

Tvorba rejstříku v \LaTeX u
Creation of the index in \LaTeX

Bakalářská práce

Ondřej Beníček

Vedoucí bakalářské práce: Mgr. Jiří Pech, Ph.D.
Jihočeská univerzita v Českých Budějovicích
Pedagogická fakulta
Katedra informatiky
2009

Prohlášení

Prohlašuji, že svoji bakalářskou práci jsem vypracoval samostatně, pouze s použitím pramenů a literatury uvedených v seznamu citované literatury.

Prohlašuji, že v souladu s § 47b zákona č. 111/1998 Sb. v platném znění souhlasím se zveřejněním své bakalářské práce, a to v nezkrácené podobě pedagogickou fakultou elektronickou cestou ve veřejně přístupné části databáze STAG provozované Jihočeskou univerzitou v Českých Budějovicích na jejích internetových stránkách.

V Českých Budějovicích dne

Anotace

Tato bakalářská práce seznamuje uživatele programu \LaTeX s tvorbou knižních rejstříků. Čtenář se zde dozví, jak psát dokumenty v \LaTeX u a jak k vlastnímu dokumentu připojit rejstřík. Dále je zde popsáno, jakými způsoby lze upravovat rejstřík a jeho položky. Práce je určena pro každého uživatele systému \LaTeX , ale především autorům odborných knih a obdobných publikací. V příloze se nacházejí zdrojové kódy.

Abstract

This bachelor work introduces creating of book indexes to \LaTeX programme users. The reader will learn how to write documents in \LaTeX and how to enclose an index to his/her own document. Next the ways of index and it's modifications are described here. The work is designed for all \LaTeX programme users, but especially for writers of specialised books and similar publications. In the enclosure you can find source codes.

Poděkování

Rád bych poděkoval Donaldu E. Knuthovi, L. Lamportovi a všem programátorům, kteří věnovali svůj čas a schopnosti na vývoji systému $\text{T}_{\text{E}}\text{X}$ a jeho nadstaveb. Dále bych chtěl poděkovat svým rodičům za podporu a přátelům, kteří mi s touto prací pomohli.

Obsah

1	Úvod a cíl práce	6
1.1	Úvod do problematiky	6
1.2	Cíl práce	7
2	Přehled literatury	8
3	Analýza problému	9
3.1	Východiska řešení	9
3.2	Metodika řešení	11
4	Návrh řešení	12
4.1	Co je rejstřík	12
4.2	Co obsahuje rejstřík	13
4.3	Fungování systému L ^A T _E X	14
4.4	Zdrojový text	15
4.5	Práce s dokumentem	17
4.6	MakeIndex	18
4.7	Definice maker a prostředí	22
4.8	Jak na rejstřík	23
4.8.1	Inicializace rejstříku	23
4.8.2	Výběr slov	24
4.8.3	Generování rejstříku	26
4.8.4	Formát položek	28
4.9	Další úpravy rejstříku	32
5	Závěr	37
	Literatura	38
	Přílohy	40
	Rejstřík	48

Kapitola 1

Úvod a cíl práce

1.1 Úvod do problematiky

Člověk, který svými ambicemi a schopnostmi tíhne k psaní publikačního materiálu, by měl zvážit, jaké možnosti se mu nabízejí v dnešní době plně technického pokroku. Přizpůsobování se a hledání alternativ může být dobrou cestou umožňující snadnější zápis svého díla. V oblasti publikace vznikla celá řada firem, které se zabývají vývojem programového vybavení pro zpracování textu, nabízející velmi příjemné uživatelské prostředí oplývající celou řadou funkcí k formátování textu. Avšak většina z nich není pro uživatele lákavá, neboť se někdy pohybují ve vysokých cenových relacích a to nehledě na sazbu textu dle složitých typografických pravidel. Záleží samozřejmě na konkrétních požadavcích. Navíc velká část funkcí, které programy nabízejí, nemá své využití.

Existuje ale systém, který se považuje za dokonalý produkt pro tvorbu dokumentů. Je jím \LaTeX , vytvořený Leslie Lamportem z Digital Equipment Corporation. \LaTeX je jednou z nejvýznamnějších nadstaveb systému \TeX (podobně jako vyšší programovací jazyk nad strojovým kódem), vyvinutý za účelem snadnějšího a přirozenějšího zápisu sázeného textu. Oproti ostatním produktům z kategorie DTP - Desk Top Publishing (publikování "na stole") se \LaTeX odlišuje hned v několika směrech. Na první pohled se může zdát, že práce v něm je dosti složitá, neboť nemá implementované žádné grafické uživatelské prostředí pro současnou dobu velmi vyžadované. Přednosti systému spočívají především v jeho volné šiřitelnosti. Je tudíž zdarma dostupný na síti. Další výhody plynou z jeho stability a nezávislosti na zařízení (platforma, operační systém, výstupní zařízení).

Zakladatelem programu \TeX , jenž je jádrem celého systému a všech jeho nadstaveb, je Donald E. Knuth ze Standfordské univerzity. Tento systém

vytvořil proto, aby mohl své texty publikovat v požadovaném tvaru, neboť sazeči v tiskárně vnášeli obvykle do matematických vztahů mnoho chyb. Jeho koncepce je navržena tak dokonale, že od první verze vytvořené roku 1983 nedošlo k žádným zásadním změnám.

V České republice působí již od roku 1990 Československé sdružení uživatelů \TeX (\CTUG), jehož cílem je zejména podpora kvalitní počítačové typografie s využitím systému \TeX a jeho nadstaveb, a to formou udržování a distribuce \TeX ové instalace, bezplatnými poradenskými službami, spoluúčastí na vydávání a distribuci odborné literatury a vydáváním odborného časopisu *Zpravodaj \CTUG* .

Základní myšlenou vedoucí k vytvoření nadstavby \LaTeX je umožnit běžným uživatelům vysázet a následně vytisknout daný text na velmi profesionální typografické úrovni. Převaha uživatelů totiž ví, co chce sázet, nikoliv jak vysázet požadovaný text. Například příkaz `\chapter{Úvod}` způsobí v sazbě vytvoření nové kapitoly s názvem Úvod, ale jak bude tento nadpis vysázen, je před uživatelem skryto. Takže příslušné umístění nadpisu, volba písma, konstrukce obsahu a podobně je již implicitně nastaveno.

Velmi atraktivní vlastností systému je jeho variabilita, neboť umožňuje vytvářet své vlastní příkazy, které poté mohou být v dokumentu použity. Ale vychvalování systému a přesvědčování čtenáře o jedinečnosti a propracovanosti \LaTeX u není cílem této práce.

1.2 Cíl práce

Knižní rejstřík je jednou z nepostradatelných částí literatury odborného nebo populárně naučného charakteru. Jeho funkce je nanejvýš důležitá při vyhledávání určitého pojmu či tématu v rozsáhlých publikacích. Vytváření rejstříků je však někdy pro mnoho uživatelů obtížným úkolem. Rutinní vyhledávání slov v textu spolu s pracným utvářením konečné podoby rejstříku je snad už zahaleno minulostí. Schopnost, která může spisovateli usnadnit tento časově náročný proces, nabízí právě \LaTeX , s jehož pomocí a s pomocí externího programu *MakeIndex*, lze velice rychle dosáhnout kýženého výsledku.

Cílem této práce je tedy seznámit potencionální autory knih a uživatele systému \LaTeX s vytvářením těchto knižních rejstříků a obeznámit s možnostmi, kterými lze rejstřík upravovat a stylizovat. Dalším cílem, který jsem si předsevzal, je vytvoření tematického rejstříku, jenž je součástí většiny knih (např. kuchařek).

Kapitola 2

Přehled literatury

Mezi základní a nepostradatelné publikace určené zejména budoucím nebo začínajícím uživatelům systému \LaTeX patří bezesporu kniha *\LaTeX pro začátečníky* [1]. Kniha se nesnaží popsat zcela vyčerpávajícím způsobem všechna zákoutí systému, ale spíše seznamuje čtenáře se základními obecnými principy tvorby elektronických dokumentů. Text knihy je taktéž doplněn o seznam příkazů, které jsou nezbytné pro psaní vlastních dokumentů. Studium této knihy si čtenář osvojí pravidla spojená s psaním zdrojových textů a bude obeznámen s množstvím funkcí, které program nabízí. Obdobné souhrnné informace poskytuje také manuál *Ne příliš stručný úvod do systému \LaTeX 2 ϵ* [3], který přeložil do češtiny Michal Kočer.

Jako další literaturu zabývající se problematikou systému \TeX lze doporučit materiál nazvaný *Typografický systém \TeX* [2], který je možno použít jako navazující na předešlé dvě publikace. Samotná knížka není manuálem k \TeX u, ale chce být spíše průvodcem po mnohdy poměrně složitých technických souvislostech a zákonitostech \TeX u a doprovodného softwaru. Text tedy klade důraz zvláště na technické záležitosti.

Nesmím zde také opomenout velmi užitečné články napsané Michalem Švambergem [4, 5], které vysvětlují tvorbu rejstříku pomocí balíku `makerindex` a jsou pro tuto práci velkým přínosem.

Již od roku 1991 vychází odborný časopis vydávaný sdružením uživatelů \CS TUG, jehož členové velmi přispěli k osvětlení nejrůznějších problémů týkajících se systému \TeX . Jedním z významných autorů, který se snaží proniknout do tajů systému, je Zdeněk Wagner. V jednom ze svých článků se zabývá problematikou tvorby rejstříků pomocí programu *MakeIndex* [6]. Navazuje zde také na jedinečnost českého a slovenského jazyka, která při řazení slov v rejstříku hraje svoji roli. Musím zde upozornit na to, že zpravodaje, které v průběhu roků vycházely, nebyly před zadáním tématu této bakalářské práce pro veřejnost v tištěné a elektronické podobě k dispozici.

Kapitola 3

Analýza problému

3.1 Východiska řešení

V současné době je \LaTeX šířen týmem $\text{\LaTeX}3$, který je veden Frankem Mittelbachem. Tento tým se již nějakou dobu snaží sjednotit všechny rozšiřující verze \LaTeX u, které postupně vznikaly od vzniku \LaTeX u 2.09. Pro odlišení staré verze od nové se používá označení $\text{\LaTeX} 2_{\epsilon}$. Tato práce se zabývá verzí $\text{\LaTeX} 2_{\epsilon}$.

Jelikož je \LaTeX standardizován především pro anglicky psané dokumenty, byla pro účely použití tohoto systému v českých a slovenských podmínkách vyvinuta úprava \CSTeX šířena prostřednictvím sdružení uživatelů \CSTUG . Tato úprava obsahuje fonty, vzory pro dělení slov a další standardy nezbytné pro psaní českých a slovenských textů.

Vytváření rejstříků v \LaTeX u se provádí systematicky ve dvou krocích. Nejdříve je zapotřebí vybrat patřičná slova, a ta posléze seřadit podle abecedy. Pro účely řazení existuje několik programů. Nejvíce je však v souvislosti se sestavováním rejstříků označován program *MakeIndex*, který se vyznačuje svou jednoduchostí.

MakeIndex napsal původně Pehong Chen a později byl upraven řadou programátorů. Původní verze byla přirozeně anglická a později byla dodatečně implementována němčina. Od podzimu roku 1992 již existuje česká a slovenská verze tohoto programu pod názvem *CSTindex*, jehož autorem je Zdeněk Wagner, který je také členem sdružení \CSTUG . Pokud se někdo rozhodne získat program *CSTindex*, pak již nepotřebuje původní *MakeIndex*. *CSTindex* totiž obsahuje kompletní distribuci programu *MakeIndex* a navíc i podporu češtiny a slovenštiny. Vhodné je vyhledat takovou verzi *CSTindex*u, která podporuje více kódování. Lze ji například nalézt na ftp serveru Masarykovy university v Brně (ftp.fi.muni.cz).

CsIndex

Hlavním cílem kladeným na *CsIndex* bylo vytvořit funkční program s vynaložením minimálního úsilí. Proto není funkce programu zdaleka optimální. Program umí pouze řadit česká a slovenská slova podle požadavků normy ČSN 01 0181 z roku 1977. Tato norma byla však v roce 1994 zrušena a nahrazena normou ČSN 97 6030, která používá poněkud odlišnější a jednodušší pravidla. Například norma ČSN 010181 výslovně upřednostňovala malá písmena před velkými. Zatím jsem nikde nezískal informace o tom, že by někdo upravil program podle zásad této nové normy. Podmínkou pro korektní práci programu je kódování písmen v jednom ze tří kódů — KOI 8 ČS, Latin 2 nebo v kódu Kamenických.

Nedostatkem *CsIndexu* je, že nepřijme přehlasovaná písmena a podobná písmena s akcenty. Existuje však řešení. *CsTUG* distribuuje program *dvi2dvi*, který dokáže tato písmena převést tak, aby se správně vytiskla i se sedmi-bitovými fonty Computer Modern (základní písmo \TeX u). Program je tedy vyvinut pro češtinu a slovenštinu a rozumí tak písmenům obvyklým v těchto jazycích, ale nedokáže si poradit s francouzským \varnothing , dánským \emptyset a podobnými znaky.

Zvláštnosti češtiny a slovenštiny

K čemu vlastně potřebujeme speciální verzi programu *MakeIndex* pro češtinu a slovenštinu? Jednodušší by možná bylo, kdybychom vytvořili nějaký filtr a využili schopností programu *MakeIndex*. V článku Zdeňka Wagnera [6], který je také autorem modifikace *CsIndex*, je vysvětleno, v čem tkví odlišnosti českého a slovenského řazení. Německé třídění nečiní žádné zvláštní obtíže, neboť přehlasovaná písmena patří v abecedě na stejná místa jako příslušná písmena nepřehlasovaná. Ve slovníku tedy můžeme najít slova v pořadí:

Gefahr \prec Gefährdung \prec Gefahrgut \prec Gefährlichkeit.

Pro české a slovenské podmínky je situace poněkud složitější. Řada písmen se řídí obdobnými pravidly, které platí i pro německé přehlásky, ovšem háčky a čárky jsou v češtině (a zřejmě i slovenštině) významotvorné. Nemůžeme tedy říci *CsIndexu*, že **a** i **á** jsou identické. V takovém případě nedostaneme správné:

lak \prec lák \prec lakýrky.

Rozdíl mezi krátkým a dlouhým písmenem je totiž ignorován. Ohled se na něj bere v případě, jsou-li slova jinak identická. Primární řadící platnost mají písmena **č**, **ř**, **š** a **ž**.

Správné pořadí

cep < cesta < česnek < čepice

by se jinak změnilo na významově špatné

česnek < cesta < cep < čepice.

Problém zde také činí dvojhláska **ch**, která má taktéž primární řadící platnost a v abecedním pořadí patří mezi **h** a **i**. Jsou známá ovšem složená slova, kde **c** a **h** netvoří **ch**. Neexistuje však algoritmus, který rozezná rozdíl dvojhlásky nebo dvou samostatných písmen. Například slovo *mochnatý* je odvozeno od slova *mochna*, a *mochnatý*, znamenající "mající mnoho končetin". Přitom se obě slova opticky odlišují pouze jednou čárkou. Daleko složitější je to se zkratkami.

Někdo si může vymyslet, že Cvičná Horská CHata bude mít zkratku **CHCH**. V tomto případě všechny algoritmy úplně selhávají a zbývá jen lidský rozum. Dá se využít toho, že \TeX prázdné složené závorky `{}` interpretuje jako nic. Může se jich proto k oddělení **C** a **H** v případech, kdy netvoří dvojhlásku, použít. Příkaz pro **CHCH** tedy vytvoříme jako `\index{C{}HCH}` (*viz* Výběr slov).

3.2 Metodika řešení

Jak zde již bylo uvedeno, tak \LaTeX není závislý na platformě. Lze ho tedy nainstalovat na široké spektrum počítačů od IBM PC výše. Na většině univerzitních počítačových sítí, které většinou využívají platformu operačního systému UNIX, bývá \LaTeX již předinstalován. Podle druhu OS se však bude lišit i průběh a druh instalace. Psaní této práce a testování výsledků jsem prováděl na počítači s OS Linux (přesněji distribuce Ubuntu 8.10), jenž je mi velmi blízký vzhledem k jeho popularitě a technické podpoře.

K usnadnění instalace \TeX u s jeho nadstavbami a potřebnými programy byla vytvořena distribuce \TeX Live, která je pravidelně každoročně aktualizována a je všeobecně podporována a rozšiřována. Její instalací je tedy zajištěna nejnovější verze systému. Pro správné fungování systému je nutno nainstalovat balíky `texlive` a `texlive-lang-czechoslovak`. Po provedení tohoto kroku se kromě instalace \LaTeX u také připojí sada konfiguračních souborů $\mathcal{C}\mathcal{S}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ pro podporu českých a slovenských standardů. Dále doporučuji doinstalovat balík programů `texlive-extra-utils`, obsahující utilitu `detex` vhodnou k vyhledávání slov ve zdrojovém textu.

Program *MakeIndex*, který se užívá pro účely vytváření seznamů je standardně součástí distribuce \TeX Live, avšak jeho modifikace (tj. *CsIndex*) je potřeba samostatně doinstalovat.

Kapitola 4

Návrh řešení

4.1 Co je rejstřík

Rejstřík je abecedně uspořádaný ukazatel, sloužící k podrobné orientaci po obsahu publikace nebo jeho části. Jde o seznam položek, který je doplněn o místo, na kterém se položka vyskytuje. Typicky je tímto místem číslo stránky.

Existuje mnoho druhů rejstříků, které se liší podle charakteru publikace, zpracovaných dat a individuálních požadavků autora. Úspěšnost nalezení pojmu závisí na konkrétní formě rejstříku. K přehlednosti rejstříku může přispět vhodná struktura a koncepce či eliminace zbytečných pojmů. Pro názornou ukázkou zde uvádím pár příkladů rejstříků, se kterými se uživatel nejčastěji setká. Nejobecnější rozdělení rejstříků je na

- jmenný (autorský, místní, geografický, ...)
- věcný (seznam příkazů, tématický slovníček, ...)

Autorský rejstřík

Tento rejstřík je jedním z těch, který se považuje za rejstřík jmenný. Jméno autora je většinou vztahováno k nějakému dílu. V případě shodnosti příjmení určuje pořadí pojmu následující písmena dalších jmen. U některých jazyků se předpony jmen (De, Van, Von) považují za součást příjmení. Někdy jsou dále uváděna i jména organizací či institucí, ve kterých daná osoba působí.

Cauchy, Auguste Luis, 294
 Cervantes Savedra, Miguel de, 3
 Cicero, 232

Čapek, Karel, 8, 14

de Morgan, Augustus, 73, 296

Leibniz, G. W., 185, 293
 Lobačevskij, N.I., 212
 Locke, John, 7
 Lukaszewicz, Jan, 92, 94, 276

Marcus Aurelius, 1
 de Morgan, Augustus, 73, 296

Rejstřík klíčových slov

Zde jsou uvedena všechna klíčová slova, která se vyskytují v dané práci charakterizující předmětný obsah a zaměření práce. Výběr těchto slov je náhodný a je přímo v kompetenci autora. Jde o rejstřík věcný.

předmětová proměnná, 118
 předpoklad, 29, 35
 — implikace, 17
 — indukční, 199
 překlad, 238
 přirozený model, 213

struktura pro jazyk, 152
 subjekt-predikátový soud, 100
 subkontrárnost, 184
 substituce proměnné, 136
 subsumpce, 184
 suma množiny, 272

Tématický rejstřík

Jedná se o specifický druh rejstříku. Zahrnuje abecední seznam pojmů uspořádaných do tématických skupin. U každého pojmu může být uveden stručný popis. Typickým příkladem může být tématický slovníček či rejstřík v kuchařce, rozdělující pokrmy podle jejich druhu.

Abstrakt

`abstract` – prostředí produkující abstrakt

Aktuální datum

`\today` – příkaz generující aktuální datum podle nastavení data v operačním systému 39

Boxy

`\depth` – délkový registr nabývající hodnoty hloubky boxu (pod účařin je použitelný jen ve volitelném parametru příkazů `\makebox`, `\framebox`, `\savebox` a `\raisebox` 167

`\height` – délkový registr udávající výšku zpracovaného boxu, viz `\depth` 167

`\lrbbox{box}` – prostředí pro úschování

4.2 Co obsahuje rejstřík

Rejstřík začíná titulkem, který může souviset s případným speciálním zaměřením rejstříku (např. Rejstřík příkazů systému \LaTeX). Titulek je poté následován abecedně seřazeným seznamem hesel jedné, dvou či tří úrovní. Pojmy v podřízených úrovních jsou zúžením obecnějších pojmů nadřazených.

Pojmy se v rejstříku nesmějí opakovat. Podřízenost pojmu se může odlišit vložení pomlčky nebo tildy (\sim) před dané heslo či odsazením pojmu od levého okraje.

Kvůli přehlednosti se rejstřík člení na oddíly obsahující primární hesla se stejným počátečním písmenem spolu s hesly nižší úrovně. Oddíly mohou být uvozeny titulkem, jenž obsahuje příslušné písmeno, nebo odděleny jen svislým odsazením. Oddíl sdružující pojmy, kde primární hesla začínají neabecedním znakem (například číslicí), může být bez titulku nebo uvozen nadpisem "Symboly".

Vyskytují-li se heslo na více stranách, jsou potom tyto číselné prvky v rejstříku odděleny čárkou a mezerou. Těmito prvky mohou být jednotlivá čísla stran či rozsah čísel, na kterých má heslo svou platnost (např. 51-70). Lze je pro lepší přehlednost v rejstříku zarovnat k pravému okraji sloupce. Některá hesla mohou odkazovat na jiná hesla v rejstříku pomocí dodatku "viz". V tomto případě se čísla stránek k pojmu nevypisují.

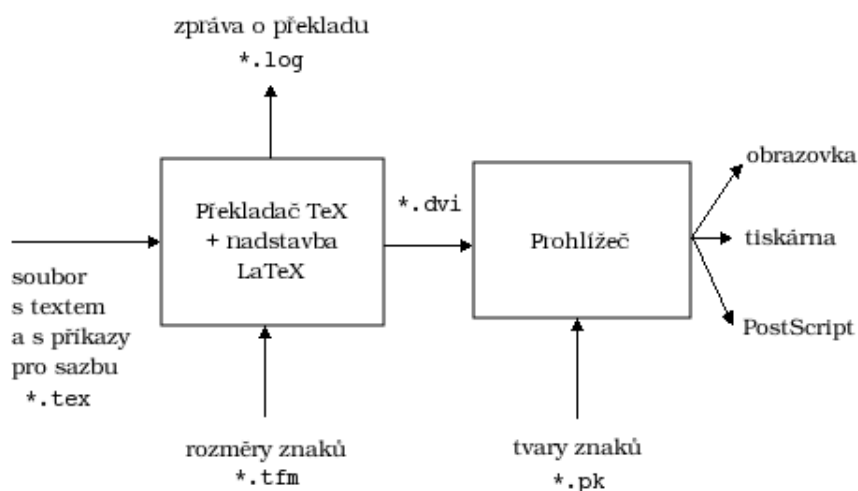
Rejstřík se vkládá až na konec knižního bloku, kde je snadno dostupný. Jeho začátek náleží vždy na pravou stránku. Titulek rejstříku by měl být naformátován stejně jako titulky kapitol. Podobně tak pro pojmy v rejstříku je vhodné použít stejné písmo jako pro základní text. Pro lepší orientaci a úsporu místa je možno členit rejstřík do dvou či více sloupců.

4.3 Fungování systému \LaTeX

Systém \TeX (společně s nadstavbou \LaTeX) tvoří jakýsi překladač, jehož vstupem je čistý textový **ASCII** soubor, vytvořený libovolným textovým editorem. Tento soubor obsahuje kromě vlastního textu také příkazy, podle kterých se \LaTeX řídí při sázení daného textu. Pro psaní dokumentu tedy není zapotřebí žádného složitého a drahého vybavení. Název vstupního souboru obvykle mívá rozšíření `.tex`.

Hlavním mechanismem programu je převedení zdrojového textu na jednotlivé znaky a jejich rozmístění do sazebního zrcadla podle jednotlivých příkazů, které jsou součástí tohoto textu. Následně je nutné také znát rozměry znaků, které jsou chápány jako obdélníky a bývají uloženy v souborech s příponou `.tfm` (\TeX Font Metric). O průběhu překladu vypisuje překladač na standardní výstup důležitá varování a chybová hlášení. Současně také vzniká textový soubor s příponou `.log` obsahující obdobné informace vypsané na standardní výstup.

Po úspěšném překladu je vytvořen výsledný soubor s vysázeným textem (rozšíření `.dvi`). Výhodou tohoto výstupního souboru je nezávislost na zobrazovacím zařízení (DeVice Independent), protože jeho koncepce je velmi obecná. Je možné ho proto poté zpracovat na různých finálních zařízeních. Neobsahuje tvary znaků, ale pouze jejich umístění. Proto je nezbytné použít nějaký program, který dané znaky přečte z disku a zobrazí výsledný vysázený text na obrazovce. Pro tyto účely lze použít program `xdvi`, který je součástí instalace \LaTeX u. Průběh překladu je znázorněn na obr. 4.1 převzatého z knihy Jiřího Rybičky [1].

Obrázek 4.1: Práce systému \LaTeX

4.4 Zdrojový text

Důležitou částí práce s \LaTeX em je znát správný zápis zdrojového textu spolu s příkazy ovlivňující sazbu. Pro psaní zdrojového textu lze použít například editory **gedit**, **pico** nebo **vi**, které jsou součástí distribuce Ubuntu. Zvolil jsem **gedit**, neboť má schopnost rozlišovat jednotlivé příkazy a tím zlepšuje orientaci v dokumentu.

Mezery

Všechny neviditelné znaky nebo několik po sobě následujících neviditelných znaků jako je mezera, tabulátor či konec řádku jsou L^AT_EXem chápány jako jedna mezera. Ignorovány jsou také neviditelné znaky na počátku řádku. Prázdný řádek mezi textovými řádky uvozuje nový odstavec. Více prázdných řádků je zpracováno jako jeden prázdný řádek.

Příkazy (makra)

Zápis příkazů může mít tyto varianty:

- jeden aktivní znak (např. `&`, `$`, `^`),
- jeden neabecední znak začínající zpětným lomítkem (např. `\@`, `\%`)
- posloupnost znaků začínající zpětným lomítkem (např. `\large`).

Systém rozlišuje mezi psaním malých a velkých písmen (tzv. case sensitive). Takže například příkaz `\huge` a `\Huge` mají jiný význam.

Některé příkazy mohou obsahovat parametry, které jsou buď povinné (psané ve složených závorkách) nebo nepovinné (psané v hranatých závorkách). Jednotlivé pořadí parametrů se musí v souladu s definicí příkazu držet.

Skupina, prostředí

Skupina je úsek textu, který je nějakým způsobem ohraničen. Ohraničení lze provést dvěma způsoby:

- složenými závorkami (např. `{\bf nějaký text}`)
- dvojicí příkazů `\begin{název}` a `\end{název}`, kde parametr `název` definuje název skupiny. Takto ohraničená *skupina* se nazývá *prostředí*. Prostředí může být implicitně nadefinované, nebo je možné vytvořit vlastní. Největším prostředím je prostředí dokumentu, ohraničený příkazy `\begin{dokument}` a `\end{dokument}`.

Některé příkazy ovlivňují sazbu textu až do konce prostředí (skupiny), v němž byly uvedeny. Hranicemi skupiny tedy nastavujeme úsek vlivu příkazů tohoto typu. Skupiny se mohou do sebe libovolně vnořovat, nesmějí se však křížit.

Komentáře

Pro účely vkládání komentářů, které nemají být součástí výstupu, slouží znak `%`. Vše, co stojí za tímto znakem, je až do konce řádku automaticky ignorováno.

Struktura dokumentu

Dokument v \LaTeX u se dělí na dvě části. První částí dokumentu je hlavička. Ta obsahuje základní informace o nastavení celého dokumentu. Hlavním a také prvním příkazem v hlavičce je nastavení třídy (stylu). Do verze 2.09 se používal příkaz `\documentstyle`, ale v této práci, která je psána ve verzi $\text{\LaTeX} 2_{\epsilon}$, je již použit příkaz `\documentclass`. Hlavičkový příkaz dokumentu z verze 2.09 umožňuje $\text{\LaTeX} 2_{\epsilon}$ rovněž .

Dále většinou následují příkazy pro zavedení balíčků maker (soubory s příponou `.sty`), které rozšiřují \LaTeX o nové příkazy, které pak mohou být v celém dokumentu použity. Připojení se provede libovolným opakováním příkazu `\usepackage`. V definici hlavičky musí být také zahrnuto kódování dokumentu. Ubuntu používá jako výchozí kódování **UTF-8**, avšak pro multiplatformní použití se spíše doporučuje kódování **cp1250**. Také je nutné doplnit hlavičku o zavedení českého standardu připojením balíku `czech` (české kódování vstupního textu, české názvy textových objektů – Kapitola, Rejstřík a podobně).

Druhou částí je pak tělo, kam se zapisuje vlastní text včetně příkazů k formátování textu. Tělo dokumentu je uvozeno dvěma příkazy `\begin{document}` a `\end{document}`. Výsledná struktura může vypadat následovně:

```
\documentclass[a4paper,12pt]{article}
\usepackage[cp1250]{inputenc}
\usepackage{czech}
\begin{document}

:

\end{document}
```

4.5 Práce s dokumentem

Kompilace

Nezbytným úkolem, při němž se přesvědčíme o správnosti zapsaného zdrojového textu a získáme tak výsledný výstupní soubor, je překlad dokumentu.

Ten může být proveden dvěma hlavními způsoby podle námi požadovaného výstupu. První možností je překlad do formátu DVI. Provádí se příkazem do příkazové řádky:

```
cslatex soubor.tex
```

Dalším velmi užitečným výstupním formátem vhodným k publikování v rámci sítě Internet je PDF. Získáme ho například převodem z formátu DVI pomocí programu `dvipdfm` příkazem:

```
dvipdfm soubor.dvi
```

nebo přímo překladem zdrojového textu programem `pdfcslatex`

```
pdfcslatex soubor.tex
```

Oba potřebné programy jsou součástí instalace \TeX .

Zobrazení

Nejočekávanějším okamžikem při tvorbě dokumentu je samozřejmě vidět vlastní text na zobrazovacím zařízení. K otevření souboru ve formátu DVI slouží již zmiňovaný program `xdvi`. Docílíme toho tímto zápisem do příkazové řádky:

```
xdvi soubor.dvi
```

Soubor konvertován do formátu PDF lze otevřít k tomu určeným programem Acrobat Reader firmy Adobe nebo v distribuci Ubuntu prohlížečem dokumentů Evince.

4.6 MakeIndex

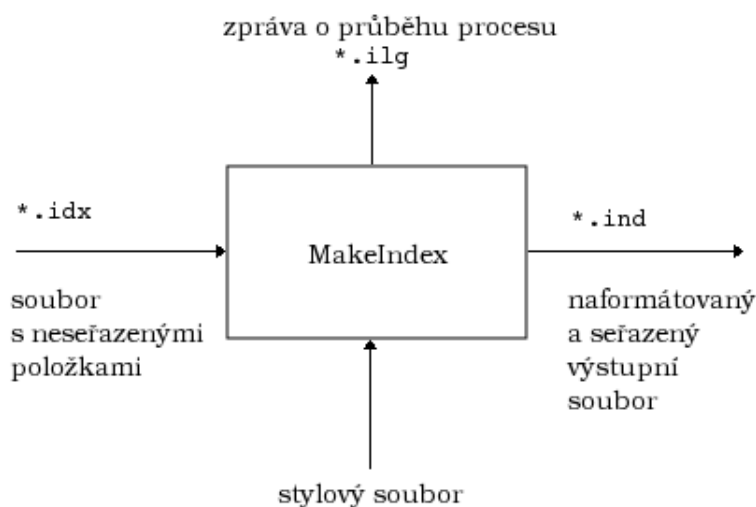
Popis

Program *MakeIndex* je obecně zaměřen na generování hierarchických rejstříků. Tento program byl původně vytvořen pro použití v \LaTeX u. \LaTeX má již nadefinovaná makra pro spolupráci s *MakeIndexem*. Součástí distribuce *MakeIndexu* je podrobná dokumentace.

Vstupem může být jeden nebo více souborů obsahující seznam hesel, které mají být ve výsledném rejstříku zahrnuty. Program tyto položky ze vstupního souboru seřadí podle abecedy a pomocí předdefinovaného stylu vytvoří výstupní soubor.

Dále také vzniká soubor s rozšířením `.ilg` (obdoba \LaTeX ovského log souboru), který uchovává data o běhu programu.

Formát vstupního a výstupního souboru může být specifikován případným stylovým souborem. Standardní formát vstupu a výstupu je převzat z \LaTeX u, pro který byl také program primárně navržen. Program povoluje maximálně tři úrovně hierarchického uspořádání rejstříku. Schéma procesu je znázorněno na obr. 4.2.



Obrázek 4.2: Práce programu *MakeIndex*

Přepínače

Program lze spouštět s řadou přepínačů, s jejichž pomocí se dá řídit a nastavit průběh procesu. Uvedeny jsou zde pouze ty hlavní. O dalších se můžete dočíst v dokumentaci k programu. Většina z nich však nejsou pro běžného uživatele potřebná.

- c** Ignoruje mezery a prázdná místa, která se nacházejí v úvodní nebo končné části textu, jež má být v rejstříku použita (např. " slovo " → "slovo").
- o ind** Nastaví výstupní soubor (argument `ind`). Standardně je jméno souboru tvořeno připojením rozšíření `.ind`.

- l Ignoruje účastnění mezer na abecedním řazení. Standardně je mezera chápána jako znak, který je na prvním místě v abecedě.
- q Tichý mód; nejsou zobrazeny zprávy na standardní chybový výstup (*stderr*). Standardně jsou informace o běhu programu a chybová hlášení poslána na tento výstup.
- r Zakáže implicitní formování stránkového rozsahu; stránkový rozsah musí být vytvořen použitím explicitních rozsahových operatorů. Standardně jsou tři a více po sobě jdoucí stránky automaticky zkráceny jako rozsah (např. 1–5).
- s *sty* Použije stylový soubor, kterým je možno specifikovat formát vstupního a výstupního souboru. Argument *sty* udává cestu, kde se případný stylový soubor nachází.
- t *log* Nastaví soubor, který poskytuje informace o běhu programu (argument *log*). Standardně je jméno tohoto souboru tvořeno připojením rozšíření *.ilg*.
- z *code* Nastaví vstupní kódování pro řazení pojmů v rejstříku. Přepínač lze použít pouze s programem *CGIndex*. Argument *code* udává jedno ze tří možných kódování (*koi8cs*, *i12* nebo *cp1250*).

Stylový soubor

Stylový soubor informuje program *MakeIndex* o formátu vstupních souborů a následném formátu výstupního finálního souboru. Tento soubor se nachází na cestě určené přepínačem *-s*. Stylový soubor obsahuje seznam dvojic *<specifikátor, atribut>*.

Existují dva typy specifikátorů: vstupní a výstupní. Dvojice se nemusí zadávat v přesném pořadí. Řádek začínající znakem *%* označuje komentář. V následujícím seznamu specifikátorů a atributů je *string* libovolný řetězec ohraničený dvojitými uvozovkami (*"..."*), *char* je znak ohraničený jednoduchými uvozovkami (*'...'*) a *number* je nezáporné číslo. Maximální délka řetězce je 2048 znaků. Vedle každého specifikátoru je uvedena standardní hodnota. Chceme-li zapsat znak zpětného lomítka či uvozovek, je nutné použít escape sekvence před tyto znaky (pomocí zpětného lomítka). Není-li uvedena dvojice *<specifikátor, atribut>*, program přiřadí standardní hodnoty.

O specifikátorech vstupního stylového souboru se zde zmiňovat nebudu, neboť jejich hodnoty není potřeba pro účely *L^AT_EX*u měnit. Uvedu zde pouze jen ty nejzákladnější. Pro pokročilejší uživatele doporučuji prostudovat dokumentaci k programu *MakeIndex*.

Specifikátory výstupního stylového souboru

`preamble` *<string>* "\\begin{theindex}\n"

řetězec, který je vložen na začátek souboru. Standardně je jím začátek prostředí `theindex`.

`postamble` *<string>* "\n\n\\end{theindex}\n"

řetězec, který je vložen na konec souboru.

`headings_flag` *<number>* 0

indikátor nastavující způsob výpisu záhlaví nové skupiny (titulek). Je-li tato hodnota záporná, bude se záhlaví vypisovat malými písmeny, a naopak kladná hodnota vyprodukuje záhlaví tvořené písmeny velkými. Standardní hodnota je 0, která netiskne žádné záhlaví.

`heading_prefix` *<string>* ""

řetězec, který je vložen před titulek. Může tak nastavovat typ a řez písma.

`heading_suffix` *<string>* ""

řetězec, který je vložen za titulek.

`group_skip` *<string>* "\n\n \\indexspace\n"

řetězec, který je vložen před začátek nové skupiny. Standardně je jím vertikální mezera.

`item_0` *<string>* "\n \\item "

řetězec, který je vložen mezi dvě primární (úroveň 0) položky.

`item_1` *<string>* "\n \\subitem "

řetězec, který je vložen mezi dvě sekundární (úroveň 1) položky.

`item_2` *<string>* "\n \\subsubitem "

řetězec, který je vložen mezi dvě položky 2. úrovně.

`item_01` *<string>* "\n \\subitem "

řetězec, který je vložen mezi položky úrovně 0 a 1.

`item_x1` *<string>* "\n \\subitem "

řetězec, který je vložen mezi položky úrovně 0 a 1, kde položce v úrovni 0 není přiřazeno číslo stránky.

`item_12` *<string>* "\n \\subsubitem "

řetězec, který je vložen mezi položky úrovně 1 a 2.

`item_x2` *<string>* "\n \\subitem "

řetězec, který je vložen mezi položky úrovně 1 a 2, kde položce v úrovni 1 není přiřazeno číslo stránky.

`delim_0, delim_1, delim_2` *<string>* ", "

řetězec, který je vložen mezi název položky úrovně 0, 1 nebo 2 a první číslo stránky, na které se pojem nachází.

4.7 Definice maker a prostředí

Definice příkazu(makra)

V některých případech nám k zjednodušení práce pomůže nadefinovat nový příkaz. Definování nového makra se provede pomocí příkazu:

```
\newcommand{jméno}[číslo]{definice}
```

kde parametr `jméno` udává přirozeně jméno nově vytvořeného příkazu, nepovinný parametr `číslo` se užívá v případě, když chceme, aby tento příkaz sám používal argumenty (parametry). Tento parametr může nabývat hodnot 1–9. Posledním parametrem `definice`, nastavujeme vlastní funkci příkazu. Na následujícím příkladu je ukázáno, jak napsat velmi jednoduché makro. Nazveme jej například `\textmacro`. Jeho funkcí bude napsání neproměnného textu.

```
\newcommand{\textmacro}{Zde máme nějaký text}
```

Makro můžeme kdykoliv předefinovat. K tomu slouží příkaz `\renewcommand`:

```
\renewcommand{\textmacro}{Zde máme pozměněný text}
```

Makro bez argumentů je ovšem velmi málo flexibilní. Pro tyto potřeby tedy využijeme parametru `číslo`. V makru se odvoláváme na parametr znakem `#`, za nímž následuje pořadové číslo argumentu. Makro se dvěma parametry se definuje takto:

```
\newcommand{\textmacro}[2]{1. parametr: #1, 2. parametr: #2}
```

Kromě \LaTeX ového příkazu `\newcommand` lze také užít i ekvivalentní \TeX ový příkaz `\def`. Zápis je poněkud jednodušší, ale mocnost větší. Forma zápisu pak má tvar:

```
\def\textmacro#1#2{1. parametr: #1, 2. parametr: #2}
```

Makro `\makro` pak budeme volat takto:

```
\textmacro{první slovo}{druhé slovo}
```

Definice prostředí

Pro další úpravy rejstříku bude nutno umět nadefinovat nová prostředí. To se provádí podobně jako u definice maker. Příkaz `\newenvironment` slouží k vytvoření nového prostředí a `\renewenvironment` k předefinování již definovaného prostředí. Struktura příkazu pak vypadá následovně:

```
\newenvironment{nazev}{%
:
}%
{ ... }
```

kde *název* přirozeně udává název prostředí. Do prvního páru složených závoerek se vkládají příkazy, které se provedou jako rozvinutí příkazu `\begin{nazev}` a do druhého páru se zapisují příkazy, které se mají provést při rozvinutí `\end{nazev}`.

Některé řádky jsou v tomto příkladu ukončeny znakem `%`, což se doporučuje z důvodu korektního provedení definice prostředí. Bez tohoto znaku by \LaTeX mohl vyvolat chybu, neboť by byl zmaten mezerami. Neví-li někdo, kde je znak procenta nutný, může pro jistotu ukončovat procentem všechny řádky.

4.8 Jak na rejstřík

4.8.1 Inicializace rejstříku

Prvním krokem při tvorbě rejstříku je náležitá inicializace, která se provede připojením balíku maker obsahující rozšiřující příkazy. Základním balíkem, který slouží pro tyto účely, je balík `makeidx`. Ten je bohužel omezen pouze na implementaci jednoho rejstříku a jednoho glosáře. Budeme-li mít v úmyslu vytvořit více na sobě nezávislých rejstříků, je nutno připojit balík `index`, který se také dá snadno použít pro jiné seznamy než jen rejstříky. V této práci je využito druhé možnosti. Připojení balíku provedeme tímto zápisem do hlavičky dokumentu:

```
\usepackage{index}
```

Po zavedení balíků jsou tedy k dispozici potřebné příkazy pro vytváření rejstříků. Samotná inicializace se však provede až příkazem:

```
\makeindex
```

nebo

```
\newindex{typ}{přípona_dx}{přípona_nd}{nadpis}
```

Tyto příkazy se zapisují opět do hlavičky dokumentu. Příkaz `\newindex` obsahuje čtyři povinné parametry. Hodnota `typ` je libovolné označení rejstříku. Ta může být nastavena na `default`, chceme-li k rejstříku přistupovat častěji. V případě, že se bude vytvářet více rejstříků (např. rejstřík zkratk nebo autorský rejstřík), je nutno je od sebe odlišit unikátním typem. Parametr `přípona_dx` určuje příponu souboru, kam se budou ukládat veškeré záznamy z dokumentu označené příkazem `\index` a vzniká při kompilaci dokumentu. Parametr `přípona_nd` udává příponu výsledného souboru obsahující již hotově zpracovaný rejstřík vygenerovaný programem *MakeIndex*. Název tohoto souboru se musí shodovat s názvem dokumentu, jinak nedojde k jeho načtení. Posledním parametrem `nadpis` definujeme nadpis rejstříku. Druhým příkazem `\makeindex` se dosáhne standardního nastavení rejstříku ekvivalentní zápisu `\newindex{default}{idx}{ind}{Rejstřík}`. Hlavička dokumentu může tedy vypadat následovně:

```
\usepackage{index}
\makeindex
\newindex{rm}{mdx}{mnd}{Abecední seznam příkazů}
\newindex{tr}{tdx}{tnd}{Tematický rejstřík}
```

Kompilací dokumentu tak vzniknou tři soubory s příponami `idx`, `mdx` a `tdx`. Přípony souborů se nemusí udávat právě tímto způsobem, ale je to běžné a doporučené používání.

4.8.2 Výběr slov

Pro výběr slov z textu, která se mají v rejstříku objevit, se v \LaTeX používá příkaz `\index`. Ten je již standardně nedefinován v prostředí \LaTeX (tj. bez zavedení potřebného balíku `maker`), ale jeho funkčnost je dá se říct nulová. Proto i bez zavedení balíku `index` bude překlad dokumentu s tímto příkazem úspěšný. Schopnosti příkazu nabudou hodnot právě až po zavedení tohoto balíku, kde dochází k předefinování makra, a při dalším použití má již příkaz námi požadované vlastnosti. Definice příkazu je:

```
\index[typ]{pojem}
```

kde parametr `typ` je hodnota udávající příslušnost k danému rejstříku označený unikátním typem. V situaci, kdy chceme přidávat záznamy do rejstříku, který je definován jako `default` nebo pokud je použito makro `\makeindex`,

nemusí se hodnota `typ` udávat. Povinným parametrem `pojem` je pak text, který má být v rejstříku zahrnut. Tento parametr může také obsahovat znaky pro formátování rejstříku (*viz* Formát položek). Výsledný zápis může vypadat například takto:

```
Slunce \index{Slunce} je zdrojem světelné a tepelné energie.
```

Příkazem `\index` se tedy označí dané slovo, které se bude dále zpracovávat. Jak si můžeme všimnout, tak slovo `slunce` je zde uvedeno dvakrát (jako slovo v textu a jako parametr). Příkaz tedy pouze vytvoří jakousi relaci na danou stránku, kde se slovo vyskytuje. Proto existuje také varianta s hvězdičkou tohoto příkazu (tj. `\index*`). Tímto zápisem se slovo, které chceme zahrnout do rejstříku, nemusí udávat dvakrát.

```
\index*{Slunce} je zdrojem světelné a tepelné energie.
```

Můžeme namítnout, proč existují dva ekvivalentní zápisy, které vedou ke stejnému výsledku? Využití může být různé. Například výraz obsažený v rejstříku nemusí striktně odkazovat pouze na slovo v textu, ale může se také vztahovat k tabulce nebo obrázku.

Výběr slov je především autorská záležitost. Kandidáty slov lze vybrat různými způsoby. Nejvíce časově náročnou metodou je rutinní označování jednotlivých výrazů. Autor dokumentu však nebude každou chvíli zvažovat, zda má či nemá daný výraz do rejstříku zařadit. K tomu může posloužit nějaký program, který vypíše všechna slova, která jsou v textu použita. Jednou z možností jak toho dosáhnout je tímto zápisem do příkazové řádky:

```
cat soubor.tex | detex | tr ' ' '\n' | sort | uniq -c | sort -n -r
```

Ten nám vyprodukuje výpis všech hesel v textu se vyskytujícími a seřazených podle četnosti. Z tohoto seznamu se vyberou vhodná slova a pak metodou "najdi a nahrad" se přidá ke každému takovému slovu řídicí příkaz `\index`.

Při vytváření více rejstříků si můžeme práci usnadnit tím, že si nadefinujeme vlastní příkaz pro výběr položek, které chceme do rejstříku zahrnout. Zdá se mi nepraktické zapisovat například makro ve tvaru `\index[rm]{polozka}`, chceme-li danou položku zahrnout do rejstříku příkazů. Pro tyto účely si vytvoříme vlastní makro:

```
\def\mindex{\index[rm]}
```

Volání příkazu se pak bude provádět zápisem `\mindex{polozka}`.

4.8.3 Generování rejstříku

Po inicializaci rejstříku, výběru slov a řádném překladu dokumentu vznikne soubor, do kterého se uložily veškeré záznamy označené příkazem `\index`. Tento pracovní soubor má příponu určenou v parametru příkazu `\newindex` nebo pokud jsme použili příkaz `\makeindex`, má tento soubor implicitně nastavenou příponu `.idx`. Otevřít takto vzniklý soubor je možno opět v libovolném textovém editoru. Obsah souboru může vypadat následovně:

```
\indexentry{hmota}{6}
\indexentry{energie}{6}
\indexentry{hmota}{9}
\indexentry{realita}{12}
\indexentry{relativní}{12}
```

Záznamy uložené v tomto pracovním souboru jsou zapsány ve formátu

```
\indexentry{položka}{stránka}
```

Na první pohled je vidět, že záznamy nejsou seřazeny podle abecedy. Ukládají se postupným překladem dokumentu. Následné setřídění a zpracování rejstříku do konečné podoby se provádí již zmíněným programem *MakeIndex*. Program se spouští následujícím zápisem do terminálu:

```
makeindex soubor
```

Pokud využijeme program *CSIndex* pro řazení českých a slovenských slov, bude zápis obdobný. Musí se pouze zadat patřičné kódování prepínačem `-z`.

```
csindex soubor -z il2
```

Tento příkaz zajistí, že program ze vstupního souboru `soubor.idx` vytvoří výstupní soubor `soubor.ind`. Není-li v zápisu u názvu souboru uvedena přípona, program má standardně nastaveno načtení souboru s rozšířením `.idx` a následně také uložení zpracovaných dat do souboru s rozšířením `.ind`.

Bude-li potřeba vytvořit soubor s jiným názvem (v případě více rejstříků), musí se použít prepínač `-o`. V tom případě je potřeba dávat pozor, aby se přípona výstupního souboru shodovala s definicí příslušného příkazu `\newindex`. Do terminálu se tedy bude například psát:

```
makeindex soubor.mdx -o soubor.mnd
```

Otevře-li se takto vzniklý výstupní soubor v textovém editoru, jeho obsah může vypadat následovně:

```
\begin{theindex}

  \item energie, 6

  \indexspace

  \item hmota, 6, 9

  \indexspace

  \item realita, 12
  \item relativní, 12

\end{theindex}
```

Záznamy v tomto souboru jsou uvozené příkazy `\begin{theindex}` a `\end{theindex}`, což je z důvodu možné implementace do prostředí L^AT_EXu. Na příkladu je vidět, že program *MakeIndex* ignoruje vícenásobný výskyt stejných slov. Místo toho napíše toto slovo na řádek a vedle něj čísla stránek, na kterých se výraz vyskytuje. Je-li nalezeno více slov na jedné stránce, její číslo je vypsáno pouze jednou.

Pokud se slovo bude například vyskytovat na stránkách 1, 2 a 3, tak se může zajistit, aby program napsal 1-3 místo 1,2,3. Dále je tu možnost vytvářet víceúrovňový rejstřík, případně používat různé efekty pro formátování položek (*viz* Formát položek). Mezi skupinami slov začínajících stejným písmenem vloží program příkaz `\indexspace`, čímž vznikne vertikální mezera pro oddělení jednotlivých skupin. Nakonec příkazem

```
\printindex[typ] [prolog]
```

na příslušné místo v dokumentu rejstřík implementujeme. Parametr `typ` opět udává unikátní název rejstříku nadefinovaný v příkazu `\newindex` a parametrem `prolog` můžeme do úvodní části rejstříku vložit text, jenž nám bude popisovat daný rejstřík (*viz* Prolog). Výsledný zápis může vypadat takto:

```
\printindex
\printindex[rm] [\prolog]
\printindex[tr]
```

Kompilací dokumentu a jeho zobrazením je již možno výsledný rejstřík na daném místě spatřit.

4.8.4 Formát položek

Aby rejstřík přinesl čtenáři co nejvíce užitku, je možné ho upravovat různými způsoby. Vzhled, řazení a strukturování položek v rejstříku lze ovlivnit informacemi z argumentu příkazu `\index`, které se posléze zpracují programem *MakeIndex* (popřípadě *CSIndex*). V tabulce 4.1 je základní přehled znaků upravující položky rejstříku.

Znak	Popis
!	oddělovač úrovní (max. tři úrovně)
	oddělovač jména a formátu čísla stránky
(stránkový rozsah - začátek
)	stránkový rozsah - konec
@	oddělovač jména položky a zobrazeného textu v rejstříku
"	slouží k zápisu výše uvedených znaků

Tabulka 4.1: Znaky pro formátování rejstříku

Toto jsou základní parametry, které se nejčastěji používají pro úpravu rejstříku. O dalších se můžete dočíst v dokumentaci k programu *MakeIndex*.

Abecední uspořádání rejstříku není závislé na tom, zda-li je položka zapísána velkými či malými písmeny. Je také respektován způsob číslování stránek, na kterých se příkaz `index` nacházel. Mezeru považuje program za obyčejný znak, který se účastní abecedního řazení a je chápána jako znak, který je první v abecedě. Způsob řazení se dá případně ovlivnit pomocí prepínačů, se kterými se program spouští (*viz* Prepínače).

Strukturování

Strukturování rejstříku se provádí pomocí znaku `!`, který slouží jako oddělovač úrovní. Mějme příkazy:

```
\index{hmotná příroda!tři kvality} (na straně 32)
```

```
\index{hmotná příroda!tři kvality!dobro} (na straně 35)
```

Pokud definice nižší úrovně chybí (v příkladu položka 'hmotná příroda'), je automaticky vytvořena, ale bez příslušné stránky, na které se nachází.

Po překladu dokumentu a následné úpravě pomocí programu *MakeIndex* vznikne soubor s obsahem

```
\begin{theindex}

  \item hmotná příroda
    \subitem tři kvality, 32
      \subsubitem dobro, 35

\end{theindex}
```

kde příkazy `\item`, `\subitem` a `\subsubitem` definují jednotlivé úrovně. Jak už je zde uvedeno, tak program *MakeIndex* povoluje maximálně tři úrovně. To je jak z hlediska náročnosti na řazení položek tak z hlediska přehlednosti dostačující. Při vytváření více úrovní, program vypíše chybu do výstupního informačního souboru s příponou `.ilg`.

Stránkový rozsah

Chceme-li, aby pojem měl platnost v určitém stránkovém rozsahu, použijeme dvojici znaků (`a`), které ohraničují začátek a konec daného rozsahu. Například příkazy

```
\index{vědomí!vyšší|()} (na straně 124)
\index{vědomí!vyšší|)} (na straně 126)
```

vznikne rejstříkový záznam v podobě

```
\begin{theindex}

  \item vědomí
    \subitem vyšší, 124--126

\end{theindex}
```

Před znaky definující stránkový rozsah je nutno zapsat znak `|`, který zde odděluje jméno položky od jejího formátování. Pokud je platnost pojmu pouze na jedné stránce, vypíše se jen číslo této stránky. Program také dokáže ignorovat výskyt stejných hesel nacházejících se v daném stránkovém rozsahu.

Číslo stránky

Další možností úpravy položky v rejstříku je formátování čísla stránky. To se provádí zápisem

```
\index{...|cmd}
```

kde `cmd` je příkaz nastavující typ či řez písma a píše se bez zpětného lomítka. Toto lomítko přidá automaticky program *MakeIndex* do výstupního souboru. Číslo stránky pak bude vypsáno ve tvaru `\cmd{číslo stránky}`. Toto je vhodné, pokud chceme rozlišit, zda-li je například na daném místě pojem vysvětlen nebo se případně vztahuje k obrázku či tabulce. Například

```
\index{pojem} (na straně 60)
\index{pojem!vysvětlení|textit} (na straně 64)
\index{pojem!tabulka|textbf} (na straně 69)
```

vytvoří položku

```
\item pojem, 60
  \subitem vysvětlení, \textit{64}
  \subitem tabulka, \textit{69}
```

Vhodnější je však vytvořit si vlastní příkazy, kterými pak můžeme ovlivňovat všechny položky v případě, že bychom je potřebovali změnit. Vytvoříme si tedy nové příkazy. Příklad je převzat z článku M. Švaberga [4]:

```
\newcommand{\pop}[1]{\textit{#1}}
% příkaz \pop{} pro popis pojmu
\newcommand{\obr}[1]{\textbf{#1}}
% příkaz \obr{} pro pojem v obrázku
\newcommand{\tab}[1]{\textbf{#1}}
% příkaz \tab{} pro pojem v tabulce
\newcommand{\poz}[1]{#1n}
% příkaz \poz{} pro pojem v poznámce pod čarou
```

Vytváření nových příkazů je již vysvětleno výše v této práci a tak se jeho popisem nebudu zabývat. Volání těchto příkazů se rozvine následovně:

```
\pop{černá díra} => \textit{černá díra}
\obr{52}          => \textbf{52}
\tab{natalita}   => \textbf{natalita}
\poz{6}          => 6n
```

V makru `\index` se pak příkazy použijí takto:

```
\index{pojem} % normální odkaz na pojem
\index{pojem|obr} % číslo stránky bude tučné
\index{pojem|pop|} % začátek rozsáhlého popisu pojmu

\index{pojem|pop|)} % konec rozsáhlého popisu pojmu
\index{pojem|poz} % pojem v poznámce, k číslu stránky přidá 'n'
```

Když bude potřeba změnit typ nebo řez písma u jednoho z těchto příkazů, tak stačí přepsat definici tohoto příkazu a dokument opět zkompileovat.

Odkaz na jiný pojem

Prostředí L^AT_EX má také schopnost odkazovat se na jiný pojem v rejstříku. K tomu je určen speciální příkaz `\see`. Použijeme-li tento příkaz pro formátování položky rejstříku, místo čísla stránky se vypíše slovo "viz" a za ním odkazované heslo. Stejně jako v předchozím případě se to tedy provádí zápisem ve tvaru

```
\index{...|see{heslo}}
```

Takže například příkazem

```
\index{transmigrace|see{převtělování}} (na straně 121)
```

vznikne položka v rejstříku

```
\item transmigrace, \see{převtělování}{121}
```

Odlíšný pojem

Někdy se stane, že potřebujeme do rejstříku zařadit pojem, jehož název se liší od pojmu, na který odkazuje, ale významově spolu souvisí. K tomu slouží oddělovač `@`, s jehož pomocí můžeme formátovat text a písmo pojmu nebo vkládat různé symboly (např. písmena řecké abecedy). Mějme například příkazy

```
\index{padesát} (na straně 35)
```

```
\index{padesát dva@\textbf{padesát dva}} (na straně 36)
```

```
\index{padesát@L} (na straně 38)
```

ty vytvoří v rejstříku tento zápis

```
\begin{theindex}
```

```
\item padesát, 35
```

```
\item L, 38
```

```
\item \textbf {padesát dva}, 36
```

```
\end{theindex}
```

Můžeme si všimnout, že řazení slov se neprovádí podle pojmů uvedených za znakem `@`, nýbrž podle pojmů, které jsou uvedeny před tímto znakem (v příkladu položka 'xx'). Pojem 'padesát dva' bude vysázen tučným písmem.

Escape

Budeme-li chtít do položky v rejstříku vložit speciální znaky zde uvedené (tedy !, | a @), provedeme to zápisem escape sekvence " před tyto znaky. Je-li nutno zapsat znak ", zapíšeme jej zdvojeně.

Všechny speciální znaky upravujících položky rejstříku se dají kombinovat. Takže není nějaké omezení pro použití pouze jednoho znaku.

4.9 Další úpravy rejstříku

Standardní vzhled rejstříku se může zdát většině uživatelů velmi strohý. Autor může mít proto touhu obohatit rejstřík o nějaké prvky, jež by zpřehlednily rejstřík a usnadnili tak nalezení patřičného hesla.

Prolog

Prolog je dobrým prostředkem k popisu rejstříku. Budeme-li využívat formátování čísla stránky k odlišení odkazů na objekty v obsahu publikace, můžeme tím čtenáři velmi pomoci při hledání požadovaných dat. Tato úprava však nemá žádnou účinnost, pokud k ní nedodáme potřebnou předmluvu. Ta by měla čtenáři objasnit, jakým způsobem autor stylizoval položky v rejstříku.

Pro tyto účely slouží nepovinný parametr `prolog` příkazu `\printindex`, o kterém je v této práci zmíněno již výše. Nadefinujeme si například příkaz `\prolog`, který bude obsahovat úvodní text, ten poté vložíme do parametru makra `\printindex`. Pozor však na pořadí parametrů, které je nutno dodržet. Vytváříme-li rejstřík typu `default`, musíme tuto definici do parametru makra zahrnout. To provedeme zápisem

```
\def\prolog{Čísla vytištěná tučně označují stránky, kde je
příslušné makro popsáno. Všechna ostatní ukazují na místo,
kde je makro použito. Položka, která nezačíná znakem
zpětného lomítka označuje balík maker.
}
\printindex[rm][\prolog]
```

Nyní stačí pouze dokument zkompileovat. Výsledek lze zhlédnout na obr. 4.3.

Titulek

Titulkem je zde míněn nadpis skupiny v podobě znaku, který se nachází v záhlaví každého nově započatého písmena v abecedním řazení slov a umož-

Abecední seznam příkazů

Čísla vytisknutá tučně označují stránky, kde je příslušné makro popsáno. Všechna ostatní ukazují na místo, kde je makro použito. Položka, která nezačíná znakem zpětného lomítka označuje balík maker.

	B				
<code>\begin</code>	16	<code>\newindex</code>	23
<code>\bf</code>	32	<code>\nopagebreak</code>	32
			P		
	D		<code>\printindex</code>	27, 32
<code>\def</code>	22			
<code>document</code>	17	R		
<code>\documentclass</code>	17	<code>\renewcommand</code>	22

Obrázek 4.3: Upravený rejstřík

ňuje tak snadnější orientaci v rejstříku. Tento titulek není při implicitním nastavení zobrazen, a tak ho musíme nastavit ručně. To provedeme pomocí stylového souboru programu *MakeIndex*, do kterého vložíme následující řádky.

```
heading_prefix "{\\bf\\hfill "
heading_suffix "\\hfill}\\nopagebreak\n"
headings_flag 1
```

Příkazy `\bf`, `\hfill` a `\nopagebreak` způsobí, že bude tento znak vytisknut tučně na střed sloupce se zákazem řádkového zlomu. Načtení stylového souboru, který nám takto naformátuje výsledný rejstřík se provede následujícím zápisem do terminálu.

```
makeindex soubor -s stylovýSoubor
```

Nyní stačí pouze zkompilovat dokument a zobrazit ho na monitor (obr. 4.3).

Úprava položek

O formátování položek rejstříku se již částečně tato práce zmiňuje. Je zde však pojednáno spíše o struktuře než o formě. Úpravy položek se uskutečňují opět zásahem do stylového souboru. Do tohoto souboru vložíme například

```
delim_0 "\\dotfill "
delim_1 "\\dotfill "
delim_2 "\\dotfill "
```

čímž upravíme formát výpisu jednotlivých položek. Příkaz `\dotfill` v tomto příkladě způsobí horizontální mezeru vyplněnou tečkami. Stejným způsobem lze použít například příkazy `\hfill` (prostá mezeru) nebo `\hrulefill` (souvislá čára). Nakonec jako v předchozí části zformátujeme výstupní soubor a následně přeložíme a zobrazíme dokument (obr. 4.3).

Tématický rejstřík

Již nějakou chvíli jsem si lámal hlavu nad tím, jak vytvořit v \LaTeX u tématický slovník či rejstřík podobný například tomu, který se nachází v knize Jiřího Rybičky [1] nebo ve většině kuchařek. Tématický rejstřík totiž obsahuje hesla, která jsou navíc rozdělena do jednotlivých tématických skupin. Názvy těchto tématických skupiny jsou ve většině případů také řazeny podle abecedy. Muselo by tedy dojít k dvojímu třídění.

Samozřejmě, že by takovýto úkol šel řešit nějakým dalekosáhlým makrem, který by měl v závěru plno nedostatků. Nakonec jsem zjistil, že jde tento problém řešit mnohem jednodušeji s minimálním zásahem do zdrojového textu a s pomocí programu *MakeIndex*.

Před tím, než se bude zasahovat do dokumentu, je nutné vymyslet seznam témat, podle kterých se budou jednotlivé pojmy třídit. Nejlépe bude si tento koncept zapsat na papír. Poté je vhodné si k těmto názvům přiřadit zkratky skládající se z prvních písmen názvů tak, aby byly unikátní pro řazení. Například

Dokument	=> dok
Písmo - typ, řez	=> pís
Prostředí	=> pro
Nová stránka	=> nov
Definice příkazů	=> def
Výplně	=> výp
Rejstřík	=> rej

Tato zkratka se pak použije v argumentu příkazu `\index` následovně

```
usepackage \trindex{dok usepackage@\verb \usepackage }
```

Tímto způsobem si označíme ta hesla, která chceme do rejstříku vložit a zkompilujeme dokument. Příkaz `\verb` je zde použit pro doslovné vysázení textu (vhodné například pro \LaTeX ové příkazy). Poté přichází na řadu program *MakeIndex*.

Takto vzniklý soubor se seřazenými hesly ručně doplníme o předem určená témata tak, že nad každou skupinu pojmů, které tvoří jeden tématický celek, vložíme název tématu a to například takto:

```
\indexspace
{\bf Dokument\hfill}\nopagebreak

\item \texttt {document}\dotfill 17
\item \verb \documentclass \dotfill 17
\item \verb \documentstyle \dotfill 17
\item \verb \usepackage \dotfill 17

\indexspace
{\bf Nová Stránka\hfill}\nopagebreak

\item \verb \nopagebreak \dotfill 32
```

K rejstříku může být případně přidán nějaký prolog zahrnující seznam tématických skupin. Nyní stačí dokument zkompilovat a zobrazit (obr. 4.4).

Více sloupců

U publikací tištěných na větší formáty (např. A4) je praktické z důvodu přehlednosti a ušetření místa členit rejstřík do více sloupců. Jelikož implicitní prostředí `theindex` má pevně nastaveny parametry rejstříku, je těžké je takto změnit. Proto je lepší si vytvořit vlastní prostředí pro tisk rejstříku například takto:

```
\newenvironment{theteminde}{\chapter*{Tématický rejstřík}
\begin{multicols}{3}[\prolog]
\IndexParms
}%
{\end{multicols}}
```

Zápis definuje prostředí `\theteminde`, kde titulek rejstříku bude mít název **Tématický rejstřík**. Poté bude následovat prolog, který se musí zvlášť nadefinovat. Dále už pokračují pojmy, které se budou členit do tří paralelních sloupců. To provádí prostředí `multicols` s jedním povinným parametrem udávajícím počet sloupců a jedním nepovinným, který vysází text uvnitř parametru nad tyto sloupce. V tomto případě tím bude makro `\prolog`. Tímto odpadá nepovinný parametr příkazu `\printindex`. Příkaz `\IndexParms` nastavuje jednotlivé parametry rejstříku. Musí se však zvlášť nadefinovat. Definice makra `\IndexParms` je převzata ze souboru `doc.sty` od Franka Mittelbacha.

```

\def\IndexParms{%
\columnsep 15pt
\parskip 0pt plus 1pt
\rightskip 15pt
\parfillskip=-15pt
\mathsurround 0pt
\columnsep 15pt
\parindent 0pt
\def\main##1{{\bf ##1}}
\def\item{\par\hangindent 30pt}%
\def\subitem{\item\hspace*{15pt}}%
\def\subsubitem{\item\hspace*{25pt}}%
}

```

V této fázi můžeme vstupní soubor obsahující neseřazené pojmy zpracovat programem *MakeIndex*. Ten nám standardně vyprodukuje prostředí *theindex*. Aby bylo možné uplatnit výše uvedené změny, je potřeba název tohoto prostředí změnit na *thetemindex*. To lze buď ručně tak, že přímo v tomto výstupním souboru přepíšeme název prostředí, nebo s využitím stylového souboru, do kterého vložíme tyto řádky:

```

preamble "\n \begin{thetemindex} \n"
postamble "\n\n \end{thetemindex} \n"

```

Pokud bylo takto postupováno, stačí dokument zkompilevat a zobrazit. Výsledek je patrný na obr. 4.4.

Definice příkazů	Písmo - typ, řez	\item.....28
\def..... 22	\bf.....32	makeidx..... 23
\newcommand.... 22, 29	Prostředí	\makeindex.... 23, 24
\renewcommand.... 22	\begin.....16	\newindex..... 23
Dokument	\end..... 16	\printindex... 27, 32
document.....17	Rejstřík	\subitem.....28
\documentclass 17	index..... 23	\subsubitem..... 28
\documentstyle 17	\index..... 24, 25, 27	theindex.....26
\usepackage..... 17	\index*.....24	Výplně
Nová Stránka	\indexentry.....25	\dotfill.....33
\nopagebreak.....32	\indexspace.....26	\hfill.....32

Obrázek 4.4: Tématický rejstřík řazený do tří sloupců

Kapitola 5

Závěr

Nemálo mých známých a příbuzných se mě ptalo, čím se zaobírám ve své práci. Když jsem jim stručně vysvětlil, o čem práce je, většina z nich se tvářila způsobem, že to je něco, co nemůže běžný uživatel využít. Z části měli samozřejmě pravdu, neboť počítačový laik si vystačí někdy i s obyčejným poznámkovým blokem. Problém pak nastává v případě, bude-li chtít vysázet o trochu složitější strukturu. K tomu už mu nepomůže kdejaký textový editor.

Proto jsem vypracoval tuto práci tak, aby z ní mohl čerpat informace jakýkoliv uživatel, nehledě na svou odbornost. Existuje několik zdrojů jak na internetu, tak mezi tištěnými publikacemi, které se zabývají problematikou tvorby rejstříků, ale žádný z nich jsem neshledal natolik použitelný, aby ho mohl využít libovolný uživatel. Vždy jsem zde nacházel hlavně nedostatky týkající se úplnosti zpracovaných dat.

V práci je popsána funkce \LaTeX u a programu *MakeIndex*. Dále jsem čtenáře seznámil s fakty a postupy, které jsou potřebné pro zdárné vytvoření a formátování rejstříku. Zbytek práce už pojednává o samotné tvorbě rejstříku a možnostech jeho úpravy.

Práce však nezohledňuje instalaci systému \LaTeX a jeho součástí, která se liší v použití OS či jeho distribuci, což někdo může považovat za chybu. Nicméně to není cílem práce. Navíc lze vyhledat na internetu spoustu kvalitních zdrojů.

Doufám, že tato práce bude pro čtenáře a uživatele systému \LaTeX užitečná a bude jim pomáhat při tvorbě a stylizaci jejich rejstříků.

Literatura

- [1] RYBIČKA, Jiří. *LaTeX pro začátečníky*, 3.vydání, Brno: Konvoj, 2003, 238 stran, ISBN 80-7302-049-1

- [2] OLŠÁK, Petr. *Typografický systém TeX*. 2.vydání, Brno: Konvoj, 2000, ISBN 80-85615-91-6

- [3] PARTL, Hubert & kol. *Ne příliš stručný úvod do systému LaTeX 2_ε: Neboli LaTeX 2_ε v 73 minutách*. Přeložil M.Kočer. [s.l.] : [s.n.], 1998. 85 s. Dostupný z WWW: <<http://www.penguin.cz/~kocer/texty/lshort2e/lshort2e-cz.pdf>>.

- [4] ŠVAMBERG, Michal. *Jak na LaTeX: rejstřík balíkem index* [online]. 2003 [cit. 2008-10-31]. Dostupný z WWW: <<http://www.root.cz/clanky/jak-na-latex-rejstrik-balikem-index>>, ISSN 1212-8309.

- [5] ŠVAMBERG, Michal. *Jak na LaTeX: rejstřík balíkem index II* [online]. 2003 [cit. 2008-10-31]. Dostupný z WWW: <<http://www.root.cz/clanky/jak-na-latex-rejstrik-balikem-index-2>>, ISSN 1212-8309.

- [6] WAGNER, Zdeněk. Tvorba rejstříku *Zpravodaj Československého sdružení uživatelů TeXu*. 1992. roč. 2, č. 4, s. 176-198. Dostupný z WWW: <<http://bulletin.cstug.cz/pdf/bul924.pdf>>.

- [7] *Zpravodaj Československého sdružení uživatelů TeXu*. 1992. roč. 2, č. 4. Dostupný z WWW: <<http://bulletin.cstug.cz/pdf/bul924.pdf>>.

- [8] POLÁCH, Eduard. *Pravidla sazby diplomových prací*. 1998. 37 s. Dostupný z WWW: <<http://home.pf.jcu.cz/~edpo/pravidla/pravidla.pdf>>.
- [9] SOJKA, Petr. *Vytváření rejstříků pomocí programů csindex, Make-index a xindy* [online]. 2008. [cit. 2008-11-10]. Dostupný z WWW: <<http://www.fi.muni.cz/~sojka/PB029/2008cv8.xhtml>>.

Přílohy

Zdrojové kódy jednotlivých rejstříků
Zdrojové kódy práce přiložené na CD

Zdrojové kódy jednotlivých rejstříků

A Rejstřík (obecný)

A.1 Soubor .idx

```

\indexentry {latex@LaTeX }{6}
\indexentry {DTP}{6}
\indexentry {tex@TeX }{6}
\indexentry {cstug@CS TUG}{7}
\indexentry {MakeIndex}{9}
\indexentry {csindex@CS Index}{9}
\indexentry {csindex@CS Index}{10}
\indexentry {dvi2dvi}{10}
\indexentry {platforma}{11}
\indexentry {Ubuntu}{11}
\indexentry {instalace}{11}
\indexentry {texlive@TeX Live}{11}
\indexentry {rejstřík| }{12}
\indexentry {rejstřík|)}{14}
\indexentry {tex@TeX }{14}
\indexentry {latex@LaTeX }{14}
\indexentry {soubor!tex}{14}
\indexentry {soubor!tfm}{14}
\indexentry {soubor!log}{14}
\indexentry {DVI}{15}
\indexentry {soubor!dvi}{15}
\indexentry {xdvi}{15}
\indexentry {dokument| }{15}
\indexentry {gedit}{15}
\indexentry {příkaz}{16}
\indexentry {makro|see{příkaz}}{16}
\indexentry {parametr}{16}
\indexentry {argument|see{parametr}}{16}
\indexentry {parametr!povinný}{16}
\indexentry {parametr!nepovinný}{16}
\indexentry {prostředí}{16}
\indexentry {komentář}{17}
\indexentry {dokument!struktura}{17}
\indexentry {dokument!hlavička}{17}
\indexentry {soubor!sty}{17}
\indexentry {balík}{17}
\indexentry {balík!czech}{17}
\indexentry {dokument!tělo}{17}
\indexentry {překlad}{17}
\indexentry {kompilace|see{překlad}}{17}
\indexentry {DVI}{18}
\indexentry {soubor!dvi}{18}
\indexentry {PDF}{18}
\indexentry {soubor!pdf}{18}
\indexentry {dvi2pdf}{18}
\indexentry {pdfcslatex}{18}
\indexentry {xdvi}{18}
\indexentry {dokument|)}{18}
\indexentry {soubor!ilg}{19}
\indexentry {MakeIndex!přepínače}{19}
\indexentry {MakeIndex!stylový soubor}{20}
\indexentry {MakeIndex!stylový soubor!specifikátor| }{20}
\indexentry {MakeIndex!stylový soubor!specifikátor|)}{22}
\indexentry {příkaz}{22}
\indexentry {balík!makeidx}{23}
\indexentry {balík!index}{23}
\indexentry {rejstřík!generování| }{26}
\indexentry {MakeIndex}{26}
\indexentry {csindex@CS Index}{26}
\indexentry {rejstřík!prolog}{27}
\indexentry {rejstřík!generování|)}{27}
\indexentry {rejstřík!formal| }{28}
\indexentry {rejstřík!struktura}{28}
\indexentry {rejstřík!položka!stránkový rozsah}{29}
\indexentry {rejstřík!položka!číslo stránky}{29}
\indexentry {rejstřík!prolog}{32}
\indexentry {rejstřík!titulek}{32}
\indexentry {rejstřík!formal|)}{36}

```

A.2 Soubor .ind

Pro generování rejstříku byl použit tento stylový soubor

```

item_1 "\n \\item --- "
item_2 "\n \\item --- --- "
item_01 "\n \\item --- "
item_x1 "\n \\item --- "
item_12 "\n \\item --- --- "
item_x2 "\n \\item --- --- "

```

Výsledný formát poté vypadá takto:

```

\begin{theindex}
\item balík, 16
\item --- czech, 16
\item --- index, 22
\item --- makeidx, 22

\indexspace

\item \CS Index, 9, 10, 24
\item \CS TUG, 7

\indexspace

\item dokument, 15--17
\item --- hlavička, 16
\item --- struktura, 16
\item --- tělo, 16
\item DTP, 6
\item DVI, 14, 17
\item dvi2dvi, 10
\item dvipdfm, 17

\indexspace

\item gedit, 15

\indexspace

\item instalace, 11

\indexspace

\item komentář, 16
\item kompilace, \see{překlad}{17}

\indexspace

\item \LaTeX , 6, 13

\indexspace

\item MakeIndex, 9, 24
\item --- přepínače, 18
\item --- stylový soubor, 19
\item --- --- specifikátor, 19--21
\item makro, \see{příkaz}{15}

\indexspace

\item parametr, 15
\item --- nepovinný, 15
\item --- povinný, 15
\item PDF, 17
\item pdfcslatex, 17
\item prostředí, 15
\item překlad, 17
\item příkaz, 15, 21

\indexspace

\item rejstřík, 12--13
\item --- forma, 26--32
\item --- generování, 24--26
\item --- položka
\item --- --- stránkový rozsah, 27
\item --- --- číslo stránky, 28
\item --- prolog, 26, 31
\item --- struktura, 27

\indexspace

\item soubor
\item --- dvi, 14, 17
\item --- ilg, 18
\item --- log, 14
\item --- pdf, 17
\item --- sty, 16
\item --- tex, 14
\item --- tfm, 14

\indexspace

\item \TeX , 6, 13
\item \TeX Live, 11

\indexspace

\item Ubuntu, 11

\indexspace

\item xdvi, 14, 17, 26
\end{theindex}

```

B Abecední seznam příkazů

B.1 Soubor .mdx

```

\indexentry {begin@\verb \begin |pop}{16}
\indexentry {end@\verb \end |pop}{16}
\indexentry {documentstyle@\verb \documentstyle }{17}
\indexentry {documentclass@\verb \documentclass }{17}
\indexentry {usepackage@\verb \usepackage }{17}
\indexentry {document@\texttt {document}}{17}
\indexentry {newcommand@\verb \newcommand |pop}{22}
\indexentry {renewcommand@\verb \renewcommand |pop}{22}
\indexentry {def@\verb \def |pop}{22}
\indexentry {makeidx@\texttt {makeidx}}{23}
\indexentry {index@\texttt {index}|pop}{23}
\indexentry {newindex@\verb \newindex |pop}{23}
\indexentry {makeindex@\verb \makeindex }{23}
\indexentry {index@\verb \index |pop}{24}
\indexentry {makeindex@\verb \makeindex }{24}
\indexentry {index*@\verb \index *}{24}
\indexentry {index@\verb \index }{25}
\indexentry {indexentry@\verb \indexentry }{25}
\indexentry {theindex@\texttt {theindex}}{26}
\indexentry {indexspace@\verb \indexspace }{26}
\indexentry {printindex@\verb \printindex |pop}{27}
\indexentry {index@\verb \index |}{27}
\indexentry {index@\verb \index }{27}
\indexentry {item@\verb \item }{28}
\indexentry {subitem@\verb \subitem }{28}
\indexentry {subsubitem@\verb \subsubitem }{28}
\indexentry {newcommand@\verb \newcommand }{29}
\indexentry {index@\verb \index |)}{30}
\indexentry {printindex@\verb \printindex }{32}
\indexentry {bf@\verb \bf }{32}
\indexentry {hfill@\verb \hfill }{32}
\indexentry {nopagebreak@\verb \nopagebreak }{32}
\indexentry {dotfill@\verb \dotfill }{33}

```

B.2 Soubor .mnd

Pro generování rejstříku byl použit tento stylový soubor

```
heading_prefix "{\\bf\\hfill "
heading_suffix "\\hfill}\\nopagebreak\n"
headings_flag 1
delim_0 "\\dotfill "
delim_1 "\\dotfill "
delim_2 "\\dotfill "
```

Výsledný formát poté vypadá takto:

```
\begin{theindex}
{\bf\hfill B\hfill}\nopagebreak

\item \verb \begin \dotfill \pop{16}
\item \verb \bf \dotfill 32

\indexspace
{\bf\hfill D\hfill}\nopagebreak

\item \verb \def \dotfill \pop{22}
\item \texttt {document}\dotfill 17
\item \verb \documentclass \dotfill 17
\item \verb \documentstyle \dotfill 17
\item \verb \dotfill \dotfill 33

\indexspace
{\bf\hfill E\hfill}\nopagebreak

\item \verb \end \dotfill \pop{16}

\indexspace
{\bf\hfill H\hfill}\nopagebreak

\item \verb \hfill \dotfill 32

\indexspace
{\bf\hfill I\hfill}\nopagebreak

\item \texttt {index}\dotfill \pop{23}
\item \verb \index \dotfill \pop{24}, 25, 27--30
\item \verb \index * \dotfill 24
\item \verb \indexentry \dotfill 25
\item \verb \indexspace \dotfill 26
\item \verb \item \dotfill 28

\indexspace
{\bf\hfill M\hfill}\nopagebreak

\item \texttt {makeidx}\dotfill 23
\item \verb \makeindex \dotfill 23, 24

\indexspace
{\bf\hfill N\hfill}\nopagebreak

\item \verb \newcommand \dotfill \pop{22}, 29
\item \verb \newindex \dotfill \pop{23}
\item \verb \nopagebreak \dotfill 32

\indexspace
{\bf\hfill P\hfill}\nopagebreak

\item \verb \printindex \dotfill \pop{27}, 32

\indexspace
{\bf\hfill R\hfill}\nopagebreak

\item \verb \renewcommand \dotfill \pop{22}

\indexspace
{\bf\hfill S\hfill}\nopagebreak

\item \verb \subitem \dotfill 28
\item \verb \subsubitem \dotfill 28

\indexspace
{\bf\hfill T\hfill}\nopagebreak

\item \texttt {theindex}\dotfill 26

\indexspace
{\bf\hfill U\hfill}\nopagebreak

\item \verb \usepackage \dotfill 17

\end{theindex}
```

C Tématický rejstřík

C.1 Soubor .tdx

```
\indexentry {pro begin@\verb \begin |pop}{16}
\indexentry {pro end@\verb \end |pop}{16}
\indexentry {dok documentstyle@\verb \documentstyle }{17}
\indexentry {dok documentclass@\verb \documentclass }{17}
\indexentry {dok usepackage@\verb \usepackage }{17}
\indexentry {dok document@\texttt {document}}{17}
\indexentry {def newcommand@\verb \newcommand |pop}{22}
\indexentry {der renewcommand@\verb \renewcommand |pop}{22}
\indexentry {def def@\verb \def |pop}{22}
\indexentry {rej makeidx@\texttt {makeidx}}{23}
\indexentry {rej index@\texttt {index}|pop}{23}
\indexentry {rej newindex@\verb \newindex |pop}{23}
\indexentry {rej makeindex@\verb \makeindex }{23}
\indexentry {rej index@\verb \index |pop}{24}
\indexentry {rej makeindex@\verb \makeindex }{24}
\indexentry {rej index*@\verb \index *}{24}
\indexentry {rej index@\verb \index }{25}
\indexentry {rej indexentry@\verb \indexentry }{25}
\indexentry {rej theindex@\texttt {theindex}}{26}
\indexentry {rej indexspace@\verb \indexspace }{26}
\indexentry {rej printindex@\verb \printindex |pop}{27}
\indexentry {rej index@\verb \index }{27}
\indexentry {rej item@\verb \item }{28}
\indexentry {rej subitem@\verb \subitem }{28}
\indexentry {rej subsubitem@\verb \subsubitem }{28}
\indexentry {def newcommand@\verb \newcommand }{29}
\indexentry {rej printindex@\verb \printindex }{32}
\indexentry {pís bf@\verb \bf }{32}
\indexentry {výp hfill@\verb \hfill }{32}
\indexentry {nov nopagebreak@\verb \nopagebreak }{32}
\indexentry {výp dotfill@\verb \dotfill }{33}
```

C.2 Soubor .tnd

Pro generování rejstříku byl použit tento stylový soubor

```
preamble "\n \\begin{thetemindex} \n"
postamble "\n\n \\end{thetemindex} \n"
```

Výsledný formát po ruční úpravě vypadá takto:

```
\begin{thetemindex}
\bf Definice příkazů\hfill\nopagebreak
\item \verb \begin \dotfill \pop{16}
\item \verb \end \dotfill \pop{16}
\indexspace
\bf Rejstřík\hfill\nopagebreak
\item \verb \def \dotfill \pop{22}
\item \verb \newcommand \dotfill \pop{22}, 29
\item \verb \renewcommand \dotfill \pop{22}
\item \texttt {index}\dotfill \pop{23}
\indexspace
\item \verb \index \dotfill \pop{24}, 25, 27
\item \verb \index *\dotfill 24
\item \verb \indexentry \dotfill 25
\item \verb \indexspace \dotfill 26
\item \verb \item \dotfill 28
\item \texttt {makeidx}\dotfill 23
\item \verb \makeindex \dotfill 23, 24
\item \verb \newindex \dotfill \pop{23}
\indexspace
\item \verb \printindex \dotfill \pop{27}, 32
\item \verb \subitem \dotfill 28
\item \verb \subsubitem \dotfill 28
\item \texttt {theindex}\dotfill 26
\bf Nová Stránka\hfill\nopagebreak
\item \verb \nopagebreak \dotfill 32
\indexspace
\bf Písmo - typ, řez\hfill\nopagebreak
\item \verb \dotfill \dotfill 33
\item \verb \hfill \dotfill 32
\indexspace
\bf Prostředí\hfill\nopagebreak
\end{thetemindex}
```

C.3 Prolog

Pro tento rejstřík je nedefinován příkaz `prologTr`

```
\def\prologTr{
```

Rejstřík zahrnuje abecední seznam příkazů uspořádaných do tématických skupin. Za hesly jsou uvedeny odkazy na příslušné stránky. Čísla vytištěná tučně označují stránky, kde je makro popsáno. Všechna ostatní ukazují na místo, kde je makro použito. Položka, která nezačíná znakem zpětného lomítka označuje balík maker.

```

\begin{center}
{\bf Seznam témat:}
\end{center}

\begin{multicols}{2}
\begin{itemize}
\item Definice příkazů
\item Dokument
\item Nová stránka
\item Písmo - typ, řez
\item Prostředí
\item Rejstřík
\item Výplně
\end{itemize}
\end{multicols}
\hspace*{1cm}
}

```

C.4 Sloupce

Pro řazení rejstříku do tří sloupců je nadefinováno nové prostředí `thetemindex`.

```

\newenvironment{thetemindex}{\chapter*{Tematický rejstřík}
\begin{multicols}{3}[\prologTr]
\IndexParms
}%
{
\end{multicols}
}
\def\IndexParms{
\columnsep 15pt
\parskip 0pt plus 1pt
\rightskip 15pt
\parfillskip=-15pt
\mathsurround 0pt
\columnsep 15pt
\parindent 0pt
\def\main##1{{\bf ##1}}
\def\item{\par\hangindent 30pt}%
\def\subitem{\item\hspace*{15pt}}%
\def\subsubitem{\item\hspace*{25pt}}%
}

```

Rejstřík

- argument, *viz* parametr
- balík, 17
 - czech, 17
 - index, 23
 - makeidx, 23
- \mathcal{C} SIndex, 9, 10, 26
- \mathcal{C} STUG, 7
- dokument, 15–18
 - hlavička, 17
 - struktura, 17
 - tělo, 17
- DTP, 6
- DVI, 15, 18
- dvi2dvi, 10
- dvipdfm, 18
- gedit, 15
- instalace, 11
- komentář, 17
- kompilace, *viz* překlad
- \LaTeX , 6, 14
- MakeIndex, 9, 26
 - přepínače, 19
 - stylový soubor, 20
 - — specifikátor, 20–22
- makro, *viz* příkaz
 - nepovinný, 16
 - povinný, 16
- PDF, 18
- pdfcslatex, 18
- platforma, 11
- prostředí, 16
- překlad, 17
- příkaz, 16, 22
- rejstřík, 12–14
 - forma, 28–36
 - generování, 26–27
 - položka
 - — stránkový rozsah, 29
 - — číslo stránky, 29
 - prolog, 27, 32
 - struktura, 28
 - titulek, 32
- soubor
 - dvi, 15, 18
 - ilg, 19
 - log, 14
 - pdf, 18
 - sty, 17
 - tex, 14
 - tfm, 14
- \TeX , 6, 14
- \TeX Live, 11
- Ubuntu, 11
- xdvi, 15, 18

Abecední seznam příkazů

Čísla vytištěná tučně označují stránky, kde je příslušné makro popsáno. Všechna ostatní ukazují na místo, kde je makro použito. Položka, která nezačíná znakem zpětného lomítka označuje balík maker.

B	
<code>\begin</code>	16
<code>\bf</code>	32
D	
<code>\def</code>	22
<code>document</code>	17
<code>\documentclass</code>	17
<code>\documentstyle</code>	17
<code>\dotfill</code>	33
E	
<code>\end</code>	16
H	
<code>\hfill</code>	32
I	
<code>index</code>	23
<code>\index</code>	24, 25, 27–30
<code>\index*</code>	24
<code>\indexentry</code>	25
<code>\indexspace</code>	26
<code>\item</code>	28
M	
<code>makeidx</code>	23
<code>\makeindex</code>	23, 24
N	
<code>\newcommand</code>	22, 29
<code>\newindex</code>	23
<code>\nopagebreak</code>	32
P	
<code>\printindex</code>	27, 32
R	
<code>\renewcommand</code>	22
S	
<code>\subitem</code>	28
<code>\subsubitem</code>	28
T	
<code>theindex</code>	26
U	
<code>\usepackage</code>	17

Tématický rejstřík

Rejstřík zahrnuje abecední seznam příkazů uspořádaných do tématických skupin. Za hesly jsou uvedeny odkazy na příslušné stránky. Čísla vytisknutá tučně označují stránky, kde je makro popsáno. Všechna ostatní ukazují na místo, kde je makro použito. Položka, která nezačíná znakem zpětného lomítka označuje balík maker.

Seznam témat:

- Definice příkazů
- Dokument
- Nová stránka
- Písmo - typ, řez
- Prostředí
- Rejstřík
- Výplně

Definice příkazů	Písmo - typ, řez	<code>\item</code> 28
<code>\def</code> 22	<code>\bf</code> 32	<code>makeidx</code> 23
<code>\newcommand</code> 22, 29	Prostředí	<code>\makeindex</code> 23, 24
<code>\renewcommand</code> 22	<code>\begin</code> 16	<code>\newindex</code> 23
Dokument	<code>\end</code> 16	<code>\printindex</code> 27, 32
<code>document</code> 17	Rejstřík	<code>\subitem</code> 28
<code>\documentclass</code> 17	<code>index</code> 23	<code>\subsubitem</code> 28
<code>\documentstyle</code> 17	<code>\index</code> 24, 25, 27	<code>theindex</code> 26
<code>\usepackage</code> 17	<code>\index*</code> 24	Výplně
Nová Stránka	<code>\indexentry</code> 25	<code>\dotfill</code> 33
<code>\nopagebreak</code> 32	<code>\indexspace</code> 26	<code>\hfill</code> 32