

Tvorba videotutoriálů pro práci s typografickým systémem LaTeX

LaTeX DTP System Video Tutorials for Beginners

Petr Kohout

Bakalářská práce

Vedoucí práce: Ing. Jan Skapa, Ph.D.

Ostrava, 2023

Zadání bakalářské práce

Student:

Petr Kohout

Studijní program:

B0714A060010 Telekomunikační technika

Téma:

Tvorba videotutoriálů pro práci s typografickým systémem LaTeX
LaTeX DTP System Video Tutorials for Beginners

Jazyk vypracování:

čeština

Zásady pro vypracování:

Cílem práce je popsat detailně instalaci systému LaTeX na platformách Linux Mint (resp. Ubuntu) a Windows včetně konfigurace editoru pro sazbu odborných publikací. Konečným výstupem bude soubor video tutoriálů, které začátečníka provedou instalací systému LaTeX a konfigurací editoru pro efektivní sazbu dokumentů.

1. Proveďte rešerši jednotlivých instalací systému LaTeX a jeho variant (ConTeXt, LuaTeX) včetně systémy Lyx. Zaměřte se na univerzálnost použití daného systému napříč platformami Windows a Linux.
2. Popište detailně instalaci systému TeXLive v základní (base) konfiguraci na platformách Windows a Linux.
3. Proveďte rešerši jednotlivých LaTeX editorů. Zaměřte se na univerzálnost použití daného editoru napříč platformami Windows a Linux.
4. Popište detailně instalaci (po konzultaci s vedoucím práce) zvoleného editoru na platformách Windows a Linux.
5. Vytvořte dokument s textem "Příliš žluťoučký kůň úpěl ďábelské ódy" a popište kompilaci dokumentu z prostředí editoru a přes příkazovou řádku.
6. Popište konfiguraci dokumentu pro korektní sazbu češtiny včetně znaků s akcenty. Popište možnosti změny stupně a řezu písma. Popište možnosti zvýrazňování textu v systému LaTeX.
7. Popište instalaci doplňkových balíčků a jejich integraci do prostředí TeXLive.
8. Popište sazbu obrázků do dokumentu včetně konfigurace parametrů pro sazbu plovoucího prostředí FIGURE a parametrů příkazu `\includegraphics`.
9. Popište sazbu tabulek do dokumentu včetně konfigurace parametrů pro sazbu plovoucího prostředí TABLE. Popište také možnosti sazby tabulek se sloučenými řádky a sloupci. Popište možnosti sazby tabulek a obrázků, otočených na stránce o 90 °.
10. Popište sazbu matematických vztahů. Zdůrazněte instalaci a využití balíčku AMSMATH.
11. Popište sazbu výpisů kódu (v bloku a v řádku textu). Zdůrazněte instalaci a využití balíčku LISTINGS včetně konfigurace.
12. Popište konfiguraci záhlaví a zápatí dokumentu.
13. Popište tvorbu křížových odkazů (`\ref`, `\cite`, `\eqref`) a hypertextových odkazů (`\href`, `\url`). Zdůrazněte instalaci a využití balíčku HYPERREF včetně konfigurace.
14. Popište tvorbu rejstříku (Index) a konfiguraci obsahu (Table of Contents).
15. Popište sazbu seznamu použité literatury s využitím balíčku BibTeX, případně Biber.
16. Popište možnosti využití nástrojů pro online sazbu v prostředí LaTeX a sdílení textu mezi jednotlivými spolupracovníky.

Seznam doporučené odborné literatury:

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

[2] KOPKA, Helmut a Patrick W. DALY. *LaTeX: podrobný průvodce*. Přeložil Jan GREGOR. Brno: Computer Press, 2004. DTP & grafika. ISBN 80-722-6973-9.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Jan Skapa, Ph.D.**

Datum zadání: 01.09.2022

Datum odevzdání: 30.04.2023

Garant studijního programu: doc. Ing. Petr Šiška, Ph.D.

V IS EDISON zadáno: 05.10.2022 12:18:59

Abstrakt

Cílem této bakalářské práce je vytvořit soubor videotutoriálů pro práci s typografickým systémem LaTeX a porovnání jeho jednotlivých variant a editorů. Porovnávat se budou tři systémy: LaTeX, ConTeXt a LuaTeX. Tato práce se také bude zabývat srovnáním vybraných LaTeX editorů. Kromě teoretických popisů bude výsledkem této práce soubor videotutoriálů, které začátečníkovi pomohou se sazbou dokumentů.

Klíčová slova

typografie; LaTeX; TeXLive; tutoriál; video tutoriál

Abstract

The aim of this thesis is to create a set of video tutorials for using the typesetting system LaTeX and comparison of its individual extensions and editors. The comparison will include three variants: LaTeX, ConTeXt and LuaTeX. This thesis will also deal with comparison of selected LaTeX editors. Apart from theoretical descriptions the result of this thesis will be set of video tutorials, which will help beginners with document typesetting.

Keywords

typography; LaTeX; TeXLive; tutorial; video tutorial

Poděkování

Chtěl bych na tomto místě poděkovat Ing. Janu Skapovi, Ph.D., za ochotný přístup, všechny rady a vedení při tvorbě této bakalářské práce.

Obsah

Seznam použitých symbolů a zkratk	8
Seznam obrázků	9
Seznam tabulek	10
1 Úvod	11
2 Porovnání instalací systému LaTeX	12
2.1 LaTeX	12
2.2 ConTeXt	12
2.3 LuaTeX	13
2.4 LyX	13
3 TeX Live	14
3.1 Instalace systému TeX Live	14
4 Editory pro LaTeX	17
4.1 Texmaker	17
4.2 TeXstudio	17
4.3 Overleaf	18
5 Instalace editoru TeXstudio	19
5.1 Windows	19
5.2 Linux	19
6 Tvorba dokumentu	20
6.1 Sazba prvního dokumentu	20
6.2 Kompilace dokumentu	21
6.3 Sazba češtiny	21
6.4 Stupeň písma	21

6.5	Řezy písma	22
6.6	Instalace doplňkových balíčků	22
6.7	Sazba obrázků	24
6.8	Sazba tabulek	25
6.9	Sazba matematických vztahů	27
6.10	Sazba výpisů kódu	29
6.11	Záhlaví a zápatí dokumentu	31
6.12	Tvorba křížových a hypertextových odkazů	32
6.13	Tvorba rejstříku a obsahu	34
6.14	Sazba seznamu použité literatury	35
6.15	Nástroje pro online sazbu a sdílení dokumentů	36
7	Závěr	38
	Literatura	39

Seznam použitých zkratek a symbolů

- WYSIWYG – What you see is what you get
- DVI – Device independent
- PDF – Portable document format
- URL – Uniform Resource Locator

Seznam obrázků

3.1	První fáze instalace TeX Live pro Windows	15
3.2	Okno hlavního instalátoru TeX Live pro Windows	15
3.3	Pokročilé možnosti instalace TeX Live pro Windows	16
3.4	Nabídka jazyků a balíčků pro instalaci TeX Live	16
6.1	Přeložený text s diakritikou před přidáním balíčku babel	21
6.2	Přeložený text s diakritikou po přidání balíčku babel	21
6.3	TeX Live Manager - instalace balíčku	23
6.4	Záhlaví a zápatí jednostranného dokumentu	31
6.5	Záhlaví a zápatí na sudé stránce - dvoustranný dokument	32
6.6	Záhlaví a zápatí na liché stránce - dvoustranný dokument	32
6.7	Overleaf - sdílení projektu	36
6.8	Cocalc - sdílení projektu	37

Seznam tabulek

6.1	Stupně písma	22
-----	------------------------	----

Kapitola 1

Úvod

\TeX je volně šířený systém počítačové vazby, který byl vytvořen jako odpověď na komerční programové systémy pro zpracování textů. Tyto systémy, pracující v textovém režimu, dnes díky technickému pokroku přecházejí do grafického systému s velkým množstvím funkcí. Výrobci se snaží o uživatelsky přívětivý produkt ale často zapomínají na zpětnou kompatibilitu jejich verzí. Uživatelé jsou proto nuceni starší verze nahrazovat novými a opakovaně za ně platit.

Proto Donald E. Knuth ze Stanfordské univerzity vytvořil systém počítačové vazby \TeX . Tento systém byl určen především pro uživatele, kteří chtějí využít maximálního výkonu výpočetní techniky a tvořit graficky dokonalé dokumenty. Koncepce systému byla navržena tak promyšleně, že od roku 1983, kdy se rozšířila první verze, nedošlo k žádným zásadním změnám.

V systému je definováno přes tři sta příkazů a obsahuje nástroj pro rozšiřování vlastních schopností. Díky tomu nad ním byly vytvořeny nadstavby, které umožňují snadnější zápis sázeného textu. Jednou z nejvýznamnějších nadstaveb je \LaTeX . \LaTeX je určen nejen pro sazbu matematických textů ale je vhodný i pro sazbu běžných publikací jako jsou články, zprávy nebo knihy. [1]

Mimo \LaTeX vzniklo i několik dalších nadstaveb. Tato bakalářská práce se bude zabývat mimo jiné také popisem nadstaveb \ConTeXt , \LuaTeX a systémem \Lyx .

Kapitola 2

Porovnání instalací systému LaTeX

Ještě předtím než začneme s instalací systému LaTeX tak si nejprve popíšeme a srovnáme jednotlivé varianty.

2.1 LaTeX

LaTeX je rozšíření originálního programu TeX. TeX obsahuje jen základní příkazy a není s nimi jednoduché vytvářet složitější dokumenty a proto vzniklo toto rozšíření. LaTeX dokument se skládá z jednoho nebo více zdrojových dokumentů, prostého textu a příkazů.

Z dokumentu je poté zpracován TeX programem, který vytvoří binární soubor ve formátu DVI. Tento soubor obsahuje přesné instrukce pro sázení jednotlivých znaků. Abychom mohli vidět vysázený text, je nutné aby byl soubor zpracován programem, který ho na daném zařízení zobrazí. Pokud požadujeme dokument ve formátu PDF lze využít zvláštního překladače pdfTeX, který místo souboru formátu DVI vytváří přímo formát PDF. [2]

Pro použití LaTeXu na platformě Windows lze využít distribuci TeX Live, která obsahuje kompletní systém TeX, LaTeX a editor k psaní dokumentů. Na Linuxu lze TeX Live nainstalovat jednoduše pomocí příkazu: `sudo apt-get install texlive-full`.

2.2 ConTeXt

LaTeX není jediný systém, který rozšířil původní TeX. ConTeXt je další rozšíření programu TeX. ConTeXt cílí především na poskytnutí jednoduchého uživatelského rozhraní s rozsáhlou podporou pro barvy, pozadí, hypertextové odkazy a prezentace. V současnosti je ConTeXt distribuován ve dvou verzích - Mark II, který běží na pdfTeX ale již není v aktivním vývoji a Mark IV, který běží na LuaTeX a je stále vyvíjen.

K instalaci na systému Windows stačí stáhnout adresář [3] a spustit instalační soubor v něm obsažený. Při použití Linuxu je postup podobný - vytvoříme si nový adresář a vložíme do něho stažený ZIP soubor [3], který extrahujeme a spustíme instalaci příkazem: `sh install.sh`

2.3 LuaTeX

LuaTeX je rozšířenou verzí pdfTeX za pomoci skriptovacího jazyku Lua. Programátor si může snadno naprogramovat a implementovat vlastní rozšíření do svého dokumentu.

Pro instalaci LuaTeX na systému Linux stačí příkaz: `sudo apt install texlive-luatex`. V operačním systému Windows stačí nainstalovat Tex Live a v něm si najdeme a nainstalujeme balíček LuaTeX.

2.4 LyX

LyX se svým uživatelským rozhraním podobá WYSIWYG textovým procesorům jako je například Microsoft Word. Je postavený na systému LaTeX ale k práci není potřeba znát příkazy LaTeXu. Pro běžné vytváření dokumentů jsou k dispozici jednotlivé položky v menu grafického rozhraní. Pro pokročilejší uživatele LyX nabízí možnost i upravovat dokument pomocí příkazů. [4]

Abychom mohli nainstalovat LyX na Windows musíme si nejprve nainstalovat TeX. Stejně jako v předchozích případech to bude TeX Live. Poté stačí stáhnout a nainstalovat LyX. V systému Linux je instalace velmi jednoduchá - postačí příkaz: `sudo apt install lyx`.

Kapitola 3

TeX Live

TeX Live je zamýšlen jako jednoduchý způsob jak začít pracovat se systémem TeX. Poskytuje kompletní systém TeX s podporou pro Linux, macOS a Windows. Obsahuje balíčky maker, písem a podporu několika jazyků. [5]

3.1 Instalace systému TeX Live

V následující části si popíšeme instalaci systému TeX Live na platformách Windows a Linux.

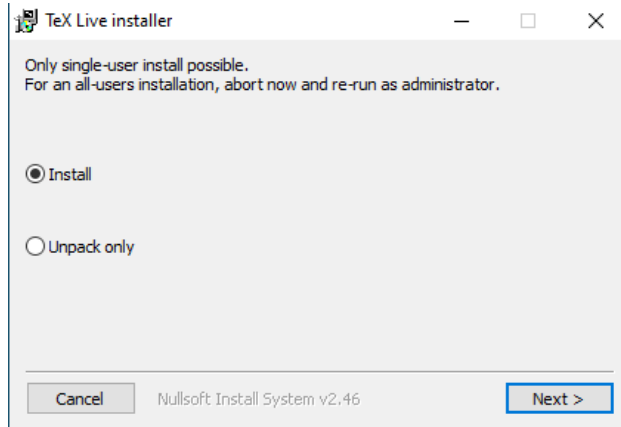
3.1.1 Windows

Nejprve si z oficiálního webu TeX Live [6] v sekci Easy Install stáhneme instalační soubor `install-tl-windows.exe`. Dvojklikem stažený soubor spustíme. Zobrazí se nám okno 3.1, kde si vybereme možnost "Install". Tímto se instalátor extrahuje a zobrazí se okno instalátoru 3.2. Jedná se pouze o jednoduchou verzi, kde je možné upravit jen základní nastavení a instaluje se TeX společně se všemi balíčky a jazyky, které pro nás mohou být zbytečné. Proto si vybereme možnost "Advanced". Ta nám nabídne mnoho dalších možností přizpůsobení 3.3.

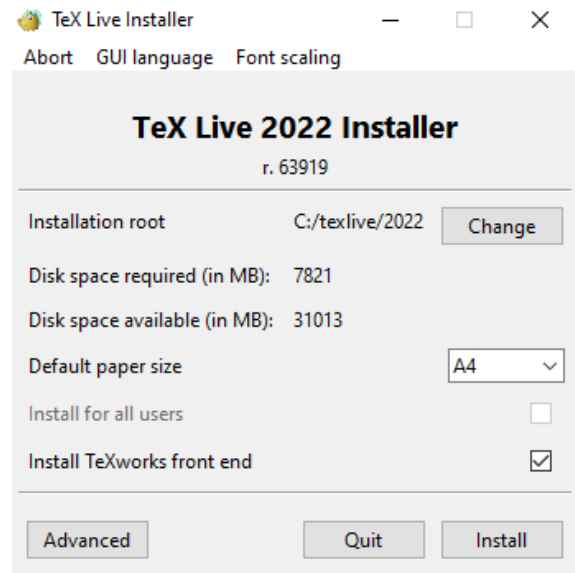
Zde si vybereme umístění instalace, schéma a kolekce balíčků. V položce "Scheme" si můžeme vybrat jedno z předpřipravených schémat. Full scheme obsahuje všechny balíčky a jazyky, proto také zabírá hodně místa v úložišti. Dále jsou na výběr schémata Medium, Small, Basic a Minimal. Je zde také možnost Custom. Po vybrání této možnosti si po kliknutí na tlačítko "Customize" vybereme balíčky a jazyky podle naší potřeby 3.4. V rámci tohoto návodu si vybereme schéma Basic. To nám značně sníží velikost instalace z původních 7800 MB na 416 MB. Nyní klikneme na tlačítko "Install" a počkáme až se nám TeX Live nainstaluje. Funkčnost instalace si ověříme přeložením TeX dokumentu příkazem `pdflatex dokument.tex`. Pokud se dokument přeloží a vznikne výsledný PDF soubor je naše instalace v pořádku.

Mimo výsledného PDF souboru, vzniknou také soubory s příponou `.log` a `.aux`. Soubor `log` obsahuje zprávu o posledním překladu včetně výpisu všech načtených souborů a balíčků použitých

v dokumentu. Překlad LaTeX dokumentu může někdy zabrat více průchodů a proto se soubor aux používá pro ukládání informací mezi cykly procesu kompilace LaTeX.



Obrázek 3.1: První fáze instalace TeX Live pro Windows

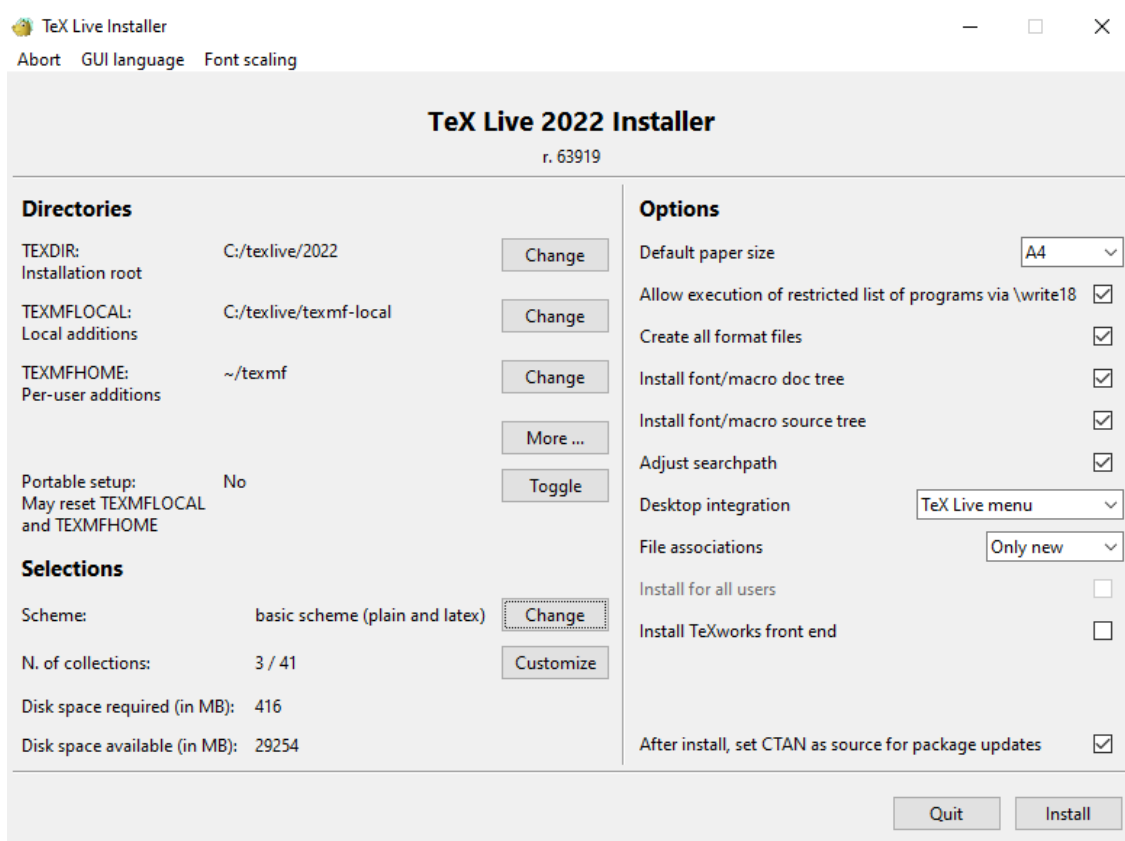


Obrázek 3.2: Okno hlavního instalátoru TeX Live pro Windows

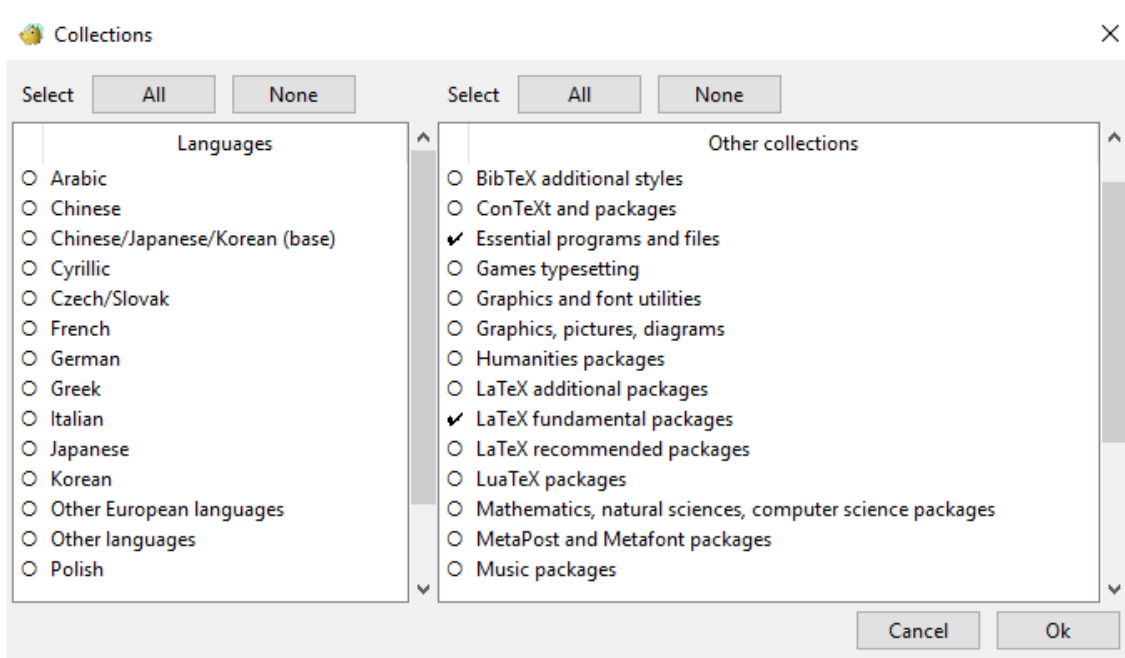
3.1.2 Linux

Pro tento návod využijeme distribuci Linux Mint a nainstalujeme si TeX Live v základní (base) konfiguraci.

Otevřeme terminál a zadáme příkaz: `sudo apt install texlive-latex-base`. To nám nainstaluje TeX Live společně se základními balíčky pro LaTeX. Instalaci si ověříme stejně jako v operačním systému Windows. Vytvoříme si TeX dokument a ten přeložíme přes příkazovou řádku příkazem `pdflatex dokument.tex`. Pokud se nám dokument v pořádku přeloží a vznikne výsledný PDF soubor je systém TeX Live nainstalován správně.



Obrázek 3.3: Pokročilé možnosti instalace TeX Live pro Windows



Obrázek 3.4: Nabídka jazyků a balíčků pro instalaci TeX Live

Kapitola 4

Editory pro LaTeX

Pro psaní LaTeX dokumentů existuje několik editorů. Tyto editory můžeme rozdělit do dvou skupin - WYSIWYG editory a zdrojové editory. Mezi WYSIWYG patří například Microsoft Word nebo LibreOffice. Jsou to editory, které uživateli umožňují napřímo upravovat a manipulovat s obsahem dokumentu. Výstup se uživateli ihned zobrazuje. U zdrojových editorů uživatel pomocí značek udává jak se bude obsah dokumentu sázet. Výsledek není hned k dispozici, uživatel nejprve musí zdrojový soubor zkompileovat.

4.1 Texmaker

Texmaker je multiplatformní LaTeX editor pro Linux, macOS a Windows. Výhodou je podpora unicode, kontrola pravopisu, automatické doplňování příkazů a zvýraznění syntaxe. Obsahuje vestavěný PDF prohlížeč. Jedná se o zdrojový editor. Výhodou oproti dále zmíněnému TeXstudio je menší velikost instalace - 200 MB. Instalace Texstudia má zhruba 500 MB.

4.2 TeXstudio

TeXstudio je stejně jako Texmaker multiplatformní LaTeX editor. Je postaven na programu Texmaker a proto s ním většinu funkcí sdílí. Navíc však obsahuje automatický překlad při změně v dokumentu. Další výhodou je vylepšené doplňování příkazů, například při psaní příkazu pro citování literatury nám TeXstudio našeptává i názvy jednotlivých zdrojů tak jak je máme nadefinované v souboru s bibliografií. Mezi další výhody, které už pro některé uživatele nemusí být tak zásadní, patří to že po přeložení dokumentu zůstane PDF výstup na stejném místě v dokumentu jako se nacházíme v editoru. Můžeme tak hned vidět změnu a nemusíme skrolovat na místo změny. Oproti editoru Texmaker zde máme více možností nastavení.

4.3 Overleaf

Overleaf je online editor pro kolaborativní psaní textů pomocí programu LaTeX. Pro psaní textů není třeba stahovat program a stačí pouze webový prohlížeč. Výhodou je jednoduchost použití, není třeba nic instalovat a sdílení mezi několika osobami. Pro sdílení obsahuje Overleaf i užitečné funkce jako je integrovaný chat a komentáře v dokumentu. Overleaf poskytuje také možnost automatické kompilace, takže můžeme při změně ihned vidět výsledné PDF. Nevýhodou je časový limit jedné minuty na kompilaci v neplacené verzi.

Kapitola 5

Instalace editoru TeXstudio

Na základě předchozího srovnání editorů pro LaTeX jsem se rozhodl pro editor TeXstudio. V následující části si popíšeme jeho instalaci na systémech Windows a Linux Mint.

5.1 Windows

Pro instalaci TeXstudio na operačním systému Windows stačí na webu [7] stáhnout instalační soubor kliknutím na fialové tlačítko "Download now". Spustíme stažený soubor a instalátor nás provede instalací. Stačí si pouze vybrat umístění na disku.

5.2 Linux

Na systému Linux Mint je instalace editoru přímočará. Otevřeme terminál a zadáme příkaz: `sudo apt install texstudio`. Tímto příkazem se editor TeXstudio nainstaluje.

Kapitola 6

Tvorba dokumentu

6.1 Sazba prvního dokumentu

Pro otestování zda nám funguje LaTeX tak jak má si vytvoříme první dokument. Otevřeme si editor TeXstudio a vybereme Soubor → Nový. Soubor s příponou .tex si uložíme v počítači. Do souboru napíšeme následující kód.

```
\documentclass{book}
\begin{document}
  Příliš žluťoučký kůň úpěl ďábelské ódy.
\end{document}
```

TeX soubor je rozdělen na hlavičku, ve které se nachází nastavení pro vlastnosti dokumentu jako je třída dokumentu, přídatné balíčky, velikost písma, titulek, a tělo dokumentu, kde píšeme náš text.

Třída dokumentu udává jakým způsobem bude dokument zpracován. Třidu vybereme pomocí příkazu `\documentclass[volby]{třída}`. Máme na výběr ze tří základních tříd: `article` (pro článek), `book` (pro knihu) a `report` (pro zprávu). Příkaz `\documentclass` může mít ještě volitelný parametr, ve kterém můžeme určit velikost písma (`12pt`), formát papíru (`a4paper`) nebo zda chceme výstup formátovat na šířku (`landscape`).

Balíčky můžeme k dokumentu připojit příkazem `\usepackage[volby]{balík}`. Tyto soubory v sobě ukrývají definice příkazů, které rozšiřují funkčnost programu.

Tělo dokumentu se nachází mezi příkazy `\begin{document}` a `\end{document}`. Zde budeme psát vlastní text. V našem případě to bude text: Příliš žluťoučký kůň úpěl ďábelské ódy. Ten obsahuje všechny české znaky a díky tomu můžeme zjistit, zda naše instalace plně akceptuje češtinu.[1]

6.2 Kompilace dokumentu

Pro získání výstupu ve formátu PDF je nutné kód nejprve zkompileovat. To můžeme provést přes grafické rozhraní editoru TeXstudio a to kliknutím na dvě zelené šipky v horním panelu, to nám kód zkompileje a zobrazí výsledné PDF. Ve výsledném PDF dokumentu můžeme vidět, že jsou háčky nad písmeny *t* a *ř* špatně vysázeny. Jak správně sázet české znaky si popíšeme v následující kapitole.

Druhý způsob jak zkompileovat kód je za pomoci příkazové řádky. V prostředí Linux Mint nebo Windows otevřeme terminál, respektive příkazovou řádku a náš TeX soubor zkompilejeme příkazem `pdflatex dokument.tex`. Tímto získáme výsledný soubor `dokument.pdf`.

6.3 Sazba češtiny

V předchozí části jsme zjistili, že naše instalace nesprávně sází některé české znaky (Obr. 6.1). Pro vyřešení tohoto problému musíme použít balík `babel` a to příkazem `\usepackage[czech]{babel}`. Tento balík který řeší například správné dělení slov, správnou sazbu diakritiky nebo české pojmenování kapitol místo anglického "Chapter". Do hranatých závorek můžeme přidat další jazyky a mít tak vícejazyčný dokument. Dokument po přeložení s balíkem `babel` má již správně vysázenou diakritiku (Obr. 6.2).

Příliš žluťoučký kůň úpěl ďábelské ódy.

Obrázek 6.1: Přeložený text s diakritikou před přidáním balíčku `babel`

Příliš žluťoučký kůň úpěl ďábelské ódy.

Obrázek 6.2: Přeložený text s diakritikou po přidáním balíčku `babel`

6.4 Stupeň písma

Pro změnu stupně písma v celém dokumentu je nutné u výběru třídy dokumentu zadat vybraný stupeň. Základní stupeň je 10 pt. Pokud chceme například celý dokument ve stupni písma 12 pt použijeme příkaz `\documentclass[12pt]{article}`. Pro změnu stupně písma v libovolné části dokumentu jsou k dispozici předem definované příkazy: [1]

Tabulka 6.1: Stupně písma

Velikost	Příkaz	Velikost	Příkaz
5 pt	<code>\tiny</code>	12 pt	<code>\large</code>
7 pt	<code>\scriptsize</code>	14,4 pt	<code>\Large</code>
8 pt	<code>\footnotesize</code>	17,28 pt	<code>\LARGE</code>
9 pt	<code>\small</code>	20,74 pt	<code>\huge</code>
10 pt	<code>\normalsize</code>	24,88 pt	<code>\Huge</code>

Příklad:

Tento text bude `{\Large větší}` a tento bude `{\scriptsize menší}`.

Tento text bude větší a tento bude menší.

6.5 Řezy písma

Různé řezy písma se používají ke zvýrazňování důležitých částí textu, ke zpřehlednění text a k usnadnění četby. Máme k dispozici několik příkazů ke změně řezu písma.

Příklad:

Tento text bude `\textbf{tučně}`, tento `\textit{kurzívou}`, tento `\underline{podtržený}` a tento `\texttt{strojově}`.

Tento text bude **tučně**, tento *kurzívou*, tento podtržený a tento strojově.

6.6 Instalace doplňkových balíčků

Balíčky přidávají do systému LaTeX další příkazy, které ulehčují sázení dokumentů a nabízí nové možnosti úpravy dokumentu. Podle vybrané instalace systému TeX Live už nějaké balíčky v počítači stažené máme.

Pro stažení dalších balíčků v operačním systému Windows využijeme TeX Live Manager, který je obsažen v instalaci TeX Live. Zde (6.3) si v sekci "Stav" vybereme možnost "Všechny" pro zobrazení všech dostupných balíčků. Poté si vyhledáme požadovaný balíček. Ve výsledku vyhledávání označíme náš balíček a klikneme na tlačítko "Instalace označeného". Nyní jen stačí v editoru naimportovat balíček příkazem `\usepackage{balíček}`.

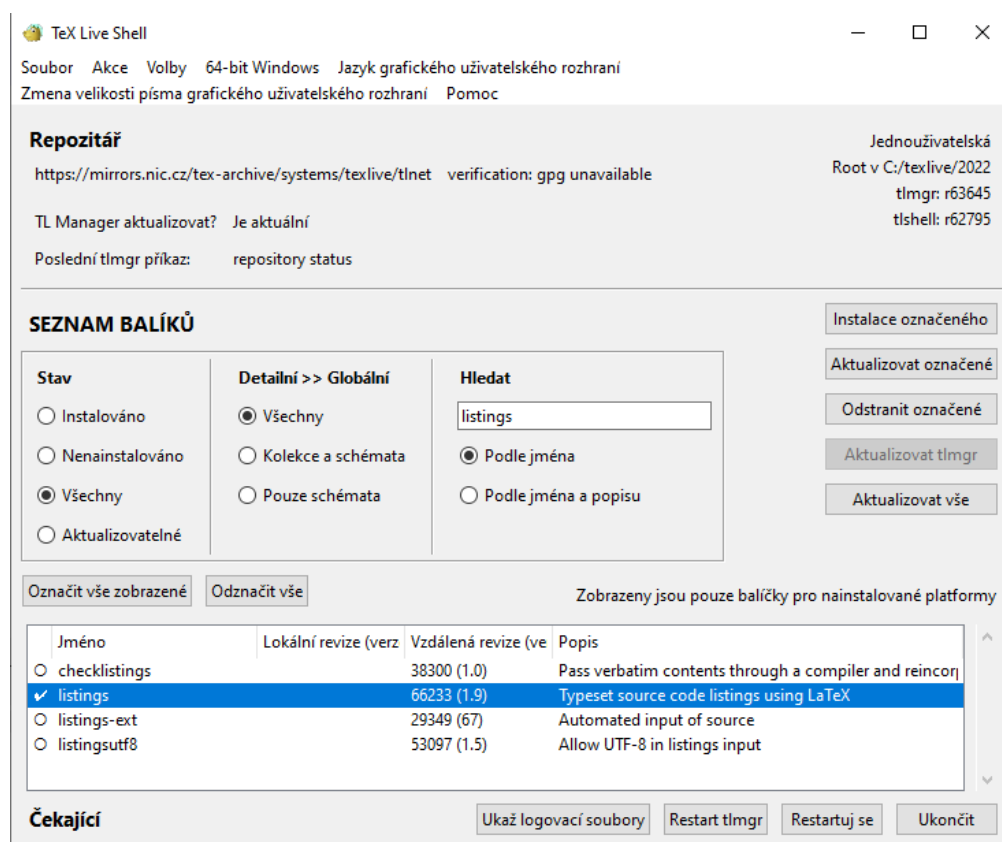
Instalace přídatných balíčků je možná také přes příkazovou řádku a to příkazem `tlmgr install <balíček>`. Pro instalaci tak musíme znát název balíčku. Ten najdeme na webu CTAN [8], který nabízí archiv všech balíčků.

V systému Linux je nutné před instalací nejprve změnit repozitář doplňků na verzi 2021, jelikož jsme TeX Live instalovali pomocí instalátoru balíčků (`apt install`) a ten není tak často aktualizovaný. Repozitář změníme příkazem:

```
tlmgr option repository ftp://ftp.tu-chemnitz.de/pub/tug/historic/systems/  
texlive/2021/tlnet-final/
```

Odkazy na staré archívy systému TeX Live, jako například v našem případě TeX Live 2021 najdeme na tomto odkazu: <https://www.tug.org/historic/>.

Následně instalujeme balíčky stejně jako v operačním systému Windows, přes příkazovou řádku příkazem `tlmgr install <balíček>`.



Obrázek 6.3: TeX Live Manager - instalace balíčku

6.7 Sazba obrázků

Pro importování a práci s grafikou je dostupný balíček `graphics` nebo jeho rozšíření `graphicx`. My použijeme `graphicx`, který nabízí modernější způsob úpravy obrázků. Abychom mohli využít tento balík, je nutné ho nejprve naimportovat. To provedeme přidáním příkazu `\usepackage[driver]{graphicx}` do hlavičky dokumentu. Parametr `driver`, neboli ovladač, nám pomáhá vysázet obrázky podle našeho výstupního formátu. Pokud tento parametr nezadáme, pak se vybere ovladač podle operačního systému. To je lepší pro kompatibilitu mezi různými operačními systémy.

Základní příkaz pro importování obrázků je `\includegraphics[klíč=hodnota,...]{soubor}`. Existují dva typy klíčů, s numerickou hodnotou a s hodnotou `true` nebo `false`. Klíč je nepovinný argument. Některé důležité klíče a hodnoty:

`scale` = číslo; škálování obrázku zadaným koeficientem

`width` = délka; šířka obrázku, pokud není zadán parametr `height` je obrázek škálován stejně jako je zadána šířka, příkazem `\linewidth` změníme délku obrázku na délku stránky

`height` = délka; výška obrázku, pokud není zadán parametr `width` je obrázek škálován stejně jako je zadána výška

`angle` = číslo; úhel kterým má být obrázek rotován proti směru hodinových ručiček, ve stupních; výška nebo šířka zadaná před úhlem je také rotována, to znamená že z výšky se stane šířka a šířka se stane výškou (kladný úhel) nebo hloubkou (záporný úhel) [2]

Pokud chceme obrázek vycentrovat vložíme ho mezi příkazy `\begin{center}` a `\end{center}`. Pro umožnění obtékání textu kolem obrázku umístíme obrázek mezi příkazy `\begin{wrapfigure}`.

6.7.1 Plovoucí prostředí figure

Figure je plovoucí prostředí sloužící pro sazbu obrázků. Pokud se obrázek nevejde na současnou stránku, počká se s jejich sazbou až do nalezení dalšího vhodného místa. Prostředí můžeme spouštět s přídatnými specifikátorem pozice, který ovlivňuje jejich vlastnosti:

```
\begin{figure}[pozice]
...
\end{figure}
```

Do `pozice` můžeme zadat jedno nebo více písmen, které odpovídají následujícím vlastnostem:

`h` objekt bude vysázen do místa, kde je zdrojový text

`t` objekt bude vysázen do horní části stránky

`b` objekt bude vysázen do spodní části stránky

p objekt bude na samostatné stránce, která obsahuje jen plovoucí objekty [9]

Jednotlivé specifikátory pozice můžeme kombinovat pro docílení požadovaných výsledků. Pokud není uvedena žádná pozice je standardně použita kombinace `tbp`.

Pokud jsou použity příkazy `\clearpage`, `\cleardoublepage` nebo `\end{document}` vysází se všechny zatím nevysázené objekty na samostatnou stránku.

Uvnitř plovoucího prostředí můžeme pomocí příkazu `\caption{text}` vysázet pod objekt popisek.

6.8 Sazba tabulek

Tabulky lze rozdělit na otevřené a uzavřené. Otevřené tabulky nejsou ohraničeny svislými linkami, ale pouze linkou v záhlaví. Občas se vyskytují s uzavírací linkou v zápatí tabulky. Jsou vhodné pro použití na celou šířku stránky. Uzavřené tabulky mají ohraničení ze všech stran, nemusí zabírat celou šířku stránky a zpravidla se používají v technických textech a knihách.

Pro sazbu tabulek je k dispozici prostředí `tabular`. Existují dvě varianty prostředí - `tabular` a `tabular*`. V základní variantě šířka tabulky závisí na obsahu jednotlivých sloupců. Varianta s hvězdičkou umožňuje vytvořit tabulku s šířkou zadanou parametrem.

```
\begin{tabular}[p]{sloupcce}
  řádek\\
\end{tabular}
```

```
\begin{tabular*}[p]{šířka}{sloupcce}
  řádek\\
\end{tabular}
```

Písmeno `p` je volitelný parametr a určuje způsob připojení tabulky k okolnímu textu:

`t` - připojení horním okrajem (top)

`b` - připojení dolním okrajem (bottom)

Pokud není žádný parametr uveden připojí se tabulka středem.

V parametru `sloupcce` určujeme zarovnání textu v jednotlivých sloupcích, počet sloupců a svislých čar mezi nimi.

`l` - obsah sloupce je zarovnán vlevo (left)

`r` - obsah sloupce je zarovnán vpravo (right)

`c` - obsah sloupce je zarovnán na střed (center)

`p{šířka}` - obsah je sazen do bloku o šířce, která je zadaná v závorkách

| - Vysází vertikální ohraničení

|| - Vysází dvojité vertikální ohraničení

Příklad hlavičky:

```
\begin{tabular}{|lcr|p{4cm}|}
```

Vysází tabulku, která má čtyři sloupce. Před prvním, před čtvrtým a za čtvrtým sloupcem budou sázeny svislé čáry. Obsah prvního sloupce bude zarovnán vlevo, druhý na střed, třetí vpravo a poslední sloupec bude zarovnán do bloku šířky 4 cm.

```
| vlevo   na střed  vpravo | blok  
| tady    bude     další  | řádek
```

Parametr `řádek` definuje jednotlivé řádky tabulky, kde každý řádek je ukončen `\\`. Tyto řádky se skládají ze sloupců oddělených znakem `&`. Každý řádek musí obsahovat stejný počet sloupců jako je definováno v parametru `sloupec`. Vodorovné čáry se na konci řádku sází příkazem `\hline`. Pokud nechceme čáru přes celou šířku tabulky, použijeme příkaz `\cline{x-y}`, kde `x` je pořadové číslo prvního sloupce a `y` je číslo posledního sloupce přes který má čára vést [1].

Často v tabulkách potřebujeme sloučit sloupce dohromady. Toho můžeme docílit za pomoci příkazu `\multicolumn{počet}{zarovnání}{text}`, kde `počet` udává přes kolik sloupců chceme umístit `text` s zadaným zarovnáním. Příklad toho co jsme si zatím popsali:

```
\begin{tabular}{|l|c|c|}\hline  
&\multicolumn{2}{c|}{Max. teplota v daném dni}\\ \cline{2-3}  
Město&Datum&Max. teplota\\ \hline  
Praha&28.8&30\textcelsius\\  
Pardubice&30.8&28\textcelsius\\  
Ostrava&3.9&25\textcelsius\\ \hline  
\end{tabular}
```

Město	Max. teplota v daném dni	
	Datum	Max. teplota
Praha	28.8	30°C
Pardubice	30.8	28°C
Ostrava	3.9	25°C

Pokud potřebujeme vysázet velkou tabulku, která se nevejde na stránku, můžeme ji otočit o 90°. K tomu využijeme balíček `rotating`, který je součástí standardního LaTeX balíčku `graphics`. K otočení tabulky stačí namísto `\begin{table}` použít příkaz `\begin{sidewaystable}`. Pro obrázky existuje příkaz `\begin{sidewaysfigure}`.

6.8.1 Plovoucí prostředí table

Podobně jako je u obrázků plovoucí prostředí `figure` existuje také pro tabulky plovoucí prostředí `table`. Jelikož dopředu nevíme kde dojde k rozdělení textu na novou stránku a nevíme tak jestli zde bude dostatečné místo pro tabulku, proto plovoucí prostředí si uloží tabulku a vysází ji později na vhodné místo v dokumentu. Pro použití plovoucího prostředí `table` použijeme příkaz:

```
\begin{table}[pozice] tabulka \end{table}
```

, kde parametr `pozice` udává umístění tabulky. Skládá se z jednoho až čtyř písmen stejně jako tomu je u `figure` (kapitola 6.7.1).

6.9 Sazba matematických vztahů

Pro sazbu matematických vztahů v LaTeXu máme k dispozici několik možností. Pro sazbu v rámci řádku využíváme znak `$` následovně:

```
$ text $
```

Pro sazbu na nový řádek:

```
$$ text $$
```

Pro sázení číslovaných rovnic využijeme balíček `amsmath`. Pokud tento balíček nemáme nainstalovaný, provedeme jeho instalaci podle návodu v kapitole 6.6. Balíček následně nainportujeme příkazem `\usepackage{amsmath}`. Balíček nabízí několik prostřední pro psaní matematických vztahů. My si představíme prostředí `equation` a `align`.

Prostředí `align` použijeme pro sazbu dvou a více rovnic, znakem `&` určujeme podle čeho se rovnice zarovnají, nejčastěji tedy podle rovnítka a nový řádek vytváříme příkazem `\\`.

```
\begin{align}
  2x - 5y &= 8 \\
  3x + 9y &= -12 \\
\end{align}
```

$$2x - 5y = 8 \tag{6.1}$$

$$3x + 9y = -12 \tag{6.2}$$

Prostředí `equation` se využívá pro sazbu jednořádkových rovnic s automatickým číslováním. Pro nečíslované rovnice použijeme `equation*`. Další rovnice lze rozdělit prostředím `split` uvnitř prostředí `equation`. Nové řádky značíme `\\`.

```

\begin{equation}
\begin{split}
a &= b + c - d \\
&+ e + f
\end{split}
\end{equation}

```

$$\begin{aligned}
 a &= b + c - d \\
 &+ e + f
 \end{aligned}
 \tag{6.3}$$

6.9.1 Prvky matematických výrazů

Nyní si představíme jak sázet matematické prvky.

Indexy a exponenty

Indexy tvoříme podtržítkem (`_`), exponenty znakem stříška (`^`). Parametrem je text indexu (exponentu). Kombinací je možno vytvářet i více úrovní indexů a exponentů.

```

x_{i+1}   x_{i+1}
x^{n+1}   x^{n+1}
y^{x^1}   y^{x^1}
y_i^n     x_i^n

```

Zlomky

Sazbu vodorovné zlomkové čáry provádíme příkazem `\frac`, který má dva povinné parametry: čí-
tatel a jmenovatele.

```

\frac{x+1}{x-1}

```

$$\frac{x+1}{x-1}$$

```

\frac{x+1}{x\frac{y}{z-1}}

```

$$\frac{x+1}{x\frac{y}{z-1}}$$

Odmocniny

Odmocniny sázíme příkazem `\sqrt` s jedním povinným parametrem (argument odmocniny) a jed-
ním volitelným parametrem (kolikátá odmocnina).

```

c=\sqrt{a^2+b^2}   c = \sqrt{a^2 + b^2}

```

```

\sqrt[n]{a+b}     \sqrt[n]{a+b}

```

Sumy a integrály

Sumy a integrály vytváříme pomocí příkazů `\sum` a `\int`. Sumy a integrály často obsahují intervaly, které sázíme pomocí znaků `^` a `_`. Pokud chceme mít limity integrálu pod znakem a nad znakem přidáme za příkaz pro integrál příkaz `\limits`.

$$\int_0^1 f(x) dx$$

$$\int_{x=0}^{y=1} f(x) dx$$

$$\sum_{i=0}^n x_i$$

6.10 Sazba výpisů kódu

Pro vkládání zdrojových kódů do dokumentu můžeme využít prostředí `verbatim`, které ignoruje LaTeX příkazy a sází text tak, jak byl napsán ve zdrojovém prostředí a nebo více flexibilní balík `listings`, který si popíšeme.

Abychom mohli začít pracovat s tímto balíčkem je potřeba ho nejprve nainstalovat podle návodu v kapitole 6.6. Do dokumentu ho naimportujeme příkazem `\usepackage{listings}`. Oproti prostředí `verbatim` nabízí zvýrazňování syntaxe programovacích jazyků a zobrazuje odsazení textu. Seznam podporovaných jazyků najdeme v dokumentaci [10]. Výpisy balíku `listings` nepodporují české znaky, ty tak musíme ručně nadefinovat a do příkazu `\lstset{}` vložit:

```
literate={á}{\{a}}1 {Á}{\{A}}1 {ä}{\{"a}}1 {Ä}{\{"A}}1
1 {č}{\v{c}}1 {Č}{\v{C}}1 {d}{\v{d}}1 {Ď}{\v{D}}1
1 {ě}{\v{e}}1 {Ě}{\v{E}}1 {é}{\{e}}1 {É}{\{E}}1
1 {í}{\{i}}1 {Í}{\{I}}1 {í}{\{l}}1 {Ĺ}{\{L}}1
1 {l}{\v{l}}1 {L}{\v{L}}1 {ň}{\v{n}}1 {Ň}{\v{N}}1
1 {ó}{\{o}}1 {Ó}{\{O}}1 {ô}{\^{o}}1 {Ô}{\^{O}}1
1 {ř}{\v{r}}1 {Ř}{\v{R}}1 {ř}{\{r}}1 {Ř}{\{R}}1
1 {š}{\v{s}}1 {Š}{\v{S}}1 {t}{\v{t}}1 {T}{\v{T}}1
1 {ú}{\{u}}1 {Ú}{\{U}}1 {ů}{\r{u}}1 {Ů}{\r{U}}1
1 {ý}{\{y}}1 {Ý}{\{Y}}1 {ž}{\v{z}}1 {Ž}{\v{Z}}1
```

Pro vysázení části kódu v řádku využijeme příkaz `\lstinline[parametry]|kód|`. Znaky `|` můžeme nahradit libovolnými znaky, které se nenachází v kódu. Jako parametr můžeme například vybrat jazyk pro syntaxi pomocí `language=jazyk` nebo výběr barev pro zvýraznění komentářů a klíčových slov.

Příklad:

```
\lstinline[language=Python]|for x in array| nám dá:  
for x in array.
```

Pokud chceme sázet bloky kódu použijeme prostředí `lstlisting`.

Příklad:

```
\begin{lstlisting}[language=Python]  
    fruits = ["apple", "banana", "cherry"]  
    for x in fruits:  
        print(x)  
\end{lstlisting}
```

```
fruits = ["apple", "banana", "cherry"]  
for x in fruits:  
    print(x)
```

Balíček `listings` si můžeme také nakonfigurovat a to příkazem `\lstset{parametry}`. Můžeme tak změnit například barvu zvýraznění syntaxe, vzhled a barvu rámečků kolem výpisu, zapnout nebo vypnout číslování řádků. Příkaz `\lstset{}` můžeme uvést jednou, nejlépe v záhlaví dokumentu, nebo před každým výpisem a definovat tak styl pro následující výpis kódu.

```
\lstset{  
    keywordstyle=\color{blue},  
    stringstyle=\color[RGB]{163,21,21},  
    commentstyle=\color{gray},  
    frame=lines  
}
```

Konfigurace výše nám dá tento výsledek:

```
#Inicializace pole  
arr = [1, 2, 3, 4, 5]  
sum = 0  
for i in range(0, len(arr)):  
    sum = sum + arr[i]  
print("Součet všech prvků pole: " + str(sum))
```

6.11 Záhloví a zápatí dokumentu

Záhloví a zápatí většinou obsahuje název dokumentu, číslo nebo název kapitoly a číslo stránky. K dispozici jsou dva příkazy: `\pagestyle{styl}`, který aplikuje vybraný styl na současnou a následující stránku a příkaz `\thispagestyle{styl}`, který ovlivní styl pouze současné stránky. Na výběr máme z následujících stylů:

`empty` Záhloví i zápatí bude prázdné.

`plain` Záhloví je prázdné, zápatí obsahuje uprostřed číslo stránky.

`headings` Zápatí je prázdné, záhloví obsahuje název sekce a číslo stránky vpravo.

`myheadings` Číslo stránky se nachází vpravo nahoře, zbytek záhloví si můžeme upravit

Jednotlivé třídy dokumentu mají nastavený automaticky výchozí styl. Třídy `article` a `report` mají styl `plain`. Třída `book` používá `headings`.

6.11.1 Balík fancyhdr

Úprava záhloví a zápatí je tak celkem omezená. Proto použijeme balík `fancyhdr` pro snazší práci a větší možnost úprav záhloví i zápatí. Do dokumentu vložíme balík příkazem `\usepackage{fancyhdr}`. Styl aplikujeme příkazem `\pagestyle{fancy}`.

V jednostranném dokumentu záhloví upravujeme příkazy: `\lhead{}`, `\chead{}`, `\rhead{}` a to podle umístění doleva, na střed nebo doprava. Zápatí se upravuje příkazy: `\lfoot{}`, `\cfoot{}`, `\rfoot{}`. Nadefinované záhloví a zápatí pak může vypadat následovně:

```
\lhead{}
\chead{}
\rhead{\bfseries Výsledky analýzy subjektů}
\lfoot{\today}
\cfoot{Petr Kohout}
\rfoot{\thepage}
```

Příkazem `\thepage` vysázíme číslo stránky. Příkazem `\today` vysázíme dnešní datum.

Výsledky analýzy subjektů

16. března 2023

Petr Kohout

2

Obrázek 6.4: Záhloví a zápatí jednostranného dokumentu

Ve dvoustranném dokumentu sázíme zápatí a záhlaví příkazy `\fancyhead[volby]{}` a `\fancyfoot[volby]{}`. V poli `volby` udáváme pozici prvků: E sudá stránka, O lichá stránka, R vpravo, L vlevo a C uprostřed. Takže například kombinací `[R0,LE]` bude text vysázen vpravo na liché stránce a vlevo na sudé stránce. [9]

```
\fancyhead[R0,LE]{\bfseries Výsledky analýzy subjektů}
\fancyfoot[LE,R0]{\thepage}
\fancyfoot[L0,CE]{\today}
\fancyfoot[CO,RE]{Petr Kohout}
```

Výsledky analýzy subjektů

2 16. března 2023 Petr Kohout

Obrázek 6.5: Záhlaví a zápatí na sudé stránce - dvoustranný dokument

Výsledky analýzy subjektů

16. března 2023 Petr Kohout 3

Obrázek 6.6: Záhlaví a zápatí na liché stránce - dvoustranný dokument

6.12 Tvorba křížových a hypertextových odkazů

V dokumentu často potřebujeme čtenáře odkázat na určitou sekci, stránku, rovnici, obrázek nebo tabulku. V době psaní dokumentu neznáme, kde bude jaký text a jak budou očíslovány obrázky nebo tabulky. K vytvoření značky použijeme příkaz `\label{název-značky}`. Tímto se vytvoří značka na kterou se později můžeme v textu odkazovat. Stránku na které se tato značka nachází můžeme vysázet příkazem `\pageref{značka}`.

Pokud je příkaz `\label{název-značky}` napsán po příkazu pro sekci, uvnitř prostředí rovnice nebo v příkazu `\caption{}`, který je uvnitř obrázku nebo tabulky. Pak příkaz `\ref{značka}` vysází číslo sekce, rovnice, obrázku nebo tabulky. [1]

Příkaz `\eqref{název-značky}` je součástí balíku `amsmath` a funguje stejně jako `\ref` s tím rozdílem, že sází čísla rovnic v závorkách stejně tak jako jsou uvedeny vedle rovnice.

Bibliografická citace slouží k jednoznačnému označení publikovaného díla. Odkaz na tuto citaci v textu vytvoříme příkazem `\cite{název-citace}`. Seznam citací můžeme napsat přímo do dokumentu nebo použít bibliografickou databázi. Pro psaní bibliografie přímo do dokumentu použijeme prostředí `thebibliography`. Parametr prostředí udává nejdelší název citace v našem seznamu. Po-

ložky v seznamu začínají příkazem `\bibitem`, jehož povinným parametrem je název citace na který se budeme odkazovat příkazem `\cite`.

```
\begin{thebibliography}{9}
  \bibitem{rybicka03}
    Jiří Rybička (2003) \emph{LaTeX pro začátečníky}, Konvoj.
\end{thebibliography}
```

Odkaz na tuto literaturu vytvoříme příkazem `\cite{rybicka03}`. Odkaz vznikne ve formě pořadového čísla literatury.

6.12.1 Hypertextové odkazy

Čísla literatury, které nám vznikly příkazem `\cite` nejsou odkaz a tak nás nepřesměruje na seznam literatury. Aby číslo fungovalo jako odkaz musí použít balík `hyperref`. Tento má nejlepší uplatnění zejména ve spojení s formátem PDF a tedy s překladačem `pdfLaTeX`. Balík konfiguruje příkazem `\hypersetup{}` například takto:

```
\hypersetup{
  pdftitle=První dokument,
  colorlinks
  urlcolor=cyan,
  filecolor=magenta
  citecolor=blue
  urlbordercolor={.5 .4 .8}
}
```

Kde příkazem `pdftitle` nastavujeme nadpis PDF souboru, `colorlinks` nastavujeme barevně zvýrazněné odkazy, pokud nevedeme, zvýraznění bude pomocí rámečků. `urlcolor` nastavuje barvu URL odkazu, `filecolor` barvu odkazu na soubor, `citecolor` mění barvu citací. Při použití rámečků měníme jejich barvy příkazy `urlbordercolor`, `filebordercolor`, `citebordercolor` a jejich barvu zadáváme v modelu RGB.

Hypertextové odkazy vkládáme příkazy `\url` a `\href`. Příkaz `\url` vkládá celé adresy zatímco `\href` vysází zvolený text, který nás po kliknutí přesměruje na danou adresu.

Příklad:

```
\url{https://www.vsb.cz/cs/}      https://www.vsb.cz/cs/
\href{https://www.vsb.cz/cs/}{VŠB}  VŠB
```

6.13 Tvorba rejstříku a obsahu

Rejstřík je abecední seznam slov a výrazů s čísly stránek na kterých se tyto výrazy nachází. LaTeX automaticky rejstřík nevytváří, dokáže připravit vstupní data, které balík MakeIndex zpracuje a vygeneruje rejstřík. Balík připojíme příkazem `\usepackage{makeidx}`. Tento rejstřík pak LaTeX dokáže zobrazit.

Pro přidání slov do rejstříku použijeme příkaz `\index`, jehož povinným argumentem je text, který chceme zahrnout do rejstříku. Tento příkaz nemá žádný efekt dokud v hlavičce nepoužijeme příkaz `\makeindex`. Ten vytvoří soubor s příponou `.idx`, který obsahuje slova, které jsme uvedli v příkazech a to ve tvaru:

```
\indexentry{heslo}{stránka}
```

Slova jsou seřazena podle abecedy a každé slovo je za příkazem `\item`, `\subitem` nebo `\subsubitem` podle jejich úrovně. Tyto úrovně definujeme při použití `\index` a to tak že jednotlivá slova oddělíme znakem `!`:

```
\index{slovo!slovo2!slovo3}
```

Položky v rejstříku se mohou odkazovat na jiné položky v rámci rejstříku. K tomu se používá `|see`. V rejstříku se místo čísla stránky vysází viz. odkazované slovo. Soubor s příponou `.idx` zpracujeme zadáním příkazu `makeindex dokument.idx` do příkazové řádky. Nebo v prostředí editoru TeXstudio: Nástroje → Rejstřík. V dokumentu rejstřík vysázíme příkazem `\printindex` a dokument znovu přeložíme.

6.13.1 Obsah

Obsah dokumentu se vytváří automaticky podle příkazů pro titulky. V místě, kde chceme obsah vysázet použijeme příkaz `\tableofcontents`. Na tomto místě se vysází nadpis “Obsah” s názvy jednotlivých titulků a čísla stránek. Aby se obsah zobrazil je nutné dokument přeložit dvakrát. Při prvním překladu se vytvoří soubor `.toc`, který obsahuje potřebné informace. Při druhém překladu se tento soubor přečte a vysází se podle něj obsah. Stejně tak můžeme vysázet seznam obrázků příkazem `\listoffigures` a seznam tabulek `\listoftables`.

Číslované titulky jsou do obsahu přidávány automaticky. Máme však možnost nastavit do jaké úrovně mají být tyto titulky zařazeny do obsahu. Příkazem `\setcounter{tocdepth}{3}`, nastavíme že se v obsahu zobrazí `\chapter`, `\section`, `\subsection`, `\subsubsection`. Nečíslované titulky se v obsahu nezobrazují, máme však možnost do obsahu přidat a to příkazem `\addcontentsline{soubor}{uroven}{text}`, kde soubor je obsahový soubor: `toc` pro obsah, `lof` pro seznam obrázků a `lot` pro seznam tabulek. Úroveň je se nastavuje stejně jako titulek příslušné úrovně. Příkaz `\addcontentsline` se většinou využívá pro zavedení nečíslovaných titulků do obsahu [1]:

```
\section*{Úvod}
```

```
\addcontentsline{toc}{section}{Úvod}
```

Příkaz `\addtocontents{soubor}{text}` se používá pro vložení libovolného textu do obsahu bez uvedení stránky. Například:

```
\section*{Úvod}
\addcontentsline{toc}{section}{Úvod}
\addtocontents{toc}{\textit{Úvodní slovo}}
```

6.14 Sazba seznamu použité literatury

V návodu na křížové odkazy (6.12), bylo popsáno jak vytvářet seznam literatury pomocí prostředí `thebibliography`. Většina autorů se ve svých pracích neustále odkazuje na stejné literatury a tak musí kopírovat prostředí `thebibliography` do svých nových dokumentů. Existuje tak lepší způsob jak vytvářet seznam literatury. Program BibTeX, který automaticky vytvoří seznam literatury podle zadaných informací o publikaci v souboru `.bib`. Ten následně přidáme do dokumentu příkazem:

```
\bibliography{bib soubor}
```

Na místě tohoto příkazu se také vysází seznam literatury. Do souboru s literaturou můžeme nové záznamy přidávat ručně. Nespočet webů také nabízí vygenerování literatury ve formátu vhodného přímo pro BibTeX, jako je například web Citace PRO. Příklad souboru `.bib`:

```
@book{Rybicka03,
  author = {Jiří Rybička},
  title = {LaTeX pro začátečníky},
  publisher = {Konvoj},
  address = {Brno},
  year = {2003},
  edition = {3. vyd},
  isbn = {80-7302-049-1},
}
```

V dokumentu se pak na citaci odkážeme příkazem `\cite{Rybicka03}`. Aby se citace vysázela je nutné nejprve dokument přeložit, poté přeložit pomocí programu BibTeX příkazem `bibtex dokument.tex` a následně znovu přeložit.

Styl citací upravujeme příkazem `\bibliographystyle{styl}`. Tímto říkáme jaký soubor se stylem (`.bst`) má BibTeX použít. Výchozí styly jsou:

- `plain` Seznam literatury je seřazen abecedně. Citace je označena číslem v hranatých závorkách.
- `unsrt` Seznam literatury je seřazen podle jejich citací v dokumentu. Citace je označena stejně jako u stylu `plain`

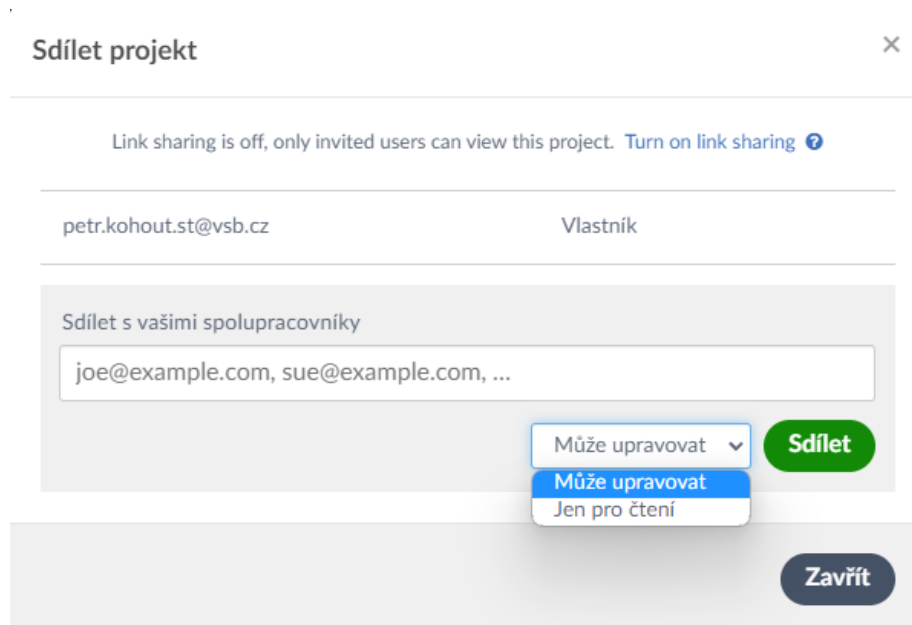
alpha Řazení literatury je stejné jako u **plain** ale jednotlivé publikace jsou označeny zkratkou autora jména plus rok vydání. Například Rybička 2003 se vysází jako [Ryb03]

abbrv Řazení a značení je stejné jako **plain**. Jména autorů jsou zkrácena.

6.15 Nástroje pro online sazbu a sdílení dokumentů

Online sazba dokumentů ulehčuje autorovi vytváření a kompilaci LaTeX dokumentů. Autor se například nemusí zabývat stahováním balíčků. Představíme si jeden z neznámějších a nejpoužívanějších online editorů - Overleaf a také webovou platformu Cocalc.

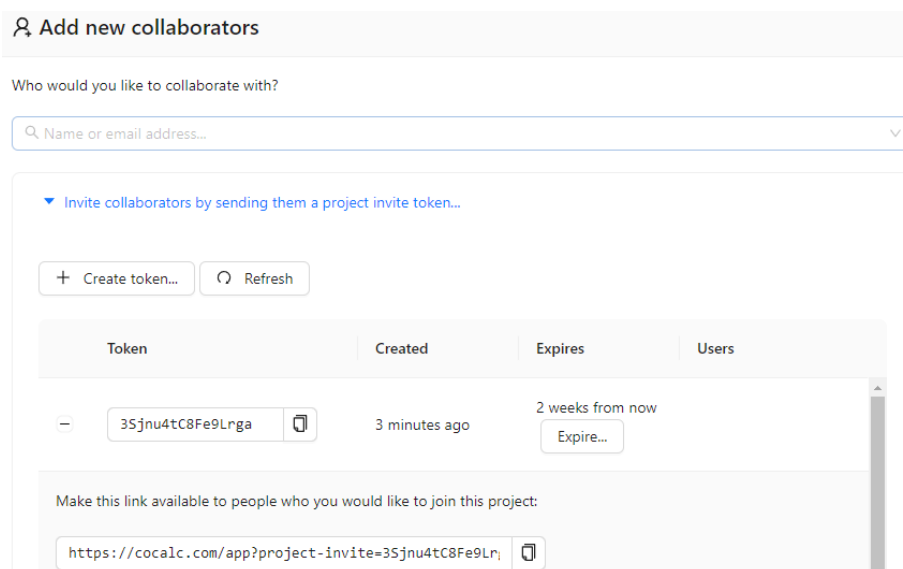
Pro vytvoření dokumentu je zapotřebí se nejprve zaregistrovat. Následně si vytvoříme projekt. Dokument píšeme stejně jako jsme si předvedli v předchozích návodech. Overleaf obsahuje všechny LaTeX balíčky, není tedy potřeba jednotlivé balíčky doinstalovávat, stačí je pouze přidat do dokumentu příkazem `\usepackage{balíček}`. V horní liště editoru máme tlačítko "Sdílet". To slouží ke sdílení projektu mezi spolupracovníky. Zde zadáme emailovou adresu spolupracovníka a určíme jeho práva, zda může číst nebo i upravovat dokument. Ten poté obdrží email s odkazem na tento projekt. V neplacené verzi je však omezen počet spolupracovníků na 1.



Obrázek 6.7: Overleaf - sdílení projektu

Další dostupný online editor je Cocalc. Ten nenabízí pouze LaTeX editor ale také například Jupyter notebook, Octave nebo Linux terminál. Stejně jako Overleaf nabízí možnost kolaborativní práce na projektu. Spolupracovníky přidáme manuálně pomocí jejich Cocalc účtu nebo můžeme vytvořit token, který rozešleme ve formě odkazu. Projekt pak může upravovat i nezaregistrovaný uživatel.

Pro vytvoření tohoto tokenu přejdeme do "Settings" a zde v sekci "Add new collaborators" token vytvoříme vybráním možnosti "Create token...".



Obrázek 6.8: Cocalc - sdílení projektu

Kapitola 7

Závěr

Cílem bakalářské práce bylo vytvořit soubor videotutoriálů pro práci s typografickým systémem LaTeX. V první části této práce jsou porovnány jednotlivé instalace systému LaTeX a to systém ConTeXt, LuaTeX a LyX.

Následuje porovnání vybraných LaTeX editorů Texmaker, TeXstudio a online kolaborativního editoru Overleaf. Z těchto je vybrán editor TeXstudio, který se následně používá v jednotlivých video návodech. Jeden video návod se pak zabývá instalací vybraného editoru. V další části je detailně popsána instalace systému TeX Live na platformách Windows a Linux Mint. K tomuto účelu je také vytvořen video návod.

Nejobsáhlejší část této práce je tvorba LaTeX dokumentů doprovázena jednotlivými video návody. Tyto návody popisují tvorbu dokumentu od základů. V prvním návodu je popsána sazba jednoduché dokumentu obsahujícího pouze jednu větu. Následuje návod na kompilaci tohoto dokumentu jak přes rozhraní editoru, tak pomocí příkazové řádky. Poté je vyřešena správná sazba češtiny. Dále jsou představeny jednotlivé stupně a řezy písma. Nechybí také video návod na instalaci doplňkových balíčků, které jsou využity v dalších návodech. Poté je popsána a předvedena sazba obrázků a tabulek společně s plovoucími prostředími figure a table.

LaTeX se často používá pro sazbu matematických textů a tak je v dalším návodu představena sazba matematických vztahů pomocí balíčku amsmath. Následují návody pro sazbu výpisů kódu, sazbu záhlaví a zápatí, tvorbu křížových a hypertextových odkazů. Nedílnou součástí každého dokumentu je také obsah. Tímto se zabývá další návod a to společně s tvorbou rejstříků. Další důvodem oblíbenosti LaTeXu je také jednoduchá sazba seznamu použité literatury, což je tématem dalšího video návodu. V posledním návodu jsou představeny dva nástroje pro online sazbu a sdílení dokumentů. Těmito nástroji jsou Overleaf a Cocalc.

Vzniklo tedy celkem 20 video návodů s délkou od minuty až po návod 11 minutový. Tyto návody by mohli být využity pro výuku nebo jako podpůrné materiály pro studenty.

Literatura

1. RYBIČKA, Jiří. *LaTeX pro začátečníky*. 3. vyd. Brno: Konvoj, 2003. ISBN 80-7302-049-1.
2. PATRICK W. DALY, Helmut Kopka a. *LaTeX: podrobný průvodce*. Vyd. 1. Brno: Computer Press, 2004. ISBN 80-722-6973-9.
3. *ConTeXt Wiki* [online]. 2022. [cit. 2023-01-17]. Dostupné z: <https://wiki.contextgarden.net/Installation>.
4. *What is LyX?* [online]. [cit. 2023-01-09]. Dostupné z: <https://www.lyx.org/WhatIsLyX>.
5. *TeX Live: TeX Users Group* [online]. 2022. [cit. 2023-01-23]. Dostupné z: <https://www.tug.org/texlive/>.
6. *Windows - TeX Live* [online]. 2022. [cit. 2023-01-27]. Dostupné z: <https://www.tug.org/texlive/windows.html>.
7. *TeXstudio: A LaTeX editor* [online]. [cit. 2023-01-30]. Dostupné z: <https://www.texstudio.org/>.
8. *CTAN: Comprehensive TEX Archive Network* [online]. 1993. [cit. 2023-02-01]. Dostupné z: <https://www.ctan.org/>.
9. PTÁČEK, Martin. *Elektronické publikování* [online]. 2005. [cit. 2023-02-03]. Dostupné z: <https://home1.vsb.cz/~s1a10/educ/EPubl/latex-docbook/ch06s01.html>.
10. HOFFMANN, Jobst. *The Listings Package* [online]. [cit. 2023-02-15]. Dostupné z: <https://mirrors.nic.cz/tex-archive/macros/latex/contrib/listings/listings.pdf>.