Použití LaTeXu pro zpracování firemní dokumentace

Martin Bruchanov

4. června 2003

Požavavky na firemní dokumentaci můžeme rozdělit na dvě samostatné skupiny. Do první se řadí požadavky na reprezentativní vzhled a bezproblémové zobrazení uživateli. Druhá skupina pokrývá neméně důležitou část spíše technických požadavků, kterými jsou snadná údržba, možnosti vyhledávání, archiv verzí různých dokumentů, atd.

Dosáhnout kvalitního vzhledu a publikovat dokument v nějakém široce přenositelném formátu jako je HTML nebo PDF by mělo být naprostou samozřejmostí.

Problémovější se jeví druhá záležitost. O způsobu jak jsem se s ní vypořádal bude pojednávat tento dokument.

1.1. Firemní dokumentace

Když jsem v roce 2000 nastoupil do jedné nejmenované firmy setkal jsem se s žalostným stavem správy dokumentů. V několika adresářích na mě čekalo několik megabajtů nesourodých dokumentů obsahujích manuály k prodávaným zařízení, dokumentaci k software či popis vyvýjeného komunikačního systému. Na souborech by se dal nádherně popsat vývoj používaných editorů u nás – nejstarší soubory ve formátu T602 a WordPerfect, následované formátem AmiPro, až po ty novější vytvořené pomocí Microsoft Office. Bohužel, neustále probíhající upgrady počítačů, vývoj operačních systémů a změny módních vln a verzí textových procesorů způsobily, že z některých souborů se stalo během několika málo roků nečitelné smetí.

Nečitelnost starších formátů nebyl jediným zádrhelem. Aktuálně používaný kancelářský balík, kromě jistých bezpečnostních problémů způsobovaných populárními makroviry, se při zpracování rozsáhlejších dokumentů choval značně nestabilně.

Tahle situace vyžadovala radikální změnu – zvolit časově stabilní a prověřený formát.

1.2. Volba nástrojů

Takových formátů se nabízí hned několik např. SGML, DocBook nebo LaTeX. Naše volba padla na poslední z jmenovaných. A to proto, že hlavním požadavkem byl co nejlepší výstup pro tisk a do PDF.

Jaké jsou hlavní výhody nového řešení? LaTeXje otevřený formát, soubory jsou uloženy jako čistý ASCII text doplněný o formátovací značky podobně jako třeba HTML. Jen tato vlastnost přidává pro zpracování dokumentů nové možnosti o které uživatelé uzavřených binárních formátů přicházejí. Možnost snadno prohledat soubory obvyklými souborovými manažery, rychlá editace a úspora místa. Pro úpravu dokumentů je možno využít mnoho utilitek pro práci s textovými soubory jako je `grep`, `sed` a další. Je možné využít nástroje pro týmovou spolupráci a archivování verzí původně určených programátorům pro správu zdrojových kódů. Jak si ukážeme později je možné všechny tyto nástroje používat svobodně a zadarmo, takže k výhodám je možné připočítat i odlehčení finanční stránky záležitosti.

Každé pro má ale i své proti. Překonatelnou nevýhodou je odlišný přístup k zpracování dokumentů, místo editování kdy uživatel ve svém editoru rovnou vidí výsledky práce je v LaTeXu dokument nejprve „označkovat“ a poté několika příkazy z něj vygenerovat DVI nebo PDF soubor, který je už možné tisknout nebo prohlížet.

2. Nástroje

2.1. Distribuce TeXu

Nejdůležitějším nástrojem pro zpracování zdrojových dokumentů je TeX, který je v současné době dostupný většinou jako rozsáhlý balík všech potřebných utilit, maker a fontů. Pro GNU/Linux je zde dostupný `tetex` pro MS-Windows `MikTeX` a nebo rozšířená distribuce `TeXLive`, která obsahuje binární soubory pro několik operačních systémů.

Instalace distribuce v případě `TeXLive` se rovná pouhému zkopírování dat z CD na lokální disk pomocí instalačního skriptu a vše je hned připraveno k použití. Uživateli/správci odpadá starost o generování souborů s metrikami písem, počešťování, které je nutno provést u nelokalizovaných distribucí.

Od prvotní snahy nainstalovat TeXzvlášť pro každého klienta jsem upustil. Jako nejvhodnější mi přišlo řešení, kdy TeXbude nainstalován pouze na jediném počítači na kterém budou uživatelé pracovat vzáleně přes síťový terminál. Tento počítač je zároveň i SMB server, takže každý klient má na svém počítači připojený zároveň síťový disk. To dovoluje, aby uživatel používal editor a PDF prohlížeč na svém počítači a překlad spouštěl přes terminál na serveru.

2.2. Editor

Volba editoru je díky formátu, který tvoří otevřený text, jen věcí uživatele, který použije ten na který je zvyklý, ať už je to `ViM`, `emacs` nebo `notepad`.

Mě vyhovuje `ViM`, který kromě mnoha předností zpříjemňujících práci obsahuje např. obarvováním syntaxe, zobrazováním chyb z logů a možnost doplňení vlastních maker či automatické doplňování slov.

Pro uživalete Windows a začátečníky je vhodný sharewarový editor `WinEdt`, který je přímo určen pro práci s LaTeXem, dovoluje doinstalovat slovník pro kontrolu překlepů a navíc obsahuje roletová menu pro vkládání LaTeXových značek.

2.3. Organizace souborů

Přepokládejme, že máme řadu dokumentů. Chceme vytvářet jednotlivé dokumenty, více dokumentů spojených v jednom PDF a chceme dodržet jednotný vzhled.

LaTeXový dokument se skládá z hlavičky, kde se definuje typ dokumentu a použité balíčky s rozšiřujícími funkcemi, následuje popis vzhledu dokumentu a pak až vlastní text. Je možné mít tyto jednotlivé části v samostatných souborech, které se pak pomocí `\input` vloží na spravné místo a v případě, že chceme nějak upravit vzhled všech dokumentů stačí upravit pouze jeden soubor.

Pro přehlednost je vhodné mít pro každý dokument vytvořený samostatný adresář, kde mít ještě ve dvou podadresářích odděleně text a obrázky.

Zastavme se ještě u obrázků. Ty se do LaTeXu mimo jiné vkládají ve formátu EPS (Encapsulated PostSctript) nebo pokud se používá pdfLaTeX tak ty vektorové ve formátu PDF a rastrové např. ve formátu PNG.

Většina vektorových editorů ovšem nemá formáty z postscriptové rodiny jako svůj přirozený formát, ale umožňuje pouze jejich export, případně import. Ideální je ukládat jak exportované soubory, tak i původní formáty editorů (xfig, sxd, cdr). Předejte se tak nepříjemným překvapením, kdy po exportu a zpětném importu zjistíme, že přerušovaná čára není přerušované čára, ale mnoho čar s mezerami nebo že kružnice už není kružnice, ale nějaký útvar složený z čar který jí jen nápadně připomíná.

Veškeré nové úpravy se pak dělají na původních souborech a ty se potom exportují do požadovaného formátu.

2.4. Teamová práce

Mnohdy nastane situace, kdy více lidí pracuje na jednom dokumentu, aby nedocházelo ke kolizím existují systémy pro archivování a teamovou spolupráci. Jedním z nich je CVS (Concurrent Version System). Funguje následujícím způsobem.

Všechny soubory projektu na kterém pracujete jsou uloženy na serveru v tzv. CVS repository. Každý člen týmu si je stáhne k sobě do svého pracovnáho adresáře (`checkout`) a pracuje na nich. Odvede kus práce a rozhodne se hotovou věc nahrát spátku do CVS repository. Mohou nastat dvě situace. Zaprvé, nová práce se uloží na server spolu s datumem uložení a poznámkou do logu o co šlo. Druhá, méně obviklá, situace je, že během práce někdo jiný upravil stejné soubory a uložil je do repository. Tahle část už vyžaduje zásah uživatele – provést znovu stažení souborů projektu, CVS označí soubory které se liší a může se pokusit sestavit rozdíly, pokud se to nepodaří jsou v souborech vyznačená místa rozdílů (podobné výstupu utilitky `diff`), takže první uživatel vidí kdo a jakou změnu v dokumentu provedl a podle toho se zařídí.

CVS funguje tak, že je jeden centrální server (nejlépe zálohovaný každý den) z kterého si klienti stahují data k sobě. V repository CVS je pak uložen, v případě ASCII textu, původní soubor a dohrávají se pouze změny. Každá změna je pečlivě archivována, takže existuje možnost kdykoliv se vrátit k dokumentům před provedením nějaké zásadní změny apod.

CVS má možnost i archivace binárních souborů (např. obrázky), ale ty je třeba při registraci souborů označit jiným atributem a potom se nearchivují ve formě patchů.

2.5. Skriptíky ulehčí spoustu práce

Spousta často opakujících se činností se dá plně zautomatizovat. V našem případě to bylo hlavně překládání dokumentů a jejich následné strkání do správných adresářů WWW a FTP serveru.

Tahle činnost se dá v GNU/Linuxu vyřešit mnoha elegantními způsoby. Já jsem si pro to zvolil nástroj GNU `make`. Jedná se o velice chytrý systém, známý především programátorům, kteří jej používají pro kompilaci rozssáhlejších programů.

Je nutné vytvořit soubor s názvem `Makefile`. Tento soubor může obsahovat veškeré skripty pro sestavení dokumentu.

Rozsáhlejší příklad obsahu `Makefile` může vypadat například takto:

```
DIR = hardware/
OUT = output
OUT_EN = $(OUT)/en
USETEXEN = pdflatex $(DIR)
TFILES_EN = $(DIR)adresa-en.tex $(DIR)styl-en.tex

MAN_SRC = $(DIR)manual/en/
PROJECT = $(DIR)manual_v12-en.tex \
      $(MAN_SRC)uvod.tex \
      $(MAN_SRC)popis.tex \
      $(MAN_SRC)nastaveni.tex \
      $(MAN_SRC)zapojeni.tex \
      $(MAN_SRC)utp_cable.tex

$(OUT_EN)/manual_v12-en.pdf:  $(PROJECT) $(TFILES_EN)
    $(USETEXEN)manual_v12-en.tex
    $(USETEXEN)manual_v12-en.tex
    $(USETEXEN)manual_v12-en.tex
    mv manual_v12en.pdf $(OUT_EN)

manual: $(OUT_EN)/manual_v12-en.pdf
```

Teď si povíme něco k obsahu. Náš `Makefile` obsahuje dvě pravidla, nás zajíma `manual`, dále několik maker, která se definují např. `USETEXEN = pdflatex $(DIR)` a při zavolání `$(USETEXEN)` se rozvinou.

Uživatel, který si stáhne z CVS potřebné soubory pak zadá jednoduše `make manual` a proběhne následující proces. Utilitka `make` se podívá do aktuálního adresáře a otevře soubor `Makefile`, zde najde pravidlo `manual` a to říká že má hledat soubor nebo zpracovat pravidlo `output/en/manual_v12-en.pdf`, soubor nenachází, takže se podívá po stejně nazvaném pravidle, tady (před rozvinutím):

```
$(OUT_EN)/manual_v12-en.pdf
```

Zde je nadefinováno opět několik souborů – jsou to soubory `PROJECT` a `TFILES_EN` nutné ke zpracování dokumentu. Za tímhle pravidlem už je uvedena řada příkazů které se mají vykonat. Je zde třikrát řádek:

```
$(USETEXEN)manual_v12-en.tex
```

který se rozvine na příkaz `pdfcslatex hardware/manual_v12-en.tex` a provede se třikrát po sobě (kvůli zpracování obsahu a zařazení křížových odkazů LaTeXem). Poté se soubor z pracovního adresáře přesune do výstupního.

Další věci, které se dají do `Makefile` zařadit, jsou pravidla pro sestavení všech dokumentů naráz, skripty pro úpravu výsledných PDF (např. rozvržení stran na tiskové archy), atd.

2.6. Co se může ještě hodit?

Tak především je to interpret jazyka postscript zvaný `GhostSctript`. Kromě převodu souborů PS a PDF do formátů jazyků pro nejrůznější tiskárny je to také GNU obdoba známého Adobe Distilleru, takže umožnuje nejrůznější „kouzla“, s poscriptovými soubory.

Další programy pro úpravu PS/PDF souborů je řada utilit z balíku `PSUtils`. Jsou to např. `psnup` – umožnujě tisk více stran na jeden fyzický list papíru, `psresize` – pro změnu velikosti stránek či univerzální `pstops` a jiné.

2.7. Odkazy

- cvshome.org – Concurrent Version System
- www.winedt.com – Editor WinEdt
- vim.org – Vi Improved
- www.gnu.org – GNU Make
- www.cs.wisc.edu/ ghost – GhostScript