

Univerzita Hradec Králové

Fakulta informatiky a managementu

BAKALÁŘSKÁ PRÁCE

UNIVERZITA HRADEC KRÁLOVÉ
FAKULTA INFORMATIKY A MANAGEMENTU

L^AT_EX jako nástroj pro tvorbu šablon závěrečných prací na FIM UHK

BAKALÁŘSKÁ PRÁCE

Autor: Matěj Zmítka

Studijní obor: Aplikovaná informatika

Vedoucí práce: doc. Ing. Hana Tomášková, Ph.D.

Hradec Králové

18.8.2015

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne _____

podpis

Děkuji vedoucímu bakalářské práce za metodické vedení práce, poskytnutí cenných rad a neutuchající trpělivost.

Anotace

Práce se zabývá vytvořením šablony pro odevzdávání závěrečných prací v nástroji pro počítačovou sazbu $\text{\LaTeX}$. V první části se práce zabývá teoretickým popisem prostředí $\text{\TeX}$, jeho historií a principem fungování. Dále jsou zde vysvětleny příkazy pokrývající základní oblasti závěrečných prací a struktura, kterou by závěrečná práce měla mít. V druhé části je popsána samotná šablona, její části, obsah a způsob vytvoření.

Annotation

The goal of the bachelor thesis is creating templates for submitting theses in tools for desktop publishing $\text{\LaTeX}$. The theoretical part contains description of the $\text{\TeX}$ environment, it's history and principle of operation. It also explains the commands covering the basic areas of theses and structures that final work should have. The practical part describes the template itself, it's parts, content and method of creation.

Obsah

1	Úvod	1
2	Cíl a metodika	2
3	T_EX	3
3.1	Historie T _E Xu	3
3.2	Princip fungování T _E Xu	4
3.3	Uspořádání souborů	5
3.4	Rozšíření a balíky maker	6
3.4.1	PlainT _E X	7
3.4.2	AmST _E X	7
3.4.3	LuaT _E X	8
3.4.4	pdfT _E X	8
3.4.5	eT _E X	8
3.4.6	XeT _E X	9
3.4.7	L ^A T _E X	9

4	Struktura dokumentu	11
4.1	Příkaz <code>\documentclass</code>	11
4.2	Preamble	12
4.2.1	Styly stránek	13
4.3	Skupina, prostředí	14
4.4	Speciální znaky	14
4.5	Mezery	15
5	Analýza využitelných příkazů	16
5.1	Typy příkazů	16
5.1.1	Řídící slova	16
5.1.2	Řídící znaky	17
5.1.3	Vkládací příkazy	17
5.1.4	Přepínače	17
5.1.5	Příkazy s parametrem	17
5.2	Příkazy pro nastavení parametrů sazby	17
5.3	Příkazy pro vzhled stránky	19
5.3.1	Příkazy pro nastavení nadpisů, kapitol	19
5.3.2	Příkazy pro číslování	20
5.3.3	Poznámky pod čarou	21
5.3.4	Zvýraznění slov	21
5.4	Křížové odkazy	21
5.5	Příkazy pro jednotlivá prostředí	21
5.5.1	Výčty	22
5.5.2	Citace	22
5.5.3	Přímý výstup	22

5.6	Grafika	23
5.6.1	Prostředí picture	23
5.6.2	Standartní balík graphics	23
5.7	Tabulky	24
5.8	Matematické vzorce	25
5.9	Automatický obsah, seznam obrázků, tabulek	25
5.10	Seznam literatury	25
6	Struktura šablony	27
6.1	Struktura souborů šablony	27
6.2	Struktura souboru sablona.tex	28
6.2.1	Základní příkazy	29
6.2.2	Deklarace proměnných	29
6.2.3	Deklarace základních údajů dokumentu	30
6.2.4	Napojení jednotlivých listů	30
6.2.5	Bibliografie	32
6.2.6	Přílohy	32
7	Závěry a doporučení	33
	Zdroje	33
A	Příloha	i
A.1	Vlastní šablona na CD	i

1 Úvod

Šablona pro odevzdávání závěrečných prací by měla být nedílnou součástí každé vysoké školy, kde se nějaké závěrečné práce vypracovávají a odevzdávají. Studenti by měli mít představu, v jakém formátu práce vypracovávat a odevzdávat.

Tato bakalářská práce se zabývá vytvořením šablony v systému $\text{\LaTeX}$, což je nástavba sázecího systému $\text{\TeX}$, který po vysázení zajišťuje nejlepší typografickou kvalitu. V první části se zabývá teorií systému $\text{\TeX}$, přehledem jeho nástavbových balíčků a analýzou vybraných příkazů. Ve druhé části potom popisem vlastní vytvořené šablony, jejích částí, rozdělení, způsobu propojení jednotlivých částí do jednoho společného celku a věcí potřebných ke správnému vysázení šablony.

Dále popisem jednotlivých souborů, které se šablonou souvisejí, ale tvoří se automaticky systémem $\text{\LaTeX}$ při vysazování jednotlivých částí.

2 Cíl a metodika

Cílem této bakalářské práce je vytvoření šablony pro odevzdávání závěrečných prací v systému L^AT_EX. L^AT_EX je typografickým systémem, který je určen k sazbě vědeckých a matematických dokumentu vysoké typografické kvality.[5]

Šablona je určena pro studenty fakulty informatiky Univerzity Hradec Králové na pomoc při tvorbě jejich závěrečných prací. Vizuální podoba šablony je vytvořena podle metodických pokynů pro vypracování bakalářských a diplomových prací univerzity Hradec Králové.

Šablona je tvořena mnoha soubory, které jsou poté spojeny do jednoho základního souboru pojmenovaného *sablona.tex*. Úvodní část, která zahrnuje hlavičku, anotaci, poděkování a prohlášení je každá z těchto částí vytvořena, jako samostatný T_EX soubor, který je vložený do šablony pomocí odkazu. Dále je doporučeno podobně přidávat obsah jednotlivých kapitol pomocí stejného principu.

3 T_EX

T_EX je volně dostupný systém pro sazbu textu a matematických rovnic, při zachování vysoké typografické úrovně výsledného dokumentu.[5] Je vybaven sofistikovaným makrojazykem. T_EX vytvořil Donald E. Knuth v 70. letech minulého století.[8]

3.1 Historie T_EXu

Na začátku roku 1977 byl Donald Ervin Knuth nespokojen se sazbou své knihy „The Art of Computer Programming”, která byla připravována nedokonalou analogovou fotosazbou. O pár týdnů později se seznámil s možnostmi digitální fotosazby, změnil plány na příští rok a rozhodl se vytvořit jazyk počítačové typografie T_EX. Na podzim roku už byla hotová verze programu, která potvrdila možnost vytvoření takového programu. Pro nově vzniklý software bylo důležité ujmout se a získat budoucí uživatele. T_EX je určen především pro sazbu náročných matematických textů, a proto Knuth 4. ledna 1978 přednesl pro Americkou matematickou společnost přednášku, kde vyzdvihl možnosti a přednosti T_EXu.[4]

V srpnu 1978 se T_EX dostal k prvním uživatelům. Dnes je tato verze označována jako T_EX 78. O dva roky později, 22. února 1980, byla založena organizace, sdružující uživatele T_EXu, která se jmenuje TUG (T_EX Users Group) a téměř současně s jejím vznikem začal vycházet TUGBoat - časopis plně věnovaný problematice T_EXu.[4]

Další významnou verzí je T_EX82, který byl rozšířen o vylepšenou skupinu fontů Computer Modern. V roce 1989 byl Knuthem T_EX přepracován tak, aby mohl pracovat se vstupním souborem, který obsahuje znaky s kódem 0-255 (do té doby podporoval pouze 128 kódů). Tak vznikl T_EX 3.0, který je navíc multilinguální, tj. v jednom textu lze dělit slova podle pravidel různých jazyků.[4]

3.2 Princip fungování T_EXu

T_EX je formátor. Je to program, kterému předložíme vstupní text dokumentu v „holé“ textové podobě doplněný textovými značkami, které vymezují strukturu dokumentu nebo dávají T_EXu pokyny o způsobu formátování dokumentu.

T_EX tedy čte na svém vstupu textový soubor s dobře definovanou syntaxí jazyka značek a na výstupu je soubor s definitivním popisem sazby. T_EX jako takový je zcela nezávislý na operačním systému. Vývoj samotného T_EXu je zastaven, takže pro uživatele nehrozí nebezpečí vzniku dalších nekompatibilních verzí. Další programy „okolo T_EXu“ tvoří společně s T_EXem distribuci.[7]

Jádro systému tvoří překladač jazyka T_EX, jehož vstupem je textový soubor pořízený libovolným editorem (pro snadnější orientaci budeme předpokládat, že název tohoto souboru obsahuje rozšíření *.tex). Do takového souboru se vedle vlastního textu zapisují také příkazy, které určují, jak má být text vysázen.[9]

Příkazy zdrojového textu mohou patřit jak do množiny příkazů základního systému T_EX, tak do množiny příkazů nadstavby L^AT_EX. Vzhledem k tomu, že nadstavba pouze rozšiřuje možnosti základního systému, odpovídá celý princip činnosti víceméně chování systému T_EX.[9]

Pořízení zdrojového textu pro překladač lze provést libovolným programovým nebo textovým editorem, nebo textovým procesorem. Musí být splněna jen jedna podmínka. Soubor, který takto vznikne, nesmí obsahovat žádné řídicí znaky, znaky v horní polovině rozšířeného kódu ASCII (vyjma znaků národní abecedy), ani vnitřní příkazy dokumentního tvaru (soubor, do něhož si editor ukládá nejen text, ale i značky pro zpracování) použitého editoru. Není tedy vhodný takový editor, který automaticky do svého dokumentního souboru vkládá pomocné příkazy pro zobrazení na displeji nebo tiskárně (například Text602). U těchto editorů je potřebné vždy provést export do obyčejného tvaru ASCII.[9]

Hlavní prací překladače je rozmístění jednotlivých znaků do sazebního zrcadla. K tomu potřebuje znát rozměry jednotlivých znaků všech použitých znakových sad. Všechny znaky jsou v tomto okamžiku chápány jako obdélníky, jejichž rozměry jsou soustředěny v souborech s rozšířením *.tfm (TeX Font Metric). Průběh překladu a všechna důležitá

varovná a chybová hlášení překladač vypisuje na standardní výstup. Současně se vytváří textový soubor (jeho název má rozšíření *.log), obsahující podrobný výpis všech informací o průběhu překladu.[9]

Výstupem překladače je soubor s vysázeným textem (rozšíření *.dvi) , který je vytvořen tak obecně, aby jej bylo možno zpracovat na různých finálních zařízeních. Jeho obsah je tedy nezávislý na zobrazovacím zařízení (DeVice Independent). Abychom mohli uvidět vysázený text, je nutné tento soubor zpracovat dalším programem, který jej na daném zařízení zobrazí.[9]

Zobrazovací program v souboru *.dvi zjistí, kam má umístit jednotlivé znaky určité znakové sady. Tvary znaků však v souboru *.dvi nejsou obsaženy, zobrazovací program je musí přečíst z disku.[9]

3.3 Uspořádání souborů

Celý systém $\text{\TeX}$ je tvořen velkým množstvím různých souborů, které jsou nutné ke smysluplnému chodu systému. Z nich můžeme jmenovat tyto spustitelné (v závorce je uvedena běžně používaná přípona vstupního a výstupního souboru):

- vlastní program $\text{\TeX}$, který překládá zdrojový soubor (tex, dvi, log, toc, aux)
- program METAFONT, který vytváří grafický (a rastrový) tvar používaných fontů (mf; gf, tfm)
- zobrazovače, které za použití fontů zobrazí text tak, jak ho poskládal $\text{\TeX}$ (dvi)
- editor, ve kterém je vytvářen zdrojový soubor a který jej ukládá v ASCII 3 tvaru na disk, tento editor není vlastní částí systému $\text{\TeX}$ a může být jakýkoliv, jen nesmí vkládat své vlastní řídicí znaky (vyhovuje dosovský QEdit, EdLin, windowsovský NotePad, i většina vyšších editorů umožňuje export v ASCII). Zpravidla se používají speciální editory, které umí pomocí zkratkových kláves vkládat příkazy $\text{\TeX}$ u (například Q, CsEd, Vim, Emacs)
- pomocné programy pro METAFONT, které zajišťují např. komprimaci fontů apod.

- pomocné programy pro $\text{\TeX}$, které zajišťují jeho konfiguraci
- program METAPOST, který slouží k tvorbě vektorových obrázků (mp)

Další hromada souborů, které systému $\text{\TeX}$ náležejí, jsou buďto nespustitelné vstupní soubory (zpravidla textové), které jsou využívány už dříve uvedenými spustitelnými soubory, nebo výstupní soubory, které některý ze spustitelných programů produkuje a které využívá další program nebo uživatel. Z typických nespustitelných souborů lze uvést tyto:

- zdrojový soubor – zpravidla s příponou *.tex, který obsahuje obyčejný text s požadovaným textem a formátovacími příkazy
- zdrojový soubor METAFONTu s příponou *.mf, který obsahuje v textovém tvaru pokyny, jak mají vypadat jednotlivé znaky a jaké mají rozměry (pro různé velikosti písma mohou existovat různé soubory mf)
- metrický soubor *.tfm, který obsahuje rozměry obdélníku (boxu) souvisejícího s šířkou, výškou a hloubkou znaku (ne jeho vzhled), s tímto souborem pracuje $\text{\TeX}$, aby mohl určovat délky řádků apod., soubor je stejný pro jeden font všech velikostí
- soubor s rastry znaků *.gf, *.pk, které obsahují přímo vzhled znaků v rozlišení *. Soubory gf jsou zpravidla rozsáhlé, proto se komprimují do souborů *.pk. Tyto soubory jsou nezbytné pro funkci zobrazovače, většina zobrazovačů umí pracovat s oběma typy
- soubory *.fmt, které už byly zmíněny a obsahují binární zápis jednotlivých formátů[12]

3.4 Rozšíření a balíky maker

Autor $\text{\TeX}$ u Donald Knuth zmrazil v roce 1989 vývoj $\text{\TeX}$ u, neboť považuje za užitečné vytvořit jistý „pevný bod v čase“. Posléze vznikla různá rozšíření $\text{\TeX}$ u, která se nesmějí (dle přání autora) přímo nazývat $\text{\TeX}$, ale mohou slovo $\text{\TeX}$ ve svém názvu obsahovat. Vesměs všechna rozšíření přidávají další sadu primitivních příkazů a rozšiřují vlastnosti $\text{\TeX}$ u. [8]

3.4.1 PlainTeX

PlainTeX je soubor základních maker pro systém $\text{\TeX}$, který vytvořil autor původního $\text{\TeX}$ u Donald E. Knuth. PlainTeX nastavuje interpretaci znaků v matematickém prostředí. Jsou zde vesměs definovány praktické řídicí sekvence tak, aby bylo možné jednoduchý článek napsat v podstatě ihned a bez větších znalostí algoritmů $\text{\TeX}$ u. Jedná se o nejjednodušší nadstavbu původního $\text{\TeX}$ u a většina ostatních maker ji používá, jako základ. V roce 1989 Donald E. Knuth prohlásil, že jej nebude nijak rozšiřovat ani měnit, pouze opravovat chyby. Číslo verze konverguje k π a s každou opravou přibere jedno desetinné místo.

3.4.1.1 OPmac

OPmac je balík jednoduchých doplňujících maker k plainTeXu umožňující uživatelům základní $\text{\LaTeX}$ ovou funkcionalitu: změny velikosti písma, automatickou tvorbu obsahu a rejstříku, práci s bib databázemi, referencemi, možnost proložení referencí hyperlinkovými odkazy atd.[6]

Balík OPmac se skládá z těchto souborů: opmac.tex (makro), opmac-u.pdf (uživatelská dokumentace napsaná za použití opmac), opmac-d.pdf (technická dokumentace napsaná za použití DocByTeXu), opmac-u.tex a opmac-d.tex (zdrojové soubory k dokumentaci).[6]

Každé makro je napsáno s cílem co největší srozumitelnosti pro lidi, kteří to budou chtít číst a měnit. Je k dispozici uživatelská dokumentace a rozsáhlá technická dokumentace, která může sloužit jako učebnice programování maker v plainTeXu.[6]

3.4.2 AmSTeX

AmSTeX je balík tříd a maker vyvinutých Americkou matematickou společností (American Mathematical Society, AMS). Jedná se o přizpůsobení pro sazbu složitých kratších matematických dokumentů.

3.4.3 LuaTeX

LuaTeX propojuje vlastnosti T_EXu s procedurálním jazykem Lua. Autoři LuaTeXu dospěli zřejmě k názoru, že makrojazyk T_EXu je místy obtížně použitelný a že je rozumné T_EX rozšířit o jazyk, v němž by se daly dělat věci lépe. Jazyk Lua je v LuaTeXu prorostlý do původních algoritmů T_EXu a je možné tyto algoritmy na různých úrovních zpracování modifikovat, nebo dokonce přeprogramovat pomocí Lua kódů. LuaTeX bez použití Lua kódu se chová jako T_EX.[8]

Výhodou LuaTeXu je skutečnost, že nabízí mocný nástroj na zpracování sazby odvozený z T_EXu, a navíc se v něm dá programovat ve srozumitelném procedurálním jazyku. Nevýhodou LuaTeXu je fakt, že jeho dokumentace je mnohasetstránková, a aby ji člověk mohl začít číst, musí stejně nejprve dokonale zvládnout a pochopit vnitřní algoritmy klasického T_EXu.[8]

3.4.4 pdfTeX

PdfTEX je jedno z rozšíření T_EXu, které vytváří místo souborů *.dvi soubory *.pdf (resp. může vytvářet i *.dvi, záleží na nastavení proměnné \pdfoutput). Pro definování rozměrů stránky slouží proměnné \pdfpageheight a \pdfpagewidth.

Každý *.pdf soubor může obsahovat informace o autorovi dokumentu, jeho názvu a podobně. Tyto informace se do dokumentu vkládají pomocí příkazu \pdfinfo {položka (obsah)}, kde položka je některé „klíčové“ slovo (Title, Author, Keywords) s příslušným obsahem, dvojic položka–obsah se může v těle příkazu vyskytovat víc (ale jednoznačně).[12]

3.4.5 eTeX

Rozšiřující primitivní příkazy eTeXu jsou dnes využívány standardně v makrech L^AT_EXu. Rozšíření eTeX je dnes běžnou součástí binárních programů pdfTeX, XeTeX a LuaTeX ve většině T_EX ových distribucí. Pomocí eTeXu je možné zapisovat výpočet numerických a metrických hodnot přímočaěji. [8]

3.4.6 XeTeX

XeTeX vytvořil v roce 2004 Jonathan Kew. Nejprve pro MAC OS X a po dvou letech i pro Linux, odkud pramenil další port na MS Windows. XeTeX obsahuje všechny vlastnosti eTeXu (pokud jsou inicializovány při generování formátu přepínačem `-etex`) a navíc přidává dvě další zásadní odlišnosti od klasického TeXu:

- interně pracuje v Unicode, na vstupu předpokládá výhradně UTF-8 kódování. Zatímco klasický TeX umožňuje pracovat jen se znaky v rozsahu `\char0` až `\char255`, v případě XeTeXu máme rozsah úplné Unicode tabulky, tj. `\char0` až `\char1114111`.
- je slinkován s knihovnou operačního systému, která aplikacím umožňuje přistupovat k fontům instalovaným v systému. XeTeX umí pracovat jako klasický TeX s fonty v TeXové distribuci, ale kromě toho lze za příkazem `\font` napsat i název fontu instalovaného v operačním systému.[8]

3.4.7 L^AT_EX

Leslie A. Lamport vytvořil pro TeX sadu maker a pojmenoval ji L^AT_EX. Snažil se v ní vyjít vstříc běžným potřebám při sazbě dokumentů, proto zařadil příkazy pro členění textu do kapitol, generování obsahu či vkládání obrázků a tabulek.

L^AT_EX je předmětem nekonečného vývoje. Svého času se masově prosadila verze 2.09, zatížená řadou nedostatků. V roce 1990 byl proto zahájen vývoj verze 3 a jako dočasný mezistupeň k ní vytvořen L^AT_EX 2_ε. Jak už tak bývá, dočasnost se stává poněkud trvalou a po dvaceti letech je L^AT_EX 3 stále v nedohlednu. Aby byla nějakým způsobem odlišena stará verze od nové, tak byla označena L^AT_EX 2_ε. [8]

3.4.7.1 BibTeX

BibTeX je subsystém systému L^AT_EX, který umožňuje automatizovat vytváření seznamů literatury při psaní odborných publikací. Umožňuje oddělit úvahy o logické struktuře a o vizuální podobě i při psaní seznamů literatury. Jeho vstupem je bibliografická databáze (nebo více takových databází najednou), a dále bibliografický stylový soubor. Bibliografická databáze je textový soubor, ve kterém jsou údaje o knihách, člancích a

jiných bibliografických položkách zapsány v pevně definovaném formátu. Stylový soubor určuje, jakým písmem a v jakém pořadí mají být v seznamu literatury finálního dokumentu zapsány údaje o autorovi, časopisu, čísle, roku vydání atd. Určuje také, jak mají být vizuálně odlišeny například knihy od časopiseckých článků.[11]

3.4.7.2 Spolupráce BibTeXu a L^AT_EXu

Máme-li bibliografickou databázi, stačí vědět, jak se jmenují záznamy v ní uvedené, a můžeme se na ně při psaní svého zdrojového textu odkazovat pomocí příkazu `\cite`. Pak stačí do místa textu, kde má LaTeX vytvořit seznam literatury, napsat příkazy například `\bibliographystyleabbrv` `\bibliographylog,compsci` Tyto příkazy předpokládají, že ve své T_EXovské instalaci máte stylový soubor `abbrv.bst` a dvě bibliografické databáze `log.bib` a `compsci.bib` obsahující záznamy (knihy, články) pojmenované `papa:comp94` a `ladn:mode77`. BibTeX tedy ze sjednocení databází jmenovaných mezi složenými závorkami v argumentu příkazu `\bibliography` vybere ty záznamy, které se ve vašem dokumentu vyskytují jako argumenty příkazu `\cite`, sestaví z nich seznam literatury a dá mu úpravu podle pokynů uvedených ve stylovém souboru jmenovaném mezi složenými závorkami v argumentu příkazu `\bibliographystyle`. [11]

Požadavky na zařazení určitých položek do seznamu literatury našeho dokumentu, které LaTeXu sdělíme použitím příkazu `\cite`, LaTeX uloží do pomocného souboru, který má stejné jméno jako zdrojový text a typ `*.aux`. [11]

4 Struktura dokumentu

První věc, co $\text{\LaTeX}$ potřebuje vědět při zpracovávání vstupního souboru je informace o typu dokumentu, který hodláme vytvářet. Tato informace se nalézá na začátku každého dokumentu a autor ji předá $\text{\LaTeX}$ u příkazem `\documentclass`. Celý příkaz vypadá takto: `\documentclass[options]{class}`.

Každý dokument určený ke zpracování systémem $\text{\LaTeX}$ má tuto rámcovou strukturu:

```
\documentclass[volby]{třída} [RRRR/MM/DD]
\usepackage{czech,a4}
%Zde je preambule
\begin{document}
%Zde je text dokumentu.
\end{document}
```

4.1 Příkaz `\documentclass`

V povinném úvodním příkazu `\documentclass` je parametr třída, který definuje styl sazby, jakým má být dokument vypracován. Tento parametr zároveň určuje jméno souboru s rozšířením `.cls` (zkratka slova `class` - třída), v němž je definice třídy uvedena. K dispozici jsou tyto standardní třídy: `article` (pro odborné články, prezentace, krátké zprávy, pozvánky), `report` (pro delší zprávy obsahující několik kapitol, menší knihy, diplomové práce), `book` (pro knihy), `letter` (pro dopisy), `slides` (pro průsvitné fólie, tato třída užívá velkého bezpatkového písma, jež je pro zpětnou projekci nejvhodnější).[5]

Příkaz `\documentclass` může mít ještě volitelný parametr, jehož pomocí lze modifikovat činnost příkazů ve zvolené třídě. Opět existují předdefinované volby, z nichž některé jsou uvedeny v následujícím výčtu:

- `10pt`, `11pt`, `12pt` - Nastavuje velikost hlavního písma dokumentu. Není-li tato volba explicitně uvedena, použije se pro základní písmo velikost `10pt`.

- twoside, oneside - Určuje zda má být generován jedno či oboustranný výstup. Třídy article a report normálně užívají jednostranné a třída book oboustranné sazby. Jednostranná sazba od oboustranné se liší rozdílnou velikostí okraje pravé a levé strany pro vazbu případně umístěním čísla stránky.
- fleqn - matematické rovnice se budou zarovnávat vlevo namísto do středu.
- a4paper - nastavení formátu papíru na A4. Podobně existují volby aBpaper, bSpaper, letterpaper, legalpaper, executivepaper.
- titlepage, notitlepage - určuje zda se bude sázet samostatná titulní strana či nikoli. Třída article neužívá samostatnou titulní stranu, zatímco třídy report a book tak činí.
- landscape - výstup bude formátován „na šířku“, tj. rozměry výšky a šířky papíru budou navzájem zaměněny.[5]

V příkazu lze uvést několik voleb, které musí být odděleny čárkami bez mezer, například:

```
\documentclass[a4paper,12pt]{article} [datum vytvoření]
```

Nepovinný parametr datum vytvoření může být použit pro specifikaci nejstaršího požadovaného data vytvoření souboru .cis. Parametr obsahuje datum ve formátu RRRR/MM/DD. Je-li příslušný soubor .cis starší, je při překladu vypsáno varovné hlášení.[9]

4.2 Preamble

V úseku dokumentu nazvaném preamble se nacházejí příkazy, jejichž platnost se vztahuje k celému textu (například nastavení rozměrů stránky). Tyto příkazy se nazývají globální. Jejich uvedení v textové části buď nemá žádný význam, nebo může vést k neočekávaným výsledkům. V této části lze k dokumentu připojovat další balíky příkazů – packages (soubory s rozšířením .sty). Jejich příkazy pak mohou být v celém dokumentu použity. Připojení se provádí libovolným opakováním příkazu \usepackage, jehož sestava parametrů je uspořádáním i významem stejná jako u příkazu \documentclass:

`\usepackage [volby] {balík} [datum vytvoření]`

Balíky jsou v distribuci uloženy v souborech s rozšířením *.dtx. Tyto soubory v sobě ukrývají vlastní definice příkazů, ale také úplnou dokumentaci. Lze ji získat v kompletní podobě tím, že daný soubor přeložíme. Výsledkem překladu je pak příslušný soubor *.sty s příkazy a soubor *.dvi s dokumentací.[9]

Pro ilustraci uvedeme názvy některých standardních balíčků, které lze nalézt v distribuci (ve stručných popisech jsou užity pojmy, které budou vysvětleny později):

- color - podporuje práci v barvách. Ke své práci potřebuje podporu ovladače. Balík má volbu pro specifikaci příslušného ovladače, který je v dané instalaci k dispozici.
- czech - umožňuje zpracovávat dokumenty s českými prvky (české kódování vstupního textu, české názvy textových objektů - Kapitola, Obrázek, Tabulka a podobně).
- graphics - umožňuje geometrické transformace textu (zvětšování, rotace, zrcadlový obraz). Vyžaduje podporu ovladače, jehož specifikaci je možné uvést jako volbu tohoto balíku.
- graphpap - definuje příkaz `\graphpaper` použitelný v prostředí picture.
- ifthen - definuje příkazovou strukturu „if ... then ... else ...”.
- imputenc - užívá se pro definici vstupního kódování znaků
- makeidx - definuje příkazy pro tvorbu rejstříku.[9]

4.2.1 Styly stránek

L^AT_EX nabízí tři předdefinované kombinace záhlaví/paty stránek — tzv. stránkové styly. Parametr style příkazu:

`\pagestyle{style}`

definuje, který ze stránkových stylů se užije. Předdefinované styly stránek mohou být:

- plain - tiskne čísla stránek na spodním okraji stránky ve středu paty stránky. Toto je základní stánkový styl.
- headings - tiskne jméno aktuální kapitoly a číslo stránky v záhlaví každé stránky a pata stránky zůstává prázdná.
- empty - nastavuje prázdné záhlaví i patu stránky.[5]

4.3 Skupina, prostředí

Pojmem „skupina“ je označován úsek textu, který je ohraničen dvěma způsoby:

- složenými závorkami
- dvojicí příkazů `\begin{název}` a `\end{název}`. Za parametr název je dosazeno určité slovo s definovaným významem. Tato dvojice příkazů ohraničuje takzvané prostředí (v některé literatuře je ve stejném významu používán pojem okolí), což je úsek textu, který má být chápán jiným způsobem než okolní text. Největším prostředím je prostředí dokumentu, ohraničené příkazy `\begin{document}` a `\end{document}`. [9]

Některé příkazy ovlivňují sazbu následujícího textu až do konce skupiny nebo prostředí, v němž byly uvedeny. Hranicemi skupiny nebo prostředí tedy nastavujeme úsek vlivu příkazů tohoto typu.

Prostředí a skupiny se mohou do sebe libovolně vnořovat, nesmějí se však křížit.[9]

4.4 Speciální znaky

Pro strukturování dokumenty jsou také důležité speciální znaky, kterým byla přiřazena určitá funkce. Pokud chceme takový znak vysázet, musíme použít speciální příkaz.

Ve zdrojovém textu lze používat komentáře. Jsou zahájeny znakem `%` a `TeX` při zpracování ignoruje vše od znaku `%` až po konec řádku (včetně). Není k dispozici žádná

konstrukce, která by vyznačila začátek a konec komentáře – u dlouhých komentářů je třeba zahájit každý jejich řádek znakem %. Kromě vkládání interních poznámek se komentáře někdy využívají k vynechání konce řádků.

Další kategorií speciálních znaků jsou ligatury neboli slitky. Jedná se o dvojice znaků, které by bezprostředně za sebou nevypadaly dobře, a proto se nahrazují hezcím dvojznakem. Kdykoliv $\text{\TeX}$ narazí ve vstupu na příslušnou kombinaci znaků, nahradí je slitkem. Interně se slitky používají i v některých dalších případech, například k rozlišení pomlček. Typografové totiž znají tři: krátký silný spojovník (–) pro dělení slov, zvrtné -li a složená slova, pomlčku (–) ve větách a intervalech a dlouhou pomlčku (—), kterou používají ve větách američtí typografové. Ve zdrojovém kódu se zapisují jako jeden, dva nebo tři po sobě jdoucí znaky -, které se při sazbě slíjí do příslušné pomlčky.[8]

4.5 Mezery

„Neviditelné“ znaky jako mezera (blank), tabulátor či konec řádku (carriage return) jsou zpracovány $\text{\LaTeX}$ em jako mezera. Několik po sobě následujících neviditelných znaků, mezer, je zpracováno jako jedna mezera. Neviditelné znaky na počátku řádky jsou $\text{\LaTeX}$ em ignorovány. Prázdný řádek mezi textovými řádky označuje konec odstavce. Několik prázdných řádků je zpracováno jako jeden prázdný řádek.[5]

5 Analýza využitelných příkazů

Hlavní prací při realizaci publikace v systému L^AT_EX je zápis zdrojového textu společně s příkazy ovlivňujícími způsob sazby.

Systém rozlišuje v zápisu slovních příkazů velká a malá písmena – například zápisy `\large` a `\Large` mají odlišný význam.[9]

Za slovními příkazy, které nemají argumenty, se musí nacházet oddělovací neabecední znak, případně konec řádku. Je-li tímto oddělovačem libovolný počet mezer, jsou při zpracování textu pohlceny a neprojeví se na výstupu.[9]

Některé příkazy mohou být doplněny o argumenty (parametry). Parametry jsou uváděny ve složených, hranatých nebo oblých závorkách v závislosti na typu příkazu a druhu parametru. Pořadí jednotlivých parametrů je třeba bezpodmínečně dodržet v souladu s definicí příkazu. Obecně platí, parametr, který má být zapisován v hranatých závorkách, je nepovinný a lze jej vynechat (i se závorkami). Parametr ve složených závorkách je naopak povinný. Jsou-li vynechány složené závorky, je jako hodnota parametru brán jeden symbol (znak, příkaz) nacházející se bezprostředně za daným příkazem. Parametry v oblých závorkách jsou používány u zvláštní skupiny příkazů, například u příkazů pro kreslení obrázků.[9]

5.1 Typy příkazů

Podle způsobu zápisu se příkazy rozdělují do jednotlivých kategorií.

5.1.1 Řídící slova

Řídící slova jsou tvořena libovolně dlouhou posloupností písmen (anglické abecedy). Jako příklad může posloužit příkaz `\TeX`, který vysází logo T_EX. Řídící slovo končí prvním nepísmenným znakem. Pokud je tímto znakem mezera, bude řídícím slovem zpracována a ze vstupu zmizí (jiné znaky zůstanou zachovány).[8]

5.1.2 Řídící znaky

Řídící znaky obsahují jediný nepísmenný znak. Například #, kterým se sází znak „#“, je řídící znak. Taktéž řídící mezera zmíněná v předchozím bodu je řídícím znakem.[8] Většina jednoznakových speciálních znaků se vysází tímto způsobem.

5.1.3 Vkládací příkazy

Vkládací příkazy vloží do místa svého výskytu příslušnou typografickou konstrukci, jako například \TeX logo nebo \today dnešní datum.[8]

5.1.4 Přepínače

Přepínače změní určitý parametr sazby. Změna je trvalá a platí až do doby, kdy příslušný parametr změňte jiným přepínacím příkazem, nebo skončí skupina, v níž ke změně došlo. Typickým příkladem je \itshape, který přepne až do odvolání na kurzívu.[8]

5.1.5 Příkazy s parametrem

Příkazy s parametrem vytvoří určitou konstrukci kolem svého parametru. Ta se typicky týká jeho podoby při sazbě, může ale být i dost složitá, například příkaz \section{Příkazy}, kterým jsem zahajoval tuto sekci, vysází nadpis, vloží jej do obsahu a podobně. Parametr se uvádí bezprostředně za jménem příkazu a uzavírá se do složených závorek. Pokud má příkaz parametrů více, má každý své složené závorky. Některé příkazy mají i nepovinný parametr, bývá uveden jako první a zapisuje se v hranatých závorkách.[8]

5.2 Příkazy pro nastavení parametrů sazby

Příkazem \usepackage{czech} zajišťujeme sazbu českého textu, podobně příkazem \usepackage{slovak} sazbu slovenského textu.

Sazba je implicitně prováděna písmem Computer Modern. Změnu typu základního písma můžeme provést připojením balíku palatino, times nebo bookman pomocí příkazu `\usepackage`. Například `\usepackage{bookman}` nastaví základní písmo Bookman.[9]

Při procesu zarovnání dochází i k automatickému dělení slov. Dělení slov se provádí algoritmem, který používá slovník výjimek. Pokud dojde k chybnému rozdělení slova, máme možnost nařídit u daného slova jiné dělení příkazem `\-`. Můžeme například zapsat: `orga\-\ni\-\zace`. Toto slovo bude v případě potřeby rozděleno buď `orga-nizace`, nebo `organi-zace` a žádným jiným způsobem. Pokud používáme v textu často určité slovo, které může být pro dělicí algoritmus problémem, můžeme na začátek textové části zapsat příkaz `\hyphenation{5eOTom}` Parametr seznam zde představuje slovo nebo slova (oddělená mezerou), v nichž pomlčkami naznačíme místa, kde je možné provést v případě nutnosti jejich dělení, například

```
\hyphenation{orga-ni-zace při-bliž-ně vždyt'}
```

Poslední slovo v seznamu ukazuje jak řešit případ, kdy slovo nemá být děleno vůbec.[9]

Dělicí algoritmus automaticky zajišťuje, že každá část rozděleného slova má určitý minimální počet znaků. Implicitně musí být v první části slova alespoň dva znaky, ve druhé části slova alespoň tři znaky. Toto nastavení je možné v některých odůvodněných případech změnit (například při sazbě do úzkého sloupce). Minimální počet znaků první části slova je možné nastavit příkazem:

```
\lefthyphenmin=n
```

podobně minimální počet znaků druhé části slova příkazem:

```
\righthyphenmin=n
```

Dělení slov je závislé na zvoleném jazyce sazby. Může se stát, že například v českém textu je uveden úsek v angličtině nebo slovenštině. Pak je nutné změnit jazyk sazby.[9]

5.3 Příkazy pro vzhled stránky

V originální verzi L^AT_EXu jsou použity pro nadpisy kapitol, obsahu, označení tabulek a obrázku a názvy měsíců v datech anglické termíny. Jejich použití by bylo v českém textu krajně nevhodné. Použitím stylu czech.sty se tyto názvy automaticky změní na jejich české ekvivalenty.[5]

5.3.1 Příkazy pro nastavení nadpisů, kapitol

Pro lepší orientaci čtenáře v autorově díle, je vhodné dílo rozdělit na kapitoly, podkapitoly a oddíly. L^AT_EX umožňuje označit logickou strukturu díla speciálními příkazy, které jako argument mají název daného logického oddílu díla. Je pouze na autorovi, aby tyto příkazy užil ve správném pořadí.[5]

Ve třídě article lze užít následujících příkazů vymezujících logickou strukturu článku:

```
\section{...}  
\subsection{...}  
\subsubsection{...}
```

Třídy report a book rozpoznávají navíc tyto příkazy:

```
\part{...} pro část práce  
\chapter{...} pro kapitolu knihy
```

Velikost mezer mezi jednotlivými oddíly, velikost fontů a jednotlivých úrovní nadpisu je nastavena automaticky L^AT_EXem.[5]

Příkaz

```
\tableofcontents
```

zajistí vytištění obsahu. Nadpisy a čísla stran se dozví L^AT_EX z předcházejícího zpracování vstupního souboru. Proto při úpravách nebo doplnění kapitoly je nutno zpracovat

dokument programem $\text{\LaTeX}$ dvakrát (někdy i třikrát), aby bylo dosaženo správného výstupu.[5]

Všechny příkazy pro logické dělení dokumentu existují také v tzv. ohvězdičkové verzi, tzn. že se za jméno příkazu přidá znak hvězdička (*). Jméno oddílu vytvořeného příkazem v ohvězdičkové verzi se neočísluje a neuvede v obsahu. Například příkaz `\section{Předmluva}` v ohvězdičkové verzi bude vypadat takto `\section*{Předmluva}`. [5]

5.3.2 Příkazy pro číslování

Příkazem `\pagenumbering` lze ovlivnit způsob výpisu čísel stránek. Protože se jedná vlastně o výpis hodnoty čítače stran, lze do parametru tohoto příkazu uvést možnosti používané pro výpis obsahu čítačů. Zapišeme-li tedy `\pagenumbering{roman}`, budou čísla stránek zapsána římskými číslicemi. Implicitně je zvoleno arabské číslování, které se nastavuje příkazem `\pagenumbering{arabic}`.

Příkaz `\pagenumbering` vždy nastaví číslo stránky na 1.[9]

Každé číslo, které $\text{\LaTeX}$ samostatně vypisuje, je spojeno s nějakým čítačem. Čítač je obdoba celočíselné proměnné v programovacím jazyce - místo v paměti, které uchovává celočíselnou hodnotu. Čítače mají svá jména. Některé jsou předdefinovány pro specifický účel, jiné si může uživatel definovat sám a použít je podle potřeby.[9]

Příkazem `\setcounter` můžeme vložit do čítače zvolenou hodnotu. Například hodnotu 4 vložíme do čítače stránek zápisem `\setcounter{page}{4}`. Příkazem `\addtocounter` můžeme přičíst k dosavadní hodnotě čítače zvolenou hodnotu. Zvětšení čítače `page` o 2 provedeme tedy příkazem `\addtocounter{page}{2}`.

Pro výpis čísla vytváří systém $\text{\LaTeX}$ ke každému čítači příkaz, jehož název je tvořen předponou `\the`. Za ni se připojí jméno čítače (například `\thesection` pro výpis čísla sekce).[9]

Chceme-li vytvořit vlastní čítač, použijeme příkaz `\newcounter`. Zápisem `\newcounter{mujcitac}` se vytvoří čítač a zároveň s ním i příkaz `\themujcitac`, který umí vypsat hodnotu čítače arabskými číslicemi. Vytvořený čítač i s ním spojený příkaz mají globální charakter - je-li definice provedena uvnitř nějaké skupiny, je obojí dosažitelné i za touto skupinou.[9]

5.3.3 Poznámky pod čarou

Příkazem `\footnote{text poznámky pod čarou}` lze vysázet poznámku pod čarou, tj. text vysázený v dolní části stránky oddělený od hlavního textu horizontální čarou a označený stejnou značkou jako odkaz na tuto poznámku v textu.[5]

5.3.4 Zvýraznění slov

V rukopisech psaných na psacím stroji se důležitá, zdůrazněná, slova podtrhují. V tištěných knihách se však tato slova zvýrazňují. Sazba se do zvýrazněného typu písma přepíná příkazem `\emph{text}`. Argumentem tohoto příkazu je text, který má být zvýrazněn.[5]

5.4 Křížové odkazy

V knihách, referátech a článcích jsou velice běžné křížové odkazy na obrázky, tabulky, rovnice či jiné oddíly textu.

L^AT_EX nabízí pro křížové odkazy pohodlný aparát, totiž příkazy:

`\label{návěští}`, `\ref{návěští}` a `\pageref{návěští}`,

kde návěští je jednoznačné označení vybrané uživatelem. L^AT_EX zamění `\ref` číslem takového oddílu, pododdílu, obrázku, tabulky či rovnice, jež byl označen odpovídajícím příkazem `\label`, tj. příkazem `\label`, který má jako argument stejné návěští jako příslušný odkaz `\ref`. Namísto `\pageref` se vysází číslo stránky, na níž se nachází odpovídající příkaz `\label`. Číslo stránek L^AT_EX získává z pomocného souboru vytvořeného v předchozím průběhu překladu programem T_EX, proto i při použití křížových odkazů je někdy třeba překládat vstupní soubor programem T_EX několikrát.[5]

5.5 Příkazy pro jednotlivá prostředí

K označení zvláštních částí textu, které mají být vysázeny jinak než běžný text, slouží tzv. prostředí. Prostředí ve vstupním textu mají tvar:

```
\begin{jmeno}  
text  
\end{jmeno}
```

kde jméno je jméno daného prostředí. Platí, že prostředí se mohou vyskytovat jedno v druhém, přičemž je třeba dbát na správné pořadí. Jednotlivá prostředí se nemohou křížit.[5]

Základní příkaz je zde `\item`, jenž v každém jednotlivém prostředí znamená začátek položky výčtu.

5.5.1 Výčty

Prostředí `itemize` je vhodné pro jednoduché výčty, prostředí `enumerate` pro číslované výčty a prostředí `description` pro popisné výčty.[5]

5.5.2 Citace

Pro kratší citace, zdůrazněné věty a příklady se používá prostředí `quote`.

Vedle prostředí `quote` existují další dvě prostředí: `quotation` a `verse`. Prostředí `quotation` je vhodné pro citace, které zahrnují několik odstavců, jelikož narozdíl od `quote` odsazuje `quotation` jednotlivé odstavce. Prostředí `verse` je užitečné pro sazbu poezie, kde je dělení na řádky předem známé. Jednotlivé verše se oddělují příkazem `\\` na konci řádku a jednotlivé sloky prázdným řádkem mezi slokami.[5]

5.5.3 Přímý výstup

Text, který je uzavřen mezi `\begin{verbatim}` a `\end{verbatim}` bude vysázen přesně tak, jak byl zapsán ve zdrojovém souboru, tj. se všemi mezerami, tisknutelnými symboly, konci řádku a bez interpretace všech příkazů $\text{\LaTeX}$ u, tedy samozřejmě kromě `\end{verbatim}`. Toho lze s výhodou použít např. pro vytištění výpisu krátkých počítačových programů.[5]

5.6 Grafika

TeX měl být multiplatformní a produkovat všude přesně stejné výsledky. Do tohoto konceptu grafika příliš nezapadá, protože grafické schopnosti různých systémů a prostředí se výrazně liší. Původní TeX proto grafiku jednoduše nepodporoval.[8]

L^ATeX se pokusil o řešení, které by zachovávalo nezávislost na platformě. Přišel s prostředím `picture`, ve kterém jste různými příkazy mohli kreslit úsečky, kružnice a podobně.[8]

Většina implementací postupně nabídla nad rámec původního standardu podporu pro vkládání obrázků v běžně používaných grafických formátech, ovšem každá po svém. K odstranění rozdílů mezi implementacemi vznikly balíky, které nabízejí jednotné prostředky pro práci s grafikou. Nejpoužívanější je `graphics`. Oba mají podobné schopnosti, liší se jen struktura jejich příkazů.[8]

5.6.1 Prostředí `picture`

Prostředí je určeno pro kreslení vektorových obrázků složených z přímých čar, šipek, kružnic, oválů a libovolných textů. Uvnitř tohoto prostředí se mohou objevit pouze příkazy pro vložení obrazového objektu (`\put` a `\multiput`) a příkazy pro nastavení velikosti a typu písma nebo délek. Prostředí `picture` začíná úvodním příkazem, v němž je obsažena definice celkové velikosti obrázku a pozice počátečního bodu. Například

```
\begin{picture}(50,30)(10,20)
```

znamená, že obrázek bude široký 50 jednotek, vysoký 30 jednotek a jeho levý dolní roh se nachází na souřadnicích (10,20). Všechny souřadnice uvedené uvnitř prostředí se vztahují k bodu (0,0).[9]

5.6.2 Standartní balík `graphics`

Ve standardním balíku `graphics` se předpokládá existence ovladače, který je schopen pracovat s obrázky. Standardně se jedná o zpracování grafiky zapsané v jazyce PostScript. Pro vložení obrázku je zde definován příkaz

`\includegraphics{jmenosouboru}`

Předpokládá se, že obsah souboru je zpracovatelný použitým ovladačem. Ovladač je možné zvolit ze seznamu instalovaných ovladačů podle konkrétní situace na konkrétním počítači. Příkaz `\includegraphics*` je schopen z obrázku vyříznout obdélník, jehož levý dolní roh a pravý horní roh je definován volitelnými parametry.[9]

Například příkazem

```
\includegraphics*[5mm,7mm][25mm,17mm]{graf.ps}
```

se provede vložení obrázku ve tvaru obdélníka o velikosti 20x10 mm, který byl vyňat z většího obrázku tak, že levý dolní roh výstřižku je vzdálen 5 mm horizontálně a 7 mm vertikálně od levého dolního rohu původního obrázku a pravý horní roh výstřižku je vzdálen 25 mm horizontálně a 17 mm vertikálně od levého dolního rohu původního obrázku.[9]

Součástí balíku graphics jsou dále příkazy, které manipulují s obrazem vysázeného textu a umožňují zvláštní efekty. Jde o příkazy:

- `\scalebox` - změna velikosti textu v ose x a y
- `\resizebox` - vyplnění textu do připraveného obdélníka
- `\rotatebox` - otočení textu o zadaný úhel
- `\reflectbox` - zrcadlové převrácení textu [9]

5.7 Tabulky

Sazba tabulek patří mezi náročnější typografické disciplíny. L^AT_EX ji zvládá, ovšem s řadou různých omezení. Proto existuje celá řada rozšiřujících balíčků, jež doplňují nejrůznější schopnosti. Základem pro tabulky je prostředí tabular, jehož povinným argumentem je specifikace sloupců. V základní podobě každému sloupci v ní odpovídá jedno z písmen l, r, c podle toho, zda má být sloupec zarovnán doleva, doprava nebo na střed. Uvnitř tabular je obsah tabulky zapsán po řádcích. Jednotlivé sloupce jsou vždy odděleny znakem & a každý řádek je zakončen příkazem `\\`. [8]

5.8 Matematické vzorce

Sazba matematiky probíhá ve specializovaném, matematickém režimu. Přesněji řečeno jsou matematické režimy k dispozici hned dva – jeden pro vzorce v textu, a druhý pro samostatné vzorce. Ten první je kompaktnější ve svislém směru, aby výšku obklopujícího řádku změnil co nejméně. Vzorce v textu jsou oficiálně uzavřeny do prostředí `math`, které ale skoro nikdo nepoužívá, protože jsou k dispozici kratší varianty. Bud' můžete před vzorec vložit `\(` a za něj `\)`, valná většina autorů ovšem sáhne po nejstručnější variantě, jíž je znak `$`. Opět použijte po jednom před vzorcem a po něm. Pro samostatný vzorec je k dispozici prostředí `displaymath`, ale výrazně častěji potkáte kompaktnější formy `\[...]` nebo `$$...$$`. [8]

5.9 Automatický obsah, seznam obrázků, tabulek

Výhoda $\text{\LaTeX}$ u je v tom, že vašemu dokumentu rozumí a po prvním „průchodu“ vaším dokumentem již ví, na jaké části ho máte rozdělen. Proto je schopen vytvořit automaticky obsah. K jeho vytvoření je však nutné zkompilevat dokument dvakrát (nejdříve si $\text{\LaTeX}$ vytvoří pomocné soubory a podruhé je načte a obsah vytvoří). [2]

- `\maketitle` – vytvoří titulek
- `\tableofcontents` – vytvoří obsah
- `\listoftables` – vytvoří seznam tabulek
- `\listoffigures` - vytvoří seznam obrázků

5.10 Seznam literatury

Ve zdrojovém textu je celý obklopen prostředím `thebibliography`. U jeho zahájení musíte uvést povinný argument odpovídající nejširší značce. Jedná se o libovolný řetězec, který nevytvoří žádný viditelný výstup. $\text{\LaTeX}$ si pouze změří, jak je široký, a podle něj

dimenzuje vodorovnou mezeru určenou pro značky položek. Pokud je čísujete, obvyklou hodnotou bývá {99} (dvě číslice by měly stačit). [8]

Každá položka v seznamu začíná příkazem `\bibitem{identifikátor}`, která vytvoří pro nově zahájenou publikaci vizuální značku a zároveň jí přiřadí identifikátor, jehož pomocí se na ni můžete odkázat. Standardní generovanou značkou je pořadové číslo.[8]

6 Struktura šablony

Samotná šablona se skládá z mnoha souborů, ze kterých některé tvoří samotný uživatel a dále souborů, které se tvoří automaticky při vysázení textu, nebo prostě jen při použití některých součástí L^AT_EXu. Zde budou popsány jak části samotné šablony, tak všechny soubory, které šablona jako celek obsahuje, způsob a důvod jejich vytvoření.

6.1 Struktura souborů šablony

Soubory, které L^AT_EX vytváří automaticky se ukládají do stejné složky, kde se nalézá i samotná šablona. Šablona také pro větší přehlednost obsahuje tři podsložky pojmenované bibliografie, obsah a startKapitoly.

Ve složce startKapitoly jsou uloženy jednotlivé soubory, které takto vytvořené tvoří základní část šablony. Zde jsou uloženy listy hlavičky, anotace, poděkování a prohlášení, které obsahuje každá bakalářská a diplomová práce. Tyto listy jsou již vytvořené podle metodických pokynů pro vypracování bakalářských a diplomových prací fakulty informatiky a managementu Univerzity Hradec Králové a nedoporučuje se na nich cokoli měnit.

Do složky obsah by pak uživatel měl pro větší přehlednost ukládat jednotlivé kapitoly své bakalářské, či diplomové práce. Složka již obsahuje nějaké soubory, aby bylo patrné, jakým stylem s ní pracovat a co přesně sem ukládat.

Složka bibliografie je připravena pro uživatelskou bibTeX databázi, čili pro databázi zdrojů, kterou bude používat pro tvorbu svojí bakalářské, či diplomové práce. Pro příklad je zde zase už základní BibTeX databáze uložena pro lepší pochopení účelu této složky.

Základní soubor, který obsahuje celou šablonu, a do kterého jsou linkovány jednotlivé listy a kapitoly z předcházejících složek se nazývá sablona.tex.

Dále základní složka obsahuje soubor sablona.aux. Tento soubor obsahuje informace o dokumentu, jako jsou poznámky pod čarou, položky bibliografie a křížových odkazů. Soubory *.aux jsou vytvořeny, když je soubor s příponou *.tex vysázen systémem

L^AT_EX. Vzhledem k tomu, generace L^AT_EX dokumentů může trvat několik průchodů, než dokument je kompletní, soubor s koncovkou *.aux se používá k ukládání informací mezi běhy procesu kompilace L^AT_EXu.

Když spustíme BiBTeX na jméno souboru, zde na soubor sablona.tex, tak se na základě souboru sablona.aux vytvoří soubor sablona.bbl, jehož obsahem je celé prostředí \thebibliography. Při vytváření souboru .bbl využije BibTeX informace v bib souboru. Způsob, jakým BibTeX přesně řádky v *.bbl vytvoří, závisí i na předepsaném stylu specifikovaném v L^AT_EXovském zdrojovém souboru příkazem \bibliographystyle{.}[10]

Složka obsahuje také soubor sablona.blg, který obsahuje log zpracovaných citací.

Při vysázení se také v systému L^AT_EX automaticky vytvoří soubor sablona.pdf, který obsahuje vytvořený dokument v pdf formátu. Toto se děje pomocí již zabudovaného systému pdfTeX, který L^AT_EX obsahuje. PdfTeX ještě vytváří soubor sablona.txt, který obsahuje interní informace pro převod do pdf formátu.

Při prvním běhu L^AT_EXu se také vytvoří soubor s příponou *.toc. Při dalším spuštění se v místě, kde je umístěn příkaz \tableofcontents vypíše právě z tohoto souboru obsah dokumentu. Dle stejného principu tvoří L^AT_EX soubory s příponou *.lof a *.lot, které obsahují informace pro příkazy \listoffigures a \listoftables.

Poslední soubor, který šablona generuje sama je archiv sablona.synctex. Tento archiv obsahuje soubor sablona.synctex. Tento soubor zajišťuje synchronizaci mezi vlastním dokumentem ve formátu T_EX a jeho převedením do formátu pdf.

Poslední soubor, který L^AT_EX negeneruje, ale je tam přidán autorem, je soubor dict-cs-2.oxt. Tento soubor je český slovník, který se importuje do TeXStudio, nebo do jakéhokoliv jiného softwaru, kde pomocí této šablony tvoříte dokument. Je důležitý pro správné využití této šablony.

6.2 Struktura souboru sablona.tex

Soubor sablona.tex je hlavní stavební pilíř celé šablony pro odevzdávání bakalářských a diplomových prací. Obsahuje nastavení pro jednotlivé listy dokumentu, jejich celkové propojení, základní příkazy pro L^AT_EX, díky kterým je možné šablonu

vůbec otevřít a využívat, dále vysazovat české znaky, zaručení jejich pochopení systémem $\text{\LaTeX}$ či příkazy pro vykreslování obrázků. Dále obsahuje deklaraci proměnných, které uživatel pouze změní na vlastní údaje a ty se potom samy automaticky zobrazí na příslušných místech v hlavičce a úvodu bakalářské či diplomové práce. To zahrnuje jak jméno autora práce, jméno vedoucího práce, datum a rok vypracovávání práce.

6.2.1 Základní příkazy

Na začátku každého $\text{\LaTeX}$ ového dokumentu je nutné nalinkovat balíčky. Jelikož samotný $\text{\LaTeX}$ neumí sázet české znaky sám od sebe, je nutné nalinkovat balíček `\usepackage[czech]{babel}`, který řeší právě počestění předdefinovaných názvů kapitol či sekcí, dále dělení českých slov a sázení kalendářních dat.

Dále balíček `imputenc`, linkovaný pomocí příkazu `\usepackage[latin2]{inputenc}` říká překladači, v jakém kódování byl dokument napsán. Překladač potom vybere správnou kódovou stránku nezávisle na tom, jestli překládáme v Linuxu nebo ve Windows. Použití tohoto balíku tedy zvyšuje přenositelnost dokumentů. Zakomentované balíčky jsou připraveny pro uživatele, kteří budou potřebovat jinou znakovou sadu.

Pro veškerou grafiku, vykreslování a manipulaci s obrázky je nutné připojit balíček `graphics` příkazem `\usepackage{graphics}`.

Jelikož anglicky psané dokumenty neodsazují první odstavec následující po nadpisu, je nutné pomocí příkazu `\usepackage{indentfirst}` připojit balíček `indentfirst`, který zajišťuje, že se odstavec následující po nadpisu odsadí tak, jak požadují česky psané dokumenty.

Poslední balíček zde použitý je balíček `fancyhdr`, který umožňuje dokonalejší ovládání hlavičky a patičky na jednotlivých stránkách dokumentu.

6.2.2 Deklarace proměnných

V další části šablony probíhá deklarace proměnných, pomocí kterých jsou vytvořeny listy hlavičky, a celé první části každé bakalářské a diplomové práce. Na tomto místě uživatel pouze změní obsahy složených závorek za jednotlivými proměnnými a ty

se pak automaticky změní v hlavičce šablony. Celá hlavička šablony je předdefinována pomocí proměnných, tudíž uživatel do nich vůbec nemusí zasahovat.

Dále jsou zde definovány cesty jednotlivých částí šablony z podsložek pro zjednodušení a zkrácení dalších příkazů. Z tohoto důvodu je souborová část šablony rozdělena do podsložek.

6.2.3 Deklarace základních údajů dokumentu

V této části šablony se deklarují základní ustanovení pro vzhled stránek, nebo upravují velikosti částí, které jsou v systému nastaveny jinak, než potřebujeme.

Příkaz `\sloppy` zapřičiňuje to, že text se nám nikdy nedostane z oblasti, která je k tomu vyhrazena. Pomocí příkazu `\linespread` se nastabuje mezera mezi jednotlivými řádky textu.

Dále následují příkazy pro nastavení, až do jaké hloubky bude obsah vypisovat jednotlivé nadpisy. Příkazy `\setcounter{tocdepth}{5}` a `\setcounter{secnumdepth}{5}` nastavujeme, že až do hloubky 5 podnadpisů se budou nadpisy číslovat a vypisovat v table of content.

Pomocí příkazů `\setlength` měníme velikost částí na stránce pro text, jeho odsazení a velikost hlavičky a příkazem `\parskip` velikost odsazení prvního řádku jednotlivých odstavců.

Následujícím souborem příkazů nastavujeme vlastnosti vysazení jednotlivých kapitol, velikost místa nad ní, Velikost místa, které bude mezi kapitolou a následujícím textem a styl číslování kapitol.

6.2.4 Napojení jednotlivých listů

V další části šablony jsou postupně, jelikož $\text{\LaTeX}$ zpracovává soubor od shora dolů, nalinkovány stránky, které obsahuje každá bakalářská a diplomová práce. Příkaz `\pagestyle{empty}` zajišťuje, že se tyto stránky nečíslují. Zde jsou tedy pomocí příkazu `\input` nalinkovány soubory `Hlavicka_desky.tex`, který obsahuje titulní stranu patřící na desky bakalářské, či diplomové práce, `Hlavicka.tex`, která obsahuje první stránku

práce, informace o autorovi, vedoucím práce, názvu práce atd. Dále stránky Prohlášení, poděkování a anotace, které už si může uživatel upravit dle libosti.

Jako poslední stránka této části se automaticky vygeneruje obsah pomocí příkazu `\tableofcontents`. Za všemi těmito linky je příkaz `\newpage`, který zadává konec stránky a začne novou.

Poté už jsou příkazem `\input` postupně přidávány jednotlivé kapitoly, které jsou už pomocí příkazu `\pagenumberingarabic` číslovány. Kapitoly jsou linkovány ze složky obsah, kam by měly být ukládány pro uživatelskou přehlednost.

6.2.4.1 Hlavicka_desky.tex

V tomto souboru je vytvořena první stránka, která obsahuje informace, které se tisknou na titulní desku bakalářské, či diplomové práce. V prostředí ohraničeném `\begin{center}` a `\end{center}`, které zajišťuje zarovnání na střed, jsou pomocí proměnných doplněny informace z hlavního souboru na místa dle metodických pokynů pro vypracovávání bakalářských a diplomových prací. Příkaz `\textsc` vytváří textové pole.

Příkaz `\vfill` zajišťuje vyplnění volným místem vertikálně místo mezi dvěma elementy.

6.2.4.2 Hlavicka.tex

V tomto souboru je vytvořena první stránka bakalářské, či diplomové práce. V prostředí ohraničeném příkazy `\begin{center}` a `\begin{end}` jsou zarovnány na střed proměnné, které sem přenáší z hlavního souboru `sablona.tex` pole `NazevSkoly` a `NazevFakulty`, dále 7 centimetrů volného místa pomocí příkazu `\vspace{7cm}` a zase pomocí proměnných doplněn název bakalářské, či diplomové práce.

Dále v prostředí mezi příkazy `\begin{flushleft}` a `\end{flushleft}` zarovnány k levému okraji informace o autorovi, vedoucím práce a studijním oboru.

Poté doděláno zápatí stránky obsahující nalevo místo a napravo datum zase linkované proměnnými z hlavního souboru. Příkaz `\hfill` vyplní volným místem prostor mezi dvěma elementy.

6.2.5 Bibliografie

Bibliografie je zde připojena pomocí příkazu `\bibliography{\cestaZdrojeZdroje.bib}`. Připojuje se zde soubor `Zdroje.bib`, který obsahuje databázi zdrojů, ze kterých daná bakalářská, či diplomová práce čerpá.

Pomocí příkazu `\addcontentsline {toc}{chapter}{Zdroje}` je pak bibliografie doplněna do obsahu, který je pomocí příkazu `\tableofcontents` generován automaticky. V prvních složených závorkách se říká do kterého seznamu se má tato položka doplnit, druhé složené závorky doplňují informaci o jaký typ položky se jedná a ve třetích složených závorkách se říká pod jakým názvem se tam má tato položka objevit.

6.2.6 Přílohy

Každá příloha je zase vytvořena jako samostatný `.tex` dokument, který je pomocí příkazu `\input{\cestaObsah Priloha1.tex}` připojena za hlavní část dokumentu. V `LATEXu` je pro přílohy vytvořeno prostředí které je ohraničeno příkazy `\begin{appendix}` a `\end{appendix}`. Mezi tyto příkazy je možno připojit libovolné množství příloh. Tyto příkazy se také postarají o zvýraznění příloh.

Pomocí příkazu `\pagenumbering{arabic}` se přílohy číslovají opět od 1 dle metodických pokynů pro vypracovávání bakalářských a diplomových prací.

7 Závěry a doporučení

Cílem této bakalářské práce bylo analyzování prostředí a vypracování šablony závěrečných prací v nástroji pro počítačovou sazbu $\text{\TeX}$, resp. v jeho nástavbě $\text{\LaTeX}$, dle metodických pokynů pro vypracování bakalářských a diplomových prací.

V první části byl popsán systém $\text{\TeX}$, jeho historie, vývoj, balíky maker, a jeho smysl a použití. Dále zde byla popsána struktura dokumentu, který se pomocí systému $\text{\TeX}$ vytváří. Pro úspěšné zpracování bakalářské práce bylo nutné pochopit celou problematiku vytváření dokumentu v systému $\text{\TeX}$, proto zde byly popsány příkazy pro zpracování jednotlivých oblastí, které se mohou objevit v závěrečných pracích.

Dále zde byla teoreticky popsána struktura samotné vypracované šablony, její celková podoba, i její jednotlivé části a jejich propojení. Dále také jednotlivé soubory, které systém vytvoří při vysázení.

V praktické části byla vypracována šablona, která svou titulní stranou a svým obsahem odpovídá metodickým pokynům FIM UHK. Šablona je kompletní a připravena posloužit všem studentům, kteří by ji chtěli v budoucnu použít jako pomůcku pro psaní a odevzdání své závěrečné práce.

Literatura

- [1] Groza Tudor, Handschuh Siegfried, Möller Knud, Decker Stefan. SALT - Semantically Annotated L^AT_EX for Scientific Publications. 2007 ISBN 978-3-540-72667-8
- [2] Hordějčuk Vojtěch. Základy jazyka L^AT_EX. [online], 1999. Dostupné z: <http://voho.cz/wiki/latex/>
- [3] Indian T_EX Users Group L^AT_EX Tutorials. A Primer [online], 2003. Dostupné z: <https://www.tug.org/twg/mactex/tutorials/ltxprimer-1.0.pdf>
- [4] Kosek Jiří. Co je to T_EX. [online], 1999. [cit. 2015-08-20] Dostupné z: <http://www.kosek.cz/clanky/cw/cojetex.html>
- [5] Oetiker Tobias, Partl Hubert, Schlegl Elisabeth, Hyna Irene, Kočer Michal, Sýkora Pavel. Ne příliš stručný úvod do systému L^AT_EX 2_ε „public domain“ dokument. [online], 1998. Dostupné z: <http://www.penguin.cz/~kocer/texty/lshort2e/lshort2e-cz.pdf>
- [6] Olšák Petr. Opmac – rozšiřující makra plaintexu. [online], s. 20, 2012. Dostupné z: <http://petr.olsak.net/ftp/olsak/opmac/opmac-u.pdf>
- [7] Olšák Petr. První setkání s T_EXem. [online], 1999. [cit. 2015-04-21] Dostupné z: <http://math.feld.cvut.cz/olsak/ftp/cstex/doc/prvni.pdf>
- [8] Olšák Petr. T_EX pro pragmatiky. [online], 2013. [cit. 2015-04-21] Dostupné z: <http://petr.olsak.net/ftp/olsak/tpp/tpp.pdf>
- [9] Rybička Jiří. L^AT_EX pro začátečníky. 3. vydání, Konvoj Brno, 2003. ISBN 80-730-2049-1

- [10] Sopuch Pavel. Manuál - L^AT_EX. [online], 2001. Dostupné z:
<http://www.it.cas.cz/manual/latex/index.html>
- [11] Švejdar Vítězslav. O bibliografickém systému bibTeX. [online], 2005. Dostupné z:
<http://www1.cuni.cz/~svejdar/texdev/BibTeXInfo.pdf>
- [12] Unknown author Drsný úvod do L^AT_EXu: aneb moc se s tím nemazlete. [online], 2005. Dostupné z: <http://apfyz.upol.cz/ucebnice/down/mini/drslat.pdf>
- [13] Wenneker Frits. How to T_EX. [online], 2012. Dostupné z:
<http://www.howtotex.com/templates/>

A Příloha

A.1 Vlastní šablona na CD

K této práci je přiloženo CD se zdrojovými kódy šablony.