DOI: 10.5300/2011-OSSConf/17

XeLaTeX – KROK SPRÁVNÝM SMĚREM

RYBIČKA, Jiří (CZ)

Abstrakt. *Mnoho uživatelů osobních počítačů má zájem vytvořit kvalitní dokumenty, nejčastěji ze své profesní oblasti, tedy odborné texty s řadou speciálních symbolů. Donedávna byla jednou z mála skutečně použitelných variant sazba v systému LaTeX – avšak vzhledem k jeho historické koncepci vznikala řada obtížně řešitelných problémů zejména s fonty a kódováním vstupu. Příspěvek se zabývá možnostmi a některými zkušenostmi při použití systému XeLaTeX, jehož vlastnosti mohou výrazně přispět k zachování kvalitní sazby jako ve známém a rozšířeném systému LaTeX, ale s mnoha zajímavými rozšířeními: využití fontů instalovaných v rámci operačního systému, možnosti vstupu znaků v kódování UTF-8 a s tím spojené možnosti sazby v různých jazycích, možnosti matematické sazby a použití speciálních symbolů potřebných pro odborné texty.*

Klíčová slova. *TeX, XeTeX, LaTeX, XeLaTeX, kompatibilita, fonty dokumentu, kódování vstupu, UTF-8, vícejazyčná sazba.*

XeLaTeX – STEP IN THE RIGHT DIRECTION

Abstract. *Many PC users interested in creating quality documents, most of their profession, i.e. technical texts with number of special symbols. One of the few really usable possibilities is LaTeX – but due to its historical concept there are difficult problems solvable in particular, e.g. the use of various fonts and input encoding. The paper deals with possibilities and some experience in using the system XeLaTeX whose properties can preserve present quality of LaTeX documents but with many interesting extensions: use of fonts installed in the operating system, the possibility of entering characters in the UTF-8 encoding and with the associated possibility of different languages characters, mathematical and special symbols needed for technical texts.*

Key words and phrases. *TeX, XeTeX, LaTeX, XeLaTeX, compatibility, document fonts, input encoding, UTF-8, multilingual typesetting.*

Úvod

Již dlouho slouží zejména pro tvorbu odborných textů mnoha uživatelům systém LaTeX. S tím je spojena řada specifik. Principy systému definované na začátku 80. let minulého století jsou na jedné straně geniální a v mnoha aspektech trvale použitelné, na druhé straně však s sebou nutně nesou nepříjemné „drobnosti“. Například jednou z nejznámějších vlastností je silná snaha o zpětnou kompatibilitu. Ta ovšem dost často vede k řešení, které je těžkopádné, pro řadu uživatelů nepohodlné a mnohdy i na první pohled nepochopitelné.

I když je například zmíněná snaha o zpětnou kompatibilitu nesporně velmi kladným prvkem, vždy je pravděpodobně potřeba najít vhodný kompromis a rozhodnout se pro optimální řešení.

Systémy postavené na principu TeXu se nacházejí v silné konkurenci jiných přístupů, které mají daleko větší rozšíření. Cílem tohoto článku není zkoumat, proč tomu tak je, příčiny totiž nejsou podstatné. Daleko důležitější je, že konkurenční systémy víceméně určují směry vývoje a uživatelé se tomu přizpůsobují, ať se jim to líbí nebo ne. Opět nebudeme hodnotit, zda je to dobře, do jaké míry je to otázka reklamy, byznysu nebo snad technických prvků.

Postavení systémů, které jsou nyní středem našeho zájmu, tedy systémů volně dostupných a založených na principu TeXu, je zcela jistě nelehké. Je-li naším cílem zvýšit počet uživatelů, kteří je budou *skutečně chtít* využívat, musíme se nutně zamýšlet nad tím, jaké konkurenční výhody tyto systémy vůbec mohou přinášet, dále nad tím, jak tuto skutečnost uživatelům sdělit, a v konečné fázi také nad tím, jak tuto snahu podpořit.

Tento článek by měl být příspěvkem k prvnímu z uvedených prvků – všimneme si možností, které mohou přispět ke zvýšení konkurenceschopnosti systémů postavených na principu TeXu, konkrétně v podání systému XeLaTeX.

Proč XeLaTeX?

Všeobecně známou silnou stránkou všech systémů postavených na principu TeXu je velká přizpůsobitelnost. Existence nástrojů, jako jsou definice nebo redefinice příkazů a dokonce schopnost změnit vnímání vstupního lexika, poskytuje nepřeberné možnosti vytvářet specifické prostředky pro řešení všemožných potřeb uživatelů. Pomineme-li v tuto chvíli skutečnost, že některá taková rozšíření jsou velmi netriviální a vyžadují skutečně hluboké znalosti mnoha komplikovaných vnitřních vlastností a algoritmů, vzniká tím také do značné míry velmi rozsáhlá a málo přehledná množina realizací, jejíž použitelnost je v některých případech opravdu diskutabilní. Jmenujme například sazbu tabulek – existuje přes 30 balíčků, které se nějak dotýkají nejrůznějších tabulkových prvků. Nejenže ani těch 30 balíčků neřeší všechno, ale některé nelze použít s jinými. Mezi uživateli pak lze vysledovat celou škálu přístupů – od těch, kteří říkají „na všechno už existuje nějaký balíček LaTeXu, jen je potřeba jej najít“, až po ty, kteří zastávají názor „nikdy nepoužívám cizí makra“.

Řekneme-li tedy XeTeX nebo XeLaTeX, běžný uživatel pravděpodobně již předem pojme podezření, že se jedná „zas o něco jako další balíček“ nebo specifickou nadstavbu, jejíž použitelnost v běžné praxi českého (slovenského) autora a sazeče nebude nijak

významná, nebo (a to je ještě nepříjemnější) bude nějak měnit již zaběhnuté, ověřené a používané postupy.

Lze však říci, že jako příspěvek ke konkurenceschopnosti systémů tohoto typu představuje XeLaTeX *krok správným směrem.* Autor Jonathan Kew jasně deklaroval jeho základní cíle: vyřešit především dva velké nedostatky dosavadních řešení – umožnit snadné použití všemožných fontů a zpracovávat zdrojový text v kódování UTF-8, a to vše tak, aby to bylo pro uživatele co nejjednodušší.

Zdrojový text

Deklarovaná jednoduchost práce s XeLaTeXem je vyjádřena například i tím, že pro jeho použití při zpracování již existujících textů, u nichž chceme dostat stejný výsledek, není potřeba téměř nic měnit. Stačí připojit balíček `xltxtra` zahrnující v sobě volání dalších potřebných balíčků: zejména `fontspec` a `xunicode`.

Příklad minimálního zdrojového textu:

```
\documentclass{article}
\usepackage[czech]{babel}
\usepackage{xltxtra}
\begin{document}
Příliš žluťoučký kůň úpěl ďábelské ódy.
\end{document}
```

Po překladu překladačem `xelatex` dostaneme výsledné PDF, v němž bude automaticky zvoleno písmo Latin Modern.

Pokusme se přiblížit alespoň některé směry, kterými kráčí XeTeX/XeLaTeX.

Směr – fonty

Kdysi bývalo zvykem, že jakýkoliv dokument, který vzešel z TeXové nebo LaTeXové dílny, se pyšnil použitým Knuthovým písmem Computer Modern (byť ve verzi CS nebo ještě nověji LM). K této pýše je „bezesporu“ dobrý důvod: toto písmo je v mnoha aspektech dokonalé a připočteme-li k tomu ještě dokonalé algoritmy sazby, výsledek (zejména v odborných textech s matematickými výrazy) byl skutečně téměř nedostižný (a je i dnes!).

Computer Modern však není jediné použitelné písmo a navíc systémem LaTeX nesázíme vždy jen nějakou matematickou perlu. Kvalitní typografický návrh dokumentu tedy počítá s tím, že zvolíme *odpovídající* písmo. Co můžeme vybírat v konkurenčních systémech? Při rozbalení nějaké jejich nabídky s písmy se objeví řádově desítky až stovky položek. Ano, je jistě potřebné ihned dodat, že mnoho z nich představuje nepoužitelný plevel, ale i přesto je samotný proces volby písma velmi jednoduchý a dojde-li k tomu, že pečlivý, znalý a zkušený uživatel si nějaké písmo obstará, do seznamu nabízených písem je dostane prostou kopií do vhodného adresáře.

Kdo z běžných uživatelů někdy ovšem pojal myšlenku, že potřebuje nějaké pěkné písmo do svého dokumentu v LaTeXu, obstaral si je a chtěl je použít, pak těžce narazil: nezbývalo než se ponořit do nepřehledné bažiny písmových formátů, národních specifik, různých transformací, konfigurací, adresářových cest a systémových nastavení. Je

zřejmé, že bez hlubokých znalostí, navíc v každém typu distribuce zcela odlišných a závislých na použitém operačním systému, nebylo možné písmo použít. A takové nároky zcela diskvalifikovaly jinak nedostižně kvalitní a do značné míry geniální TEXové systémy v očích „běžných“ uživatelů, kteří nemají ambice stát se odborníkem na počítačová písma, ale „pouze“ chtějí vytvořit třeba pěknou pozvánku na promoci…

V tomto kontextu je tedy systém umožňující použít *libovolný* font instalovaný v operačním systému doslova zázrakem. Základem je možnost systému XETEX, kdy lze pomocí primitivu `\font` připojit libovolný dostupný font zadáním jeho jména. Například zápis

```
\font\nadpis="Minion Pro Bold" at 16pt
\fbox{\nadpis Příliš náročná firemní výroba větrných elektráren}
```

jednoduše umožňuje vysázet:

Příliš náročná firemní výroba větrných elektráren

Připomeňme, že font Adobe Minion Pro Bold je ve formátu Open Type, na disku se jmenuje MinionPro-Bold.otf a je nakopírován v adresáři spolu s ostatními systémovými fonty. Tato rodina je také součástí volně dostupného Adobe Readeru. Všimněte si, že je automaticky použita ligatura „fi“ a nejsou problémy s národními znaky.

To ale není zdaleka všechno. LATEXový komfort volby písma, který umožňuje ovládat celý dokument symbolickými příkazy bez detailní specifikace konkrétního písma, je uživateli dostupný pomocí balíčku `fontspec`, jehož autorem je Will Robertson. Nejenže lze nastavit písmovou rodinu, ale také ovládat řezy a stupně standardními LATEXovými příkazy a ve spolupráci s dalšími balíčky využívat dosavadní způsoby sazby speciálních znaků. Již vytvořené zdrojové texty lze tedy znovu použít.

V přehledu uvedeme základní možnosti balíčku `fontspec`:

`\fontspec` – jedná se o základní příkaz pro výběr fontu. V povinném parametru je potřebné uvést jméno požadovaného fontu, v nepovinném parametru pak lze uvést řadu voleb sloužících k upřesnění výběru a chování fontu. Například zápis
`{\fontspec[Scale=0.75]{Lucida Console}Toto je strojopis.}`
nastaví strojopisný font Lucida Console a vysází příslušný text v dané skupině:
`Toto je strojopis.`

Po nastavení fontu je možné používat běžné příkazy pro změny řezu, a pokud příslušný řez existuje, je automaticky zapnut. Například v písmu Georgia:
Základní řez – *kurzíva* – **tučné** – ***tučná kurzíva.***

Volba `Scale` umožňuje změnit originální velikost s daným koeficientem, což například u strojopisného fontu vede k vyrovnanějšímu obrazu s okolním textem.

Kromě konkrétního koeficientu (jak bylo v příkladu použito 0,75) lze použít i inteligentní hodnoty `MatchLowercase` a `MatchUppercase`, které nastavují velikost rovnu minuskové, resp. verzálkové velikosti okolního fontu.

Jednou z výhod písem formátu OTF je řada typografických vymožeností, například variantní číslice. Verzálkové číslice vhodné do tabulek (mají stejné šířky kvůli správnému zarovnání) a minuskové číslice vhodné do běžného textu mohou být zařazeny volbou `Numbers=Uppercase`, resp. `Numbers=OldStyle`.

Například ve fontu Cambria: verzálkové číslice do tabulek 1234567890 a minuskové číslice do textu 1234567890.

Z dalších možných voleb zmíníme `Mapping=tex-text` (umožňuje používat běžné příkazy pro speciální znaky, například dva spojovníky pro pomlčku apod.), `Color` (definuje barvu, zadává se jako trojice hexadecimálních hodnot definujících složky RGB a volitelně čtvrté číslo jako průhlednost), `LetterSpace` (prostrkání, hodnotou je procento stupně písma použité pro prostrkávací mezery), `FakeSlant`, `FakeStretch` a `FakeBold` (mechanické změny tvaru; pomůcka pro modifikaci sklonu, zúžení řezu nebo změnu duktu – tyto změny jsou realizovány jen počítačovou transformací, proto je potřeba jich využívat střízlivě a v odůvodněných případech).

`\setmainfont` – nastavení základního (antikvového) písma dokumentu.
Příkaz má stejnou syntax jako `\fontspec`.

`\setsansfont` – nastavení fontu, který v dokumentu představuje bezserifovou rodinu.
Syntax opět stejná jako `\fontspec`.

`\setmonofont` – nastavení fontu, který v dokumentu představuje strojopisnou rodinu.
Syntax je stejná jako `\fontspec`.

Uvedenými třemi příkazy lze definovat celkové podání dokumentu. Pro změnu rodiny, řezu a stupně fungují běžné LaTeXové nástroje, nastavená písma fungují i v příkazech pro nadpisy atd. Pokud nejsou žádné fonty zavedeny, automaticky se zvolí rodina Latin Modern, reprezentující tradiční pojetí – sazbu dokumentu písmem odvozeným z Knuthova originálu.

`\defaultfontfeatures` – příkaz umožňující nastavit volby, které se pak použijí v každém dalším příkazu nastavujícím konkrétní font. Užitečné je například nastavit
`\defaultfontfeatures{Mapping=tex-text}`
a tuto volbu pak není nutné uvádět u žádného dalšího příkazu `\fontspec`.

Jak již bylo zmíněno, popsaný balíček `fontspec` je volán z balíčku `xltxtra`. Ten zabezpečuje správnou funkci několika prvků, které jsou přechodem na univerzální kódování vstupu a libovolné fonty porušeny (textové indexy a exponenty, body volitelného dělení, prostředí `verbatim*` a příkaz `\verb*`) a také definuje loga TeX a další.

Směr – kódování vstupu

V současné době je i v textech primárně psaných v jednom jazyce často potřeba uvádět slova (například vlastní jména) s jiným pravopisem a jiným repertoárem znaků. V tomto směru měly systémy postavené na principu TeXu vždy rozsáhlé možnosti, všechny zápisy rozličných národních a jiných znaků však byly v původní podobě zapisovány sedmibitovými znaky tabulky ASCII (univerzální způsob dosažitelný na jakémkoliv zařízení). To s sebou samozřejmě nese i určitou těžkopádnost, takže například běžný český text takto nezapisujeme a přirozeně předpokládáme, že národní znaky přímo vidíme v editoru.

Tento způsob práce se s rozvojem konkurenčního programového vybavení stal již standardem. Pro běžného uživatele takových systémů je těžko představitelné, že by například v českém textu zapsal jméno François Didôt jako `Fran\c{c}ois Did\^ot`,

případně jméno azbukou nebo dalšími exotičtějšími jazyky ještě podstatně složitěji. Vyžaduje, aby přímo v pořizovacím programu byly všechny znaky vidět. Zapisuje tedy přímo z klávesnice `François Didôt` nebo `Александр Пушкин` apod. Tuto funkčnost zajišťuje univerzální kódování národních znaků, jakým je UTF-8. Na univerzální kódování přechází většina programů, bylo by tedy velmi svazující, kdyby se TEXové systémy do této množiny nezařadily.

Důležitost univerzálního kódování vstupu je podtržena ještě i skutečností, že nejde jen o nějaké víceméně exotické národní znaky, které používá jen zlomek uživatelů, ale i o nejběžnější typografické rekvizity potřebné v prakticky jakémkoliv textu.

Porovnejme si alespoň některé možnosti, které byly doposud k dispozici, s možnostmi systému XƎLATEX se vstupem v kódování UTF-8.

Uvozovky – Ty byly dostupné vždy pomocí nějakého příkazu, nejčastěji `\uv{...}`, případně `\clqq` a `\crqq`. Tyto dva příkazy jsou definovány buď ve specifickém stylu `czech`, nebo v české podpoře vložené do stylu `babel`. Úhlové uvozovky byly poněkud problematičtější, byly vkládány jako konkrétní znak daného fontu. Díky kódování UTF-8 se můžeme vyhnout jakémukoliv příkazu a přímo zapisovat „`slovo`“, nebo také »`slovo`«. Nevýhodou může být skutečnost, že používaný editor neumí tyto znaky efektivně vkládat přímo z klávesnice, to je však většinou možné snadno překonat. Vhodných editorů, které plně respektují celou znakovou sadu UTF-8, je dostatečné množství.

Pomlčky – Tento znak je pravděpodobně jediný, jehož doposud užívaný zápis je natolik efektivní, že jej není potřeba příliš měnit. Zápis dvou nebo tří spojovníků za sebou vede na ligaturu představující půlčtverčíkovou nebo čtverčíkovou pomlčku. Tohoto efektu je dosaženo i v novém pojetí (viz `fontspec`), pokud font zavedeme s volbou `Mapping=tex-text`. Kromě toho lze ovšem použít i vložení příslušného znaku s kódem 2013(hex) nebo s kódem 2014(hex). To je výhodné zejména tehdy, pokud editor sám například promění spojovník zapsaný s mezerami za pomlčku.

Měny – Jeden z nejčastějších dotazů poslední doby je „jak prosím zapíšu znak euro?“ a dost trapnou odpovědí bylo „víte, to musíte připojit balíček `eurosym` a pak zapisovat `\euro`“. Uživatel, který klade takový dotaz, je spíše začátečník a předpokládá, když se tento znak jednoduše zapisuje všude jinde, stejně jednoduše to půjde i v tak dokonalých systémech, jaké jsou ty TEXové. Tento předpoklad je splněn až nyní. Opět jde o zlepšení výsledku, protože příslušný znak je použit z daného fontu, nikoliv z náhradního fontu používaného balíčkem `eurosym`. Příklad: 15 €. Podobně lze využít znaku pro britskou libru a japonský jen: 15 £ a 15 000 ¥.

Násobení – Znak byl dostupný dosud jen jako matematický symbol `$\times$`. Nevýhodou tohoto řešení je zejména skutečnost, že kresba matematického symbolu nemusela dobře korespondovat s kresbou použitého textového fontu. Vezmeme-li v úvahu skutečnost, že se jedná více o textovou než matematickou rekvizitu, je možnost vložení přímého znaku určitě přínosná. Porovnejme matematický symbol: $5\times$ nebo přímý znak daného fontu: 5×.

Vybrané matematické symboly – Je možné, že by v matematické sazbě existovalo něco, co by systémy založené na principu TEXu neuměly nejlépe ze všech? Jak už bylo

uvedeno u znaku pro násobení, nejde většinou přímo o matematické výrazy, ale spíše prvky, které se matematickým prostředím vždy řešily, ale nyní to už není nutné a navíc získáváme určité zlepšení. Jedním z takových symbolů je π – ve smyslu konstanty. Tu je podle typografických pravidel potřebné sázet vzpřímeným řezem, ale matematický symbol `\pi` dává kurzívní variantu π. Obdobně symbol µ používaný jako předpona mikro-. Z dalších užitečných znaků jmenujme symbol negace ¬, ochranná známka ®, stupeň °, plus minus ±, exponenty ^{1}, ^{2}, ^{3}, paragraf §.

Směr – za hranice českých a slovenských specifik

Již od začátku jsou systémy na principu TeXu vybaveny určitými možnostmi národních specifik (uveďme například široký repertoár akcentů nebo příkaz `\discretionary`), ale přece jen to bylo považováno za nedostačující a pro dosažení kvalitních a prakticky použitelných výsledků byla vždy doplňována rozsáhlá podpora specifická pro české a slovenské uživatele.

Rozhodně nelze říct, že by na tom bylo něco špatného – rozšíření pro kvalitní českou a slovenskou sazbu je velmi propracované a v určitém směru předčilo svými kvalitami původní možnosti, dokonce lze říci, že vůbec umožnilo běžné použití TeXového systému obyčejnými uživateli.

V tomto přístupu nebyla komunita českých a slovenských uživatelů TeXu výjimkou. Podobnými rozšířeními a doplňky byly například vybavovány i instalace pro sazbu polských i německých textů. To s sebou automaticky přinášelo určitou povinnost tyto instalace udržovat a distribuovat, současně to přinášelo jisté obtíže při přenosech zdrojových textů mezi různými takto upravenými instalacemi.

Ukazuje se, že údržba a další rozvoj specifických instalací není nejlepší volbou – možnost využít univerzálního řešení, které je podporováno mezinárodně a u kterého je větší naděje na kontinuitu a další vývoj, je pro národní specifika do budoucna pravděpodobně jedinou schůdnou cestou.

Využití XeLaTeXu může takovou cestu představovat. Již zmíněné možnosti vstupu zdrojového textu v kódování UTF-8 a možnosti využití nejrůznějších fontů obsahujících národní znaky nejen české a slovenské abecedy, ale i cyrilice apod. jsou toho jistým dokladem. Krokem k univerzálnímu řešení národních specifik sazby je balíček `babel`, který je již důstojnou alternativou specifických národních rozšíření. Kromě toho však lze využít balíček `polyglossia`, o němž se krátce zmíníme.

Balíček `polyglossia`, jehož autorem je François Charette, má v manuálu podtitul „náhrada Babelu v XeLaTeXu“. Nejedná se však jen o náhradu, ale o nové možnosti využívající plně existujících rozšíření. Balíček zavede pro všechny jazyky použité v dokumentu odpovídající vzory dělení, umožňuje nastavit font při přepnutí jazyka (využívá při tom balíček `fontspec`), nastaví určité typografické konvence (mezerování kolem interpunkce apod.), předefinuje standardní řetězce „kapitola“ „obrázek“ atd., přizpůsobí formátování kalendářního data, případně nastaví i správný směr sazby (využívá balíček `bidi`).

Zavedení balíčku lze provést alespoň dvěma způsoby. První způsob je podobný způsobu balíčku `babel`, tedy `\usepackage[lang1,lang2,...]{polyglossia}`, ale oproti balíčku `babel` je jazyk uvedený v nepovinném parametru jako *první* brán jako jazyk implicitní.

Druhou možností je zavést balíček `polyglossia` bez volitelného parametru a jednotlivé jazyky aktivovat příkazem `\setdefaultlanguage[volby]{jazyk}` pro implicitní jazyk a příkazem `\setotherlanguage[volby]{jazyk}` pro alternativní jazyky použité v dokumentu.

Přepínání mezi jazyky lze provést pro krátké texty příkazem `\text`*jazyk*`{text}`.

Například příkaz `\textrussian{\today}` dává po vysázení vysázení 30 мая 2011 r. nebo `\textgreek{\today}` zase 30 Μαΐου 2011.

Pro delší texty v daném jazyce je k dispozici prostředí s názvem jazyka, například: `\begin{slovak}...\end{slovak}`.

Volitelné parametry u aktivovaných jazyků umožňují různá nastavení a jsou k dispozici pro každý jazyk zvlášť. Jejich podrobný seznam je uveden v manuálu. Protože sazba exotických jazyků není hlavním cílem většiny našich uživatelů, nebudeme se rozsáhlým možnostem balíčku `polyglossia` v těchto oblastech věnovat.

Závěr

Systém XeLaTeX představuje bezesporu nástroj, který poskytuje uživatelům LaTeXu významná ulehčení při realizaci typografického návrhu dokumentů, rozšíření možností a řešení drobných, ale nepříjemných problémů.

Podobně jako výstupní formát PDF znamenal významné přiblížení TeXových nástrojů světu konkurenčních systémů počítačové sazby, je i XeLaTeX systémem, jež posunuje vývoj podobným směrem.

Z praktických zkušeností vyplývá, že s novými přístupy jsou spojeny i dílčí nedostatky – ne všechno funguje, jak je popisováno v manuálech, ne všechno je zcela transparentní a zcela jistě se musíme smířit se změnami, které nezaručují stoprocentní zpětnou kompatibilitu.

Seznámení s možnostmi XeLaTeXu může uživateli přinést významný efekt. Pokud k tomu přispěje i tento článek, byl jeho cíl bezezbytku splněn.

Kontaktní adresa

Doc. Ing. Jiří Rybička, Dr.,
Ústav informatiky Provozně ekonomické fakulty Mendelovy univerzity v Brně,
Zemědělská 1, 613 00 Brno, e-mail: `rybicka@mendelu.cz`