Online Preprints for the 22nd Annual Meeting and Conference of the TeX Users Group at the University of Delaware, Newark, DE, USA on August 12–16, 2001, are available on the TUG website at: <http://www.tug.org/tug2001/bulletin/>

Two special T<sub>E</sub>X Lion Awards were made, to Hàn Th<sup>ệ</sup> Thành in recognition for his most valuable contribution of pdfT<sub>E</sub>X to the community, and a Lifetime Achievement Award to Anita Schwartz, University of Delaware, for 15 years of volunteer service to TUG.

## **April 29–May 3: EuroBachoT<sub>E</sub>X, Poland**

T<sub>E</sub>X AND BEYOND

<http://www.gust.org.pl/EuroBachoTex/>  
[EuroBachoTeX-org@gust.org.pl](mailto:EuroBachoTeX-org@gust.org.pl)

The 13th European and 10th Polish T<sub>E</sub>X Conference, EuroBachoT<sub>E</sub>X 2002 was held on April 29–May 3, 2002 at Bachotek (Brodnicza Lake District), Poland.

The theme of the conference was “ $\text{T}_{\text{E}}\text{X}$  and Beyond”. Bachotek is situated in a very beautiful part of Poland and having all the participants stay at the same resort added a special feeling of camaraderie during the gatherings around the campfire to sing songs and to network with colleagues.

## **2002 September 4–7: Thiruvananthapuram, Kerala, India**

TUG 2002: STAND UP AND BE PROUD OF  $\text{T}_{\text{E}}\text{X}$

TechnoPark Conference Center

<http://www.tug.org.in/tug2002/>

[tug2002@tug.org.in](mailto:tug2002@tug.org.in)

As we write this, more than 70 participants from 11 countries have gathered in Trivandrum to “Stand Up and be Proud of  $\text{T}_{\text{E}}\text{X}$ ” at TUG 2002. The 23rd Annual Meeting and Conference of the TeX Users Group is taking place now at the TechnoPark Conference Center, Thiruvananthapuram, Kerala, India. This conference was organized by the Indian TeX Users Group, spearheaded by C. V. Radhakrishnan of Focal Image (India) Pvt. Ltd., Trivandrum, and was co-sponsored by very generous donations from the Department of Information Technology, Government of Kerala, and TechnoPark Karyavattom, Trivandrum, India.

The opening took place at 9 A. M. on Wednesday September 4. Participants getting off the conference bus were startled and delighted to be greeted by a ceremonial elephant and his mahout and the roar of Indian drums and trumpets. Newspaper reports of this TUG event, appeared in the Business Express section of the local Kerala newspaper, The New Indian Express and in the national paper, The Hindu. Copies of the report will be posted at: <http://www.tug.org.in/tug2002/>

Foreign participants are staying at the Hotel Samudra, Kovalam Beach. The reception and banquet were held here in the garden outdoors under the milky way, beneath swaying coconut palms, and to background music of the breaking ocean waves on the beach. Local entertainment was provided by martial artists, beautiful young Indian dancers from the Shri Chitra Home (for orphans), and an Indian Bamboo flautist, V.C. George. The hospitality and friendliness of the Indian people made everyone feel welcome and participants were treated to a feast of Indian spicy dishes every day.

Interesting papers were presented on current developments in the  $\text{T}_{\text{E}}\text{X}$  world, including ConTeXt, Fonts, MathML, Metapost, XML, and the status of typesetting in Middle Eastern and Asian languages.



## June 24–27: EuroTeX 2003

BACK TO TYPOGRAPHY

ENST Bretagne-Brest, France

<http://omega.enstb.org/eurotex2002/>

[sympa@enst-bretagne.fr](mailto:sympa@enst-bretagne.fr)

In 2003, the EuroTeX will be organized in Brest (Brittany, France), by ENST de Bretagne. The theme of this conference covers the electronic document, digital typography, typographical heritage and its survival in the computer era, new technologies applied to education, book production process, fonts, document internationalization and exchange, etc.

Talks will be held in English or any other European language. The proceedings of the conference will be published in the TUGboat journal, before the fall of 2003.

## 2003 July 20–24: Waikoloa, Big Island, Hawaii USA

TUG 2003: SILVER ANNIVERSARY — 25 YEARS! — of T<sub>E</sub>X.

Outrigger Waikoloa Beach Resort, Big Island, Hawaii

<http://www.tug.org.in/tug2003/>

[tug2003@tug.org](mailto:tug2003@tug.org)

A grand reunion of T<sub>E</sub>X users worldwide is planned for the 24th Annual Meeting and Conference of the T<sub>E</sub>X Users Group in 2003 for the Silver Anniversary — 25 YEARS! — of TeX.

Professor Hermann Zapf and cartoon artist, Duane Bibby, will be just two of what we hope will be a long list of renowned guests and long-time friends of T<sub>E</sub>X, who will help us celebrate this wonderful event.

We would like to make this a big gathering of the people who have played a role in providing you with tools, development, and support for T<sub>E</sub>X, which is the best free software available for beautiful typesetting and typography.

All users of TeX (since its birth) are invited to participate as speaker, attendee, volunteer, sponsor, or donor to help make this conference a success.

## What You Can Do to Help!

You can contribute in several ways to the success of TUG 2003:

1. Come to the meeting and invite all your TeX friends to come along too. Register with the very early-lions *before the end of the year 2002* and no

later than April 9, 2003, to benefit from special conference and accommodation rates.

2. Make a donation, (however large or small) as a token of appreciation for having TeX available to you as free software, and let your colleagues and your bosses know about this meeting so that they too can contribute towards the event, and to the future development of the T<sub>E</sub>X family of programs. T<sub>E</sub>X is very much alive and doing well both as a front-end for authors, and a back-end as a tool in the automated processes that provide easy conversion to XML, PDF, HTML.
3. Publishers, vendors, and other commercial users, developers, and distributors of products that use, or are prepared in, T<sub>E</sub>X (Plain,  $\mathcal{A}\mathcal{M}\mathcal{S}$ -T<sub>E</sub>X,  $\mathcal{A}\mathcal{M}\mathcal{S}$ -L<sup>A</sup>T<sub>E</sub>X, L<sup>A</sup>T<sub>E</sub>X, pdfT<sub>E</sub>X, METAFONT, METAPOST, ConT<sub>E</sub>Xt, etc.) can sponsor or co-sponsor events (coffee breaks, lunches, reception, banquet) and most of all towards renowned plenary speakers and TUG 2003 as a whole.
4. The T<sub>E</sub>X Heritage Committee, chaired by Sebastian Rahtz, will be organizing projects and happenings for the Silver Anniversary of T<sub>E</sub>X. One of these will be to publish a special volume of reflections by key developers and users of T<sub>E</sub>X. The book will be (temporarily) entitled:  
“T<sub>E</sub>X Retrospective: milestones and byways on the road to beautiful typography”  
More information will be published on the website at: <http://www.tug.org/tug2003/heritage/>

Note: Generous donors and sponsors (\$500.00 or more) will receive a complimentary copy of the silver anniversary volume.

So start planning now to join us in Hawaii in body and/or in spirit for TUG 2003 on July 20–24, 2003. Don’t forget to bring along your snorkel . . .

Aloha!

The TUG 2003 Organizers

[tug2003@tug.org](mailto:tug2003@tug.org)

<http://www.tug.org/tug2003/>

## Zpravodaj Československého sdružení uživatelů $\TeX$ u

ISSN 1211-6661 (tištěná verze), ISSN 1213-8185 (online verze)

Vydalo: Československé sdružení uživatelů  $\TeX$ u  
vlastním nákladem jako interní publikaci

Obálka: Antonín Strejc

Počet výtisků: 700

Uzávěrka: 30. května 2002

Odpovědný redaktor: Zdeněk Wagner

Redakční rada: Petr Aubrecht, Matěj Cepl, Jiří Demel,  
Jana Chlebíková, Jiří Kosek, Jaromír Kuben,  
Petr Sojka, Martin Tkadlčík

Tisk a distribuce: KONVOJ, spol. s r. o., Berkova 22, 612 00 Brno,  
tel. 05-740233

Adresa:  $\zeta$ TUG, c/o FI MU, Botanická 68a, 602 00 Brno

fax: 05-412 125 68

e-mail: [cstug@cstug.cz](mailto:cstug@cstug.cz)

Zřízené poštovní aliasy sdružení  $\zeta$ TUG:

[bulletin@cstug.cz](mailto:bulletin@cstug.cz), [zpravodaj@cstug.cz](mailto:zpravodaj@cstug.cz)

korespondence ohledně Zpravodaje sdružení

[board@cstug.cz](mailto:board@cstug.cz)

korespondence členům výboru

[cstug@cstug.cz](mailto:cstug@cstug.cz), [president@cstug.cz](mailto:president@cstug.cz)

korespondence předsedovi sdružení

[gacstug@cstug.cz](mailto:gacstug@cstug.cz)

grantová agentura  $\zeta$ TUGu

[secretary@cstug.cz](mailto:secretary@cstug.cz), [orders@cstug.cz](mailto:orders@cstug.cz)

korespondence administrativní síle sdružení, objednávky CD-ROM

[cstug-members@cstug.cz](mailto:cstug-members@cstug.cz)

korespondence členům sdružení

[cstug-faq@cstug.cz](mailto:cstug-faq@cstug.cz)

řešené otázky s odpověďmi navrhované k zařazení do dokumentu  $\zeta$ FAQ

[bookorders@cstug.cz](mailto:bookorders@cstug.cz)

objednávky tištěné  $\TeX$ ové literatury na dobírku

ftp server sdružení:

<ftp://ftp.cstug.cz/>

www server sdružení:

<http://www.cstug.cz/>