

Česká zemědělská univerzita v Praze
Provozně ekonomická fakulta



Využití sázecího systému $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ pro odborný časopis

Diplomová práce

Vedoucí práce:
Doc. Ing. Vojtěch Merunka, Ph.D.

Autor:
Bc. Jan Kropáč

copyright 2009

Čestné prohlášení

Prohlašuji, že jsem práci Využití sázečního systému \LaTeX pro odborný časopis vypracoval samostatně a použil jsem výhradně zdroje uvedené v seznamu použité literatury.

Praha 10.4.2009

.....

Poděkování

Děkuji vedoucímu diplomové práce Doc. Ing. Vojtěchu Merunkovi, Ph.D. za odborné vedení a za cenné rady, které mi pomohly při psaní této práce.

Dále cítím povinnost poděkovat autorům $\text{T}_{\text{E}}\text{X}$ u a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u, pánům Donaldu E. Knuthovi a Leslie Lamportovi, protože bez jejich skvělých produktů bych neměl o čem psát a neměl bych ani nástroje pro psaní práce na jiné téma.

Využití sazecího systému \LaTeX pro odborný časopis

Tato diplomová práce se zabývá možnostmi profesionální sazby textu v prostředí operačního systému Microsoft WindowsTM. V úvodních kapitolách krátce popisuje vznik a vývoj typografie do její současné podoby. Dále specifikuje základní typografická pravidla nutná ke správnému používání typografických programů. Praktická část této práce definuje programovou publikační platformu sestavenou z volně dostupných programů. Všechny použité programy spolupracují s balíkem maker \LaTeX , který zajišťuje výslednou sazbu pomocí typografického programu \TeX . Je vyvinuta nová třída dokumentů pro \LaTeX , která usnadňuje sazbu článků pro odborný časopis. K této třídě je dále vytvořena definice rozvržení pro program LyX. V závěru práce je uvedeno kritické zhodnocení možností navržené publikační platformy, třídy dokumentu pro \LaTeX a definice rozvržení pro LyX.

Klíčová slova: typografie, sazba, \TeX , \LaTeX , $\text{MiK}\TeX$, TeXnicCenter , LyX

Typesetting system \LaTeX used in professional journal

The main goal of this master thesis is a description of typesetting possibilities on Microsoft WindowsTM operating system. At the beginning there is a short historical review of typesetting beginnings and its evolution till it's current state. The basic typographical terms and principles are specified, as it is essential for understanding and proper use of typesetting programs. Practical part of this thesis is focused on defining program platform for computer publishing, that is based on free software. All used programs collaborate together with the package of typesetting macros \LaTeX , which uses \TeX software for the final output. New \LaTeX document class is being developed in order to simplify professional journal article typesetting. For this new class There is also a new layout for LyX program being developed. The conclusion of this thesis contains the critical evaluation of proposed publication platform, of \LaTeX document class and LyX document layout.

Key words: typography, typesetting, \TeX , \LaTeX , $\text{MiK}\TeX$, TeXnicCenter , LyX

Obsah

1	Úvod	5
2	Cíl práce a metodika	7
3	Typografie a sazba	8
3.1	Typografie	8
3.1.1	Definice typografie	8
3.1.2	Vznik klasické typografie	8
3.2	Obecné základy typografie	9
3.2.1	Typometrické jednotky	9
3.2.2	Rozměry písma	9
3.2.3	Písmo	10
3.2.4	Odstavce	11
3.2.5	Spojovník, pomlčky a mínus	13
3.2.6	Uvozovky	13
3.2.7	Kerning, ligatury a prostrkávání	14
3.2.8	Zvýraznění textu	15
3.3	Publikování dokumentů pomocí počítače	15
3.3.1	Historie počítačové typografie	15
3.3.2	TEXa jeho varianty	16
3.3.3	L ^A T _E X	17
4	Základy sazby textu v makru L^AT_EX	18
4.1	Příkazy L ^A T _E Xu a jejich parametry	18
4.2	Základní struktura dokumentu	18
4.3	Třídy dokumentů	19
4.4	Sazba textu a odstavců	21
4.5	Písmo	22
4.5.1	Rodina písma	22
4.5.2	Velikost písma	22

4.5.3	Řez písma	23
4.6	Uvozovky a pomlčky	23
4.7	Vnitřní členění dokumentu	24
4.8	Křížové odkazy a poznámky pod čarou	24
4.9	Prostředí	25
4.9.1	Výčty	25
4.9.2	Tabulky	26
4.9.3	Matematické vzorce	27
4.9.4	Přímý výstup	28
4.10	Plovoucí objekty	28
4.11	Zvláštnosti českého jazyka	29
4.11.1	Nastavení prostředí	29
4.11.2	Dělení slov	30
4.11.3	Přetahování předložek	30
5	Návrh řešení na využití L^AT_EXu pro sazbu odborného časopisu	31
5.1	Definice požadavků	31
5.2	Softwarové vybavení pro publikování v L ^A T _E Xu	31
5.2.1	MiK _T E _X	31
5.2.2	L ^A T _E XDraw	32
5.2.3	T _E XnicCenter	33
5.2.4	LyX	34
5.3	Tvorba šablony pro L ^A T _E X	35
5.3.1	Identifikace	36
5.3.2	Iniciace	36
5.3.3	Deklarace parametrů	38
5.3.4	Vykonání parametrů	38
5.3.5	Nahrání děděné třídy	38
5.3.6	Hlavní blok kódu	38
5.4	Uložení třídy a konfigurace MiK _T E _X u	46

5.5	Použití vytvořené třídy v \LaTeX u	47
5.6	Tvorba šablony pro LyX	48
5.7	Zpřístupnění šablony v LyXu	52
6	Závěr	54
7	Seznam literatury	56
A	Výsledný dokument třídy <i>clanek</i>	57
B	Definice třídy dokumentu pro \LaTeX	59
C	Šablona pro LyX	64
D	Seznam obrázků	68
E	Seznam tabulek	69

1 Úvod

Jestliže se minulá století nazývala století páry a století elektřiny, lze bez nadsázky současné století nazvat stoletím informací či informačních technologií. Informační technologie se staly nedílnou součástí našeho každodenního života a málokdo by si dnes dokázal život bez nich představit. Internetová periodika pomalu vytlačují své papírové předchůdce a elektronická pošta už dávno nahradila listonoše. Ve všech těchto případech snižují informační technologie náklady na cestu informací k jejich příjemci a zkracují dobu cesty od autora ke čtenáři.

Mohlo by se zdát, že nastává konec typografických systémů, a že tvorba dokumentů pro stránková média postupně ztrácí smysl, ale není tomu tak. Informace totiž nemají jen svoji obsahovou část, ale mají také svoji formu. U elektronické pošty či aktuálních zpráv samozřejmě převažuje aktuálnost a informační hodnota nad vzhledem, ale například u knih tomu tak není. Vizuální část dokumentu má velký vliv na schopnosti čtenáře rozlišit důležité informace od méně podstatných a může mít i vliv na udržení pozornosti. Špatná volba řezu písma, nebo příliš husté řádkování znesnadňují orientaci v textu a u delších dokumentů můžou čtenáře odradit od jejich čtení.

Většina z nás se v práci i v osobním životě často setkává s nutností vytvořit textový dokument. Vysokoškolští studenti pravidelně píší seminární práce a zakončují své studium pracemi diplomovými a bakalářskými. V zaměstnání zase lidé často píšou obchodní dopisy či vytvářejí dokumentaci ke své práci. Tvorba textových dokumentů se stala nedílnou součástí našeho dnešního života.

Na první pohled by se mohlo zdát, že výpočetní technika nahradila klasickou sazbu a že o ní autoři dokumentů nemusí nic vědět. Lidé používají vyspělý kancelářský software, jehož každá nová verze přináší další a další zlepšení. Kontrola pravopisu pomáhá autorům vyvarovat se hrubých chyb a její pokročilejší verze již kontrolují i větnou stavbu. Slovníky synonym zase pomáhají hledat náhradu za často se opakující slova. Programové nástroje a hlavně jejich propagační materiály se snaží všechny zákazníky přesvědčit, že za ně vykonají většinu práce a že jejich výstup je kvalitní.

Je tomu ale tak? Je pravda, že počítačové programy všechno udělají za nás? Můžeme zapomenout na pravidla pravopisu a vše nechat na inteligenci svého textového editoru? Patří typografické příručky do polic antikvariátů a do soukromých sbírek původních sazečů?

Doba se sice změnila, ne všechny současné vymoženosti však beze zbytku plní očekávání svých uživatelů a marketingová hesla z propagačních materiálů. Kontrola pravopisu sice dokáže najít vyložené chyby a překlepy, ale nelze se na ni stoprocentně spolehnout. Jak by mohl počítačový program správně vyhodnotit například

shodu podmětu s přísudkem a určit správnou koncovku pro přísudek v čase minulém? Programátoři textových editorů se sice o řešení podobných problémů snaží, ale výsledky nejsou zatím dostatečně spolehlivé. I jedna hrubá chyba v obchodním dopise vrhá na pisatele špatné světlo, takže je zatím stále nutné znát pravidla pravopisu a nespolehat se jen a pouze na počítačové programy.

Podobným problémem, jako je automatická kontrola pravopisu, je způsob sazby. Možnosti počítačových programů se sice stále zlepšují, dokonalé kvality sazby však dosahují jen profesionální produkty s cenou v řádu desetitisíců korun. Běžně používaný kancelářský software takové kvality sazby zdaleka nedosahuje a jeho výstup je výrazně horší než výstup klasické ruční sazby. Rozdíl mezi kvalitní a nekvalitní sazbou z pohledu laického čtenáře není v ničem konkrétní, laik nepozná jednotlivé typografické chyby. Velký rozdíl ale je v celkovém pocitu z dokumentu. Kvalitně vysázený dokument totiž působí výrazně lépe, aniž by čtenář tušil proč.

Důvodů nedokonalosti textových editorů je několik, mezi jeden z hlavních patří i rozdílnost jednotlivých národních prostředí. Jak pravidla pravopisu, tak typografická pravidla jsou specifická pro konkrétní jazyk. Většina hlavních textových editorů vzniká v anglosaském světě a tudíž logicky podporuje primárně angličtinu. Zakomponování podpory pro další jazyk je nákladné a český trh není dostatečně velký, aby měl pro softwarové firmy prioritu. Většinou považují za dostatečné zabudovat do svého produktu základní kontrolu pravopisu na úrovni jednotlivých slov a tím veškerá podpora jazyka končí.

Typografie je věda, která existuje několik set let. Za dobu své existence se vyvinula a všechna její pravidla mají svá opodstatnění. Tato práce si klade za cíl popsat možnosti tvorby kvalitních dokumentů, které dodržují všechna typografická pravidla, a to vše pomocí programů dostupných zdarma. Nebude se tedy jednat o klasické textové editory, ale o specializované sazecí programy. Tyto programy část práce se sazbou dělají automaticky, ale část práce zůstává na autorech dokumentu. Práce s nimi není úplně intuitivní, ale po krátkém seznámení je pro středně zkušené uživatele počítačů možná. Výsledek rozhodně stojí za to.

2 Cíl práce a metodika

Cílem této práce je popsat a vysvětlit základy typografie, navrhnout programovou platformu pro počítačové publikování a vysvětlit její použití. Jako příklad počítačové publikace bude sloužit odborný časopis.

Pro správné pochopení popisované problematiky je potřeba nejprve zavést některé odborné termíny z oblasti typografie a pochopit její vývoj. Kapitola „Typografie a sazba“ tedy začíná vznikem a vývojem klasické sazby. Historie typografie je pro současnou sazbu důležitá, protože z ní vycházejí některé stále platné principy. V dalších částech této kapitoly jsou postupně popisována základní typografická pravidla. Závěr kapitoly tvoří část o počítačovém publikování dokumentů. Zde jsou popsány programy $\text{T}_{\text{E}}\text{X}$ a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, co se týče jejich funkce, a vazba mezi nimi.

V následující kapitole „Základy sazby textu v makru $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ “ je popsána práce s typografickým systémem $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Hlavní důraz je kladen na pochopení struktury vstupního dokumentu a použití nejčastějších příkazů a prostředí. Mezi popisovaná prostředí patří například výčty, tabulky, prostředí pro sazbu matematických vzorců a podobně. Jedna podkapitola je věnována práci s plovoucími objekty, které jsou typickým rysem sazby v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u. Opomenuty nezůstanou ani specifika sazby v českém jazyce.

Praktickou část práce zastupuje kapitola „Návrh řešení na využití $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u pro sazbu odborného časopisu“. Začátek této kapitoly je věnován definici zadání ukázkové počítačové publikace, tedy v tomto případě odborného časopisu. Další část je věnována popisu vybraného programového vybavení, které tvoří vlastní publikační platformu. Množina vybraných programů zahrnuje distribuci systému $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ určenou pro Microsoft WindowsTM, speciální vývojové prostředí pro editaci vstupního dokumentu, vektorový grafický editor pro přípravu grafických prvků a nakonec WYSIWYM¹ textový editor LyX pro zjednodušenou práci se vstupním souborem. V závěrečné části kapitoly je postupně podle definice požadavků vytvořen styl dokumentu pro $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ a dále definice šablony pro LyX.

V kapitole „Závěr“ je provedeno kritické zhodnocení dosažených výsledků práce. Kapitola obsahuje závěrečné shrnutí výhod a nevýhod popsané publikační platformy, možnosti jejího rozšíření a vylepšení.

¹What You See Is What You Mean

3 Typografie a sazba

3.1 Typografie

3.1.1 Definice typografie

Existuje mnoho více či méně rozsáhlých definic typografie, pro účely této práce je nejvhodnější následující ^[1] :

„Typografie je disciplína zabývající se písmem, především jeho správným výběrem, použitím a sazbou. Cílem typografie je zajistit čtenáři snazší čtení, efektivnější vnímání čteného textu a případně i vyloučit možné chyby a nejednoznačnosti plynoucí z více možných zápisů těžké věty.“

Tato práce se nebude primárně zabývat výběrem a použitím písma, ale bude se spíše věnovat sazbě písma a nástrojům, které ji zjednodušují.

3.1.2 Vznik klasické typografie

Vznik typografie je úzce svázán se znovuobjevením ² knihtisku roku 1444. Autorem tohoto postupu byl Johannes Guttenberg, který přišel s myšlenkou umístit jednotlivá písmena (lityry) do dřevěné sazebnice. Podobné postupy byly už v té době známy, ale pro tvorbu jednotlivých liter se zpravidla používalo dřevo, takže kvůli opotřebením nebylo možné lityry používat delší dobu. Guttenberg přišel s vlastní směsí cínu, olova a antimonu, kterou používal pro jejich odlévání.

Knihtisk rychle nahradil stávající metodu tisku (deskotisk), který překonal právě možností použít jednotlivé lityry i sazebnici vícenásobně. V dnešní době se už klasický knihtisk skoro nepoužívá, až na výjimky ho nahradila počítačová sazba. Z původního řemesla se ale dochovala sazečská terminologie, rozměrové jednotky a některá základní typografická pravidla.

Od vzniku knihtisku se postupně vymezovala pravidla pro sazbu. Tato pravidla určují způsob sázení liter do sazebnice podle významu sázeného textu a používají se dodnes. Každý jazyk může mít různá pravidla pro sazbu a čeština je toho zářným příkladem, takže v jedné z následujících kapitol bude věnován čas výkladu specifik sazby v českém jazyce.

²Tisk pomocí sazby pohyblivých znaků se v Číně používal už v 11. století a podobná metoda byla známa i v antickém Římě. Už Cicero zmiňuje možnost tisku textu sestaveného z jednotlivých písmen.

V dnešní době naštěstí většinu práce obstarají textové editory a speciální nástroje pro sazbu, ale vždy je potřeba lidského „dohledu“ na práci počítače a kontroly výstupní sazby. Také při psaní vstupního textu je potřeba dodržet několik zásad, aby výsledek odpovídal platné normě³. Podrobnosti o zásadách správné sazby budou popsány v dále.

3.2 Obecné základy typografie

3.2.1 Typometrické jednotky

Pro popis jednotlivých prvků při sazbě se používá několik typů jednotek. Použití jednotlivých typů většinou závisí na tom, kterého objektu se rozměr týká. Zásadní dělení měrných jednotek je na absolutní a relativní.

Absolutní jednotky se nejčastěji používají pro nastavení základních rozměrů stránky, výšky záhlaví a zápatí, šířky postranních okrajů a podobně. Tyto rozměry se dnes většinou uvádějí v metrické a angloamerické soustavě, tedy v milimetrech a palcích.

Druhým klasickým případem použití absolutních jednotek je určení velikosti jednoho bodu při tisku. Jeho velikost, tedy vlastně kvalita tisku, se standardně zapisuje jako počet bodů na jeden palec a značí se DPI⁴. Čím je hodnota DPI vyšší, tím je výstupní rastr jemnější. Při výstupu na některá specifická zařízení, jako jsou osvitové jednotky, se někdy používá počet linek na palec, tedy LPI⁵.

Relativní měrné jednotky vycházejí z rozměru použitého písma a používají se nejčastěji pro nastavení řádkování, odsazení odstavců a určení velikosti mezer. Nejčastěji používaná jednotka *em* je dána velikostí písmene „m“.

3.2.2 Rozměry písma

Základní velikost písma, tedy jeho stupeň, je určen světlou výškou písma. Hodnota stupně písma je přibližně určena vzdáleností mezi horní a spodní dotažnicí (viz obr. 1 na str. 10), ale je vždy o něco vyšší. Historicky tento rozměr udává velikost kuželky při tradiční sazbě. Jako jednotka pro určení stupně písma se používají body [b]. Nejčastější stupeň běžného textu je 8–12 b.

³ČSN 01 6910 Úprava písemností psaných strojem nebo zpracovaných textovými editory. Praha, Český normalizační institut, 1997

⁴Dots Per Inch

⁵Lines Per Inch



Obrázek 1: Rozměry písma

3.2.3 Písmo

Jako v mnoha jiných odvětví i zde platí, že méně je někdy více. Pro volbu barev a písma neexistují žádná striktní pravidla a jejich použití je velmi závislé na typu sázeného dokumentu. Nikdy v tomto ohledu neškodí střídmost. U čistě textových dokumentů většinou není důvod používat několik barev písma, pro zvýraznění části textu jsou vhodnější jiné prostředky. Obecně se považuje za vhodné pro zvýraznění textu použít jiný řez stejného typu písma.

Každý *typ* písma může mít několik *řezů*, které dohromady tvoří *rodinu* písma. Základní řezy jsou *kurzíva*, *polotučné písmo*, *tučné písmo*, *velmi tučné písmo*, *tučná kurzíva* a *kapitálky*. Každý řez písma může být navíc ještě podtržený. Pro zvýraznění textu v tištěných materiálech je nejčastěji používána kurzíva. Podtrhávání či tučné písmo působí na čtenáře příliš agresivně a označený text až příliš zdůrazňují.

Kurzíva je řez základního písma s mírným sklonem. Dělí se na kurzívu pravou a kurzívu nepravou. Pravá kurzíva je samostatný řez navržený v mírném náklonu, nepravá kurzíva je vytvořena počítačem nakloněním základního řezu. U některých písmen se v řezu nepravé kurzívy projevují viditelné deformace.

Tloušťka řezu je určena šířkou tahu písma, takzvaným duktem. Duktus se udává jako poměr tloušťky tahu a výšky písma. Písma s vyšším duktem jsou tučná, písma s nižším se nazývají zeslabená. Příliš vysoká hodnota duktu má negativní vliv na čitelnost.

Kombinací základního řezu, kurzívy a duktu vzniknou všechny uvedené řezy kromě kapitálek. Kapitálky jsou odvozené od verzálek (velkých písmen) snížením výšky na střední výšku písma. I zde, stejně jako u kurzívy, existují kapitálky pravé a nepravé. Pravé jsou samostatným řezem písma, zatímco nepravé jsou počítačem sniženou verzí řezu verzálek. I zde se u nepravé varianty verzálek projevuje deformace, v tomto případě se projeví zeslabením duktu.

Rodiny písem se dále dělí podle dvou základních kritérií na patková a bezpatková a dále na proporcionální a neproporcionální⁶. Patka u patkových písem je tvořena příčným tahem na zakončení některých tahů. Typickým zástupcem skupiny patkových písem je rodina písma Times New Roman, bezpatkovou zastupuje rodina písma Helvetica (Arial). Obě dvě uvedené rodiny písma patří do skupiny písem proporcionálních, tedy každé písmeno může mít jinou šířku.

Na rozdíl od proporcionálních písem mají písmena písma neproporcionálního jednotnou šířku. Tato písma se v typografii nikdy moc nepoužívala, jejich vznik je spíše spojen s psacím strojem a s počáteční fází publikování textů na počítači. Jednotná šířka písmen neumožňuje provést několik základních postupů, které jsou v typografii běžné a budou popsány později. Hlavním představitelem neproporcionálních písem je rodina písma Courier. V moderní typografii se neproporcionální písma používají jen ojediněle a většinou pouze pro odlišení části textu. V této práci jsou neproporcionálním písmem vysázené například ukázky programového kódu.

Pro tištěné dokumenty se častěji používají patková písma. Patky u jednotlivých písmen totiž pomáhají čtenáři udržet zrak na jednom řádku a tím usnadňují čtení širších bloků textu. Bezpatková písma se někdy používají pro odlišení nadpisů. Na rozdíl od tištěných dokumentů se například při tvorbě internetových stránek většinou používá opačný postup, tedy patková písma pro nadpisy a bezpatková pro obyčejný text.

serif	sans-serif	monospace
Základní řez	Základní řez	Základní řez
Polotučné písmo	Polotučné písmo	Polotučné písmo
<i>Kurzíva</i>	<i>Kurzíva</i>	<i>Kurzíva</i>
<i>Polotučná kurzíva</i>	<i>Polotučná kurzíva</i>	<i>Polotučná kurzíva</i>
KAPITÁLKY	KAPITÁLKY	KAPITÁLKY

Tabulka 1: Řezy písma

3.2.4 Odstavce

Při sazbě knih se dnes nejčastěji používá zarovnání odstavců do bloku, tedy tak, že všechny řádky mají stejnou délku. Aby výsledný dokument vypadal hezky, je vhodné použít dělení slov, jinak na řádcích vznikají zbytečně dlouhé mezery.

⁶V odporné terminologii se bezpatková písma označují jako *sans-serif*, patková jako *serif* či *roman* a neproporcionální jako *monospace*.

Pravidla dělení slov jsou dána Pravidly českého pravopisu. Dělení se provádí mezi jednotlivými slabikami. Preferované je dělení v místě spojení složených slov, nebo za předponou. Slova se spojovníkem je také vhodné dělit v místě spojovníku, podle Pravidel českého pravopisu se pak spojovník opakuje i na začátku následujícího řádku. Uvedená pravidla se používají při úzké sazbě do třiceti znaků na řádek. V případě širší sazby přicházejí v úvahu další pravidla.

- žádná část děleného slova by neměla mít hanlivý význam (ná-držka)
- nikdy neoddělujeme jednopísmenné slabiky (a-beceda)
- žádná část by neměla vypadat nezvykle (mr-kl)
- dělení slov by se nemělo opakovat na více než šesti po sobě jdoucích řádcích

Mezi časté chyby při sazbě odstavců patří takzvané „parchanty“. Tento pojem v sobě slučuje dvě chyby známé též jako *vdovy* a *sirotcí*. K těmto chybám dochází, pokud se odstavec nachází na místě stránkového zlomu, tedy pokud začíná na jiné stránce než na jaké končí. Pojmem *sirotek* se v typografii označuje poslední (východový) řádek odstavce, který se nachází na další stránce. Analogicky pojem *vdova* označuje první řádek odstavce, který při sazbě uvízl na předchozí stránce. Vdovy nejsou striktně zakázány, ale sirotci ano. Tyto jevy můžou kromě stránkového zlomu nastat i například při sloupcové sazbě při přechodu mezi jednotlivými sloupci.

Další možnou chybou při sazbě odstavců je jev, který se v české terminologii nazývá *řeky*. Tento jev lze jednoduše popsat tak, že se v několika řádcích nad sebou vyskytují na stejném místě mezery, které se spojí do bílého pruhu. Ve výsledné sazbě to nevypadá dobře a sazba vypadá děravě.

Poslední problém při sazbě odstavců je specifický pro česká typografická pravidla a jedná se o „přetahování předložek“. Podle českých pravidel totiž nesmí řádek končit jednohláskovou předložkou či spojkou. Toto pravidlo není při zarovnání odstavců do bloku příliš praktické, protože v jeho důsledku může docházet k nedostatku textu na řádku a ke vzniku moc širokých mezer.

V případě ruční sazby šlo jistě o velmi namáhavou práci, dnes naštěstí sazeče více či méně úspěšně zastupují počítačové programy. Inteligentní sázecí algoritmy optimalizují celý odstavec jako celek. Pro dosažení optimálního vzhledu v různé míře přizpůsobují mezery mezi slovy, písmeny, řádky, někdy se i nepatrně mění stupeň písma. Účelem je docílení správné sazby odstavce, aniž by si čtenář něčeho všiml. V některých případech ale ani počítačové algoritmy nejsou schopny zadaný text

úspěšně vysázet, pak je potřeba upravit vstupní text. Záleží na autorovi, jaké úpravy použije, většinou se ale používají následující:

- změna rozdělení textu do odstavců
- použití, nebo naopak rozepsání zkratk
- vyjádření číslic slovy a naopak
- použití synonym
- přidání či vypuštění části textu

3.2.5 Spojovník, pomlčky a mínus

Mezi nejčastější typografické chyby patří zaměňování znaků, které vypadají jako pomlčka. Ve skutečnosti existují čtyři různé znaky a každý má své specifické použití.

Nejkratším z uvedených znaků je spojovník „-“. Tento znak se v češtině používá hlavně pro vyjádření těsného spojení slov a také pro řádkové dělení (bude-li, Frýdek-Místek). Tento znak se vždy sází bez mezery těsně za první část spojení. Pokud dojde k řádkovému dělení slova v místě spojovníku, sází se navíc i těsně před druhou část.

Druhým z uvedených znaků je krátká pomlčka. Tento znak se také nazývá půlčtverčiková pomlčka „-“, v cizí literatuře „En Dash“. Krátká pomlčka se používá pro znázornění intervalu a někdy také ve významu *versus* (pondělí-pátek, Sparta-Slavia).

Nejdelší z pomlček je dlouhá čtverčiková pomlčka „—“, v cizí literatuře nazývaná *Em Dash*. Tento znak odděluje části textu, nebo označuje pauzu v řeči a v češtině se moc nepoužívá (ano — nebo ne?). Kolem čtverčikové pomlčky se v češtině vždy dělají mezery, v angličtině se sází bez mezer. Při řádkovém zlomu zůstává čtverčiková pomlčka vždy na konci prvního řádku.

Posledním často zaměňovaným znakem je matematické znaménko mínus „-“. Hlavní rozdíl u tohoto znaku oproti pomlčkám je ten, že znak mínus je umístěn výš, aby byl zarovnán se znaménkem plus.

3.2.6 Uvozovky

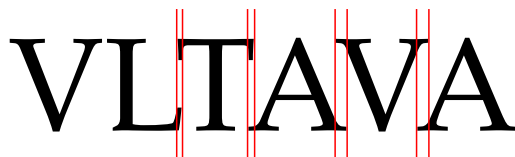
Při sazbě v českém jazyce se používají jiné znaky pro uvozovky, než například v angličtině. Při správném použití uvozovek v českém jazyce se při tiskové sazbě, na rozdíl

od textu psaného na stroji, používají jiné znaky pro otevírací a uvozovací uvozovky. Správný znak pro otevírací uvozovky je „, a pro uzavírací “.

3.2.7 Kerning, ligatury a prostrkávání

Kerning a ligatury jsou způsoby upravení mezer mezi následujícími písmeny, respektive se jedná o způsob sazby konkrétních skupin znaků, pokud se vyskytují za sebou v jednom slově. Účelem těchto úprav je zlepšení celkového optického dojmu z výsledné sazby.

Pro kerning se v české terminologii používají názvy vyrovnání a podřezávání. Název podřezávání vychází z klasické sazby, kdy se některým sousedním literám uříznul roh, aby je bylo možné usadit blíž k sobě. Tím se zajistilo, že vzdálenost problematických dvojic písmen vypadá stejně velká jako vzdálenost písmen ostatních.



Obrázek 2: Kerning

V dnešní době jsou kerningové informace většinou uloženy v definici počítačového fontu, bohužel ne všechny programy tyto informace automaticky využívají. Naopak některé programy umějí doplnit kerningové informace automaticky podle geometrického tvaru znaků i u fontů, které tyto informace neobsahují (autokerning).

Pro ligaturu se v češtině někdy používá název slitek a i zde se jedná o upravení pozice dvou po sobě jdoucích liter. Na rozdíl od kerningu zde nejde pouze o přiblížení dvou písmen, ale i o jejich spojení. Historicky se umísťovala dvě písmena na jeden kovový odlitek a většinou byla nějak propojena.



Obrázek 3: Ligatura

V dnešní době jsou informace o ligaturách uloženy přímo v definici počítačového fontu. V latině se nejčastěji používají ligatury *ff*, *fi* a *fl*. Použití ligatur je závislé na písmu a jazyce, například v arabštině jsou takto svázána skoro všechna písmena,

takže arabské fonty musí pro všechna písmena obsahovat ligaturní informace pro následující případy:

- písmeno je na začátku slova
- písmeno je uprostřed slova
- písmeno je na konci slova
- písmeno je použito samostatně

Některé jazyky, jako například hebrejščina, nepoužívají ligatury vůbec.

3.2.8 Zvýraznění textu

Existuje několik způsobů, jak zvýraznit část textu. V celém dokumentu by se vždy měl používat jednotný systém zvýrazňování odpovídajících částí textu.

Mezi nejčastěji používané způsoby zvýraznění patří *kurzíva* a **tučné písmo**. Mezi méně časté patří *prostrkávání*, tedy zvětšení mezer mezi jednotlivými písmeny. Za vyloženě nevhodný způsob zvýraznění textu je považováno podtrhávání, protože výrazně snižuje čitelnost.

Při zvýrazňování textu se často objevuje jedna typografická chyba. Pokud se zvýrazňuje výčet slov oddělený čárkami či spojkami, vždy je nutné zvýraznit jen jednotlivá slova a nikdy se nezvýrazňují spojky a čárky.

3.3 Publikování dokumentů pomocí počítače

3.3.1 Historie počítačové typografie

Již v době rozšiřování osobních počítačů mezi běžné obyvatele, tedy v devadesátých letech dvacátého století, patřily textové editory k základní výbavě každého počítače. Výstupy tehdejších textových editorů však měly ke kvalitní sazbě opravdu velmi daleko. V naprosté většině případů používaly neproporcionální písma, takže ani nemohly podporovat většinu sazečských fines. Samozřejmě už tehdy existovaly profesionální nástroje pro sazbu textu, ale ty byly pro většinu uživatelů nedosažitelné buď finančně, nebo svojí složitostí.

V pozdější době se na poli textových editorů prosadil jako de facto standard textový editor Microsoft Word. Tento editor přinesl tehdy nevídané množství rodin písem, barev a jiných efektů. Přínos této změny lze posuzovat ze dvou stran:

1. Na jedné straně s příchodem proporcionálních písem výrazně vzrostla kvalita sazby běžných textových editorů. Uživatelé si postupně přivykli na používání stylů a osvojili si některé dobré zvyky při formátování dokumentů.
2. Na druhé straně spousta uživatelů začala používat zbytečně velké množství rodn písmen v jednom dokumentu a tyto dokumenty nevypadaly většinou nejlépe.

Z typografického hlediska jsou jasným přínosem funkce pro vyrovnání textu do bloku či kerning a ligatury. Tyto funkce Word sice podporuje, jejich podpora ale zdaleka není dokonalá. Ani aktuální verze Wordu neprodukuje typograficky kvalitní výstup.

3.3.2 \TeX a jeho varianty

Program pro sazbu textu \TeX vznikl na konci sedmdesátých let dvacátého století a jeho autorem je vysokoškolský profesor Donald E. Knuth. Profesor Knuth údajně nebyl spokojený s kvalitou tisku své poslední publikace, a tak se rozhodl vytvořit vlastní systém pro počítačovou sazbu. Tento systém byl od začátku určen, kromě sazby běžného textu, také k sazbě matematických vzorců a rovnic. Písmeno X na konci názvu pochází z řeckého písmena χ , proto jsou jméno tohoto programu i odvozená jména souvisejících programů vyslovována s písmenem „ch“. V této práci popisované prostředí pro počítačovou sazbu je založeno na \TeX u verze 3.1415926. Číslování verzí totiž u \TeX u neprobíhá zvyšováním čísla verze, ale přidáváním číslí z desetiného rozvoje čísla π , takže se číslo verze asymptoticky blíží k přesné hodnotě π .

Pod pojmem program u \TeX u nejde v žádném případě o WYSIWYG⁷ textový editor, ale spíše o překladač ze speciálního jazyka do formátu PostScript. Jazyk použitý pro sazbu se také nazývá \TeX a skládá se z přibližně 300 primitivních funkcí pro ovlivnění tvorby stránky a pro změnu chování samotného interpreteru. Tato primitiva je možné rozdělit do následujících kategorií:

- funkce pro nastavení způsobu zlomu odstavců a dělení slov
- funkce pro nastavení zlomu stránky a umístění plovoucích objektů
- funkce pro nastavení velikostí mezer mezi všemi objekty
- funkce pro zarovnání textu v tabulkách a maticích
- funkce pro práci se symbolickými názvy

⁷What You See Is What You Get, tedy „co vidíš, to dostaneš“.

- funkce pro nastavení chování $\text{T}_{\text{E}}\text{X}$ u
- funkce pro ovládání $\text{T}_{\text{E}}\text{X}$ u

V současné době existuje mnoho variant $\text{T}_{\text{E}}\text{X}$ u, za zmínku stojí například $\text{P}_{\text{d}}\text{fT}_{\text{E}}\text{X}$, který bude použit v praktické části této práce. Tyto varianty existují pro všechny dnes běžně používané operační systémy a jsou dostupné i ve formě zdrojových kódů.

3.3.3 $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ je balík maker, který usnadňuje autorům sazbu textu. Původním autorem tohoto balíku je Leslie Lamport, dnes se ale o rozvoj stará $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$ team vedený Frankem Mittelbachem. Hlavním cílem tohoto teamu je vytvořit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ verze 3, ale zároveň se i aktivně stará o údržbu stávající verze $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}2_{\epsilon}$.

Vlastní sazba textu je v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u realizována pomocí sázecího systému $\text{T}_{\text{E}}\text{X}$. V této spolupráci $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ určuje styl dokumentu. Tento styl obsahuje například pravidla pro použití písma, odsazení odstavců, velikost meziřádkové mezery, délku řádku apod. $\text{T}_{\text{E}}\text{X}$ podle příkazů $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u vytvoří finální sazbu dokumentu.

Pro $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ existuje velké množství doplňkových balíčků, které dále rozšiřují jeho funkčnost. Jedná se o balíky doplňkových symbolů, balíky pro sazbu čárových kódů či not, balíky pro tvorbu vektorové grafiky a podobně. Správa instalovaných balíčků může být pro technicky méně zdatné uživatele dost náročná, naštěstí v dnešní době existují distribuce $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u pro Microsoft WindowsTM, které tuto proceduru značně usnadňují.

4 Základy sazby textu v makru \LaTeX

4.1 Příkazy \LaTeX u a jejich parametry

Vstupní soubor pro zpracování \LaTeX em je tvořen textem ve formátu ASCII⁸, který může být doplněn o příkazy \LaTeX u. Tyto příkazy vždy začínají zpětným lomítkem „\“, které je následováno jménem příkazu. Jméno příkazu je tvořeno pouze písmeny a překladáč rozlišuje u těchto jmen malá a velká písmena. Jméno příkazu je ukončeno jakýmkoliv znakem, který není písmenem. Za ním mohou v některých případech následovat nepovinné parametry uzavřené v hranatých závorkách [] a povinný parametr ve složených závorkách { }. Obecný tvar příkazu je tedy následující:

```
\prikaz[nepovinne parametry]{povinnny parametr}
```

\LaTeX za jménem příkazu ignoruje jakékoliv neviditelné znaky, takže pokud příkaz nevyžaduje zadání žádných parametrů a má být následován mezerou, je za ním potřeba napsat prázdné složené závorky, nebo znak pro vložení mezery.

4.2 Základní struktura dokumentu

Každý vstupní dokument pro \LaTeX má pevně danou strukturu, která se skládá ze dvou částí: preambule a tělo dokumentu. Na prvním řádku preambule musí vždy být definována třída dokumentu, na dalších řádcích jsou příkazy ovlivňující vzhled dokumentu a příkazy pro načtení doplňkových balíčků maker. V těle dokumentu je zapsán samotný text dokumentu doplněný o formátovací příkazy. Základní kostra vstupního souboru může vypadat například následovně:

```
% začátek preambule
% papír formátu A4, základní velikost písma 12pt, typ article
\documentclass[a4paper,12pt]{article}
% balík pro podporu různých jazyků
\usepackage[czech]{babel}
\begin{document} % konec preambule, začátek těla dokumentu
% zde by měl být obsah dokumentu
\end{document} % konec těla dokumentu
```

⁸American Standard Code for Information Interchange, americký standardní kód pro výměnu informací

Vstupní soubor může kromě příkazů obsahovat také komentáře. Za komentář je považováno cokoli mezi znakem % a koncem řádku. Komentáře mohou být použity jak v preambuli, tak v těle dokumentu.

Některé dokumenty může být výhodné rozdělit do několika souborů a ty pak vložit do hlavního vstupního souboru. Pro tento případ má \LaTeX dva různé příkazy:

1. Příkaz `\include{soubor}` vloží uvedený soubor do hlavního vstupního souboru. Před vloženým souborem \LaTeX začne novou stránku. Pomocí příkazu `includeonly{soubor1,soubor2,...}` je možné v preambuli určit, které soubory budou vloženy. Pokud je nějaký soubor uveden jako parametr příkazu `\include`, ale není uveden v seznamu souborů jako parametr příkazu `\includeonly`, nebude vložen do vstupního souboru
2. Příkaz `\input{soubor}` vloží na své místo uvedený soubor, ale před vložením souboru nezačne novou stránku.

4.3 Třídy dokumentů

Jak bylo uvedeno v kapitole 4.2, nejdůležitější a jedinou povinnou částí preambule je definice třídy dokumentu. Podle vybrané třídy dokumentu a podle nepovinných voleb \LaTeX nastaví základní parametry stránek. Úplný formát příkazu pro definici třídy dokumentu je:

```
\documentclass[options]{class}
```

V následující tabulce je uveden seznam tříd dokumentů.

<code>letter</code>	Třída dokumentu používaná pro krátké dokumenty, jako jsou dopisy. Tato třída byla zavedena v $\text{\LaTeX} 2_{\epsilon}$.
<code>article</code>	Třída pro články a krátké zprávy, které se dělí na sekce
<code>report</code>	Třída pro delší práce, které se dělí na kapitoly
<code>book</code>	Třída pro psaní knih.
<code>slide</code>	Třída pro přípravu prezentací, vhodná pro projekci. Tato třída byla doplněna v $\text{\LaTeX} 2_{\epsilon}$.

Tabulka 2: Třídy dokumentů

Třída dokumentu v sobě spojuje základní nastavení běžná pro daný typ dokumentu. Tato nastavení je možné změnit pomocí nepovinných parametrů tříd. Některé volby ovlivňují chování pouze vybraných tříd.

Jednou z nejdůležitějších voleb je nastavení stupně základního písma. Možné hodnoty jsou 10pt, 11pt a 12pt a výchozí hodnota je 10pt. Nastavení stupně základního písma nemá vliv na dokumenty třídy *slides*.

Dalším důležitým nastavením je volba formátu papíru. L^AT_EX rozlišuje formáty `letterpaper`, `legalpaper`, `executivepaper`, `a4paper`, `a5paper` a `b5paper`. Výchozí hodnota je `letterpaper`. Kromě formátu je samozřejmě možné použít orientaci papíru na šířku, to se provádí uvedením parametru `landscape`.

V závislosti na třídě dokumentu L^AT_EX může, ale nemusí rozlišovat pravé a levé stránky. Důležité je to hlavně kvůli nastavení záhlaví a zápatí při oboustranném tisku. Třída *book* ve výchozím nastavení levé a pravé stránky rozlišuje, to je ale možné změnit použitím parametru `oneside`. Ostatní třídy naopak levé a pravé stránky nerozlišují, je to však možné pomocí parametru `twoside` nastavit. Tato nastavení neovlivňují chování třídy *slides*.

Další parametr ovlivňuje začátky kapitol u tříd dokumentů, které se na kapitoly dělí. Jedná se tedy pouze o třídy *book* a *report*. Kapitoly v těchto třídách vždy začínají na nové stránce. Ve třídě *book* kapitoly začínají vždy pouze na pravé stránce. Použitím parametru `openany` se toto omezení odstraní a kapitoly budou začínat vždy na nejbližším příštím začátku stránky. Ve třídě *report* naopak začínají kapitoly na jakémkoliv začátku stránky. Pro omezení začátků kapitol pouze na pravé stránky slouží parametr `openright`.

Všechny základní třídy dokumentů kromě třídy *slides* podporují sazbu ve více sloupcích. Pro nastavení vícesloupcové sazby slouží parametr `twocolumn`, pro přepnutí do jednosloupcové zase `onecolumn`.

Třídy dokumentů se také liší tím, jakým způsobem sází titulní stranu a abstrakt⁹. Třída *article* po těchto blocích přímo pokračuje sazbou vlastního dokumentu. Pro vysázení samostatné titulní stránky a samostatné stránky s abstraktem slouží přepínač `titlepage`. Ostatní třídy naopak sází titulní stránku i abstrakt samostatně a pro změnu tohoto chování se používá parametr `notitlepage`. Toto nastavení nemá vliv na dokumenty třídy *letter*, u kterých se s použitím titulní stránky či abstraktu nepočítá.

Hlavní zbraní T_EXu byla v začátcích možnost kvalitní sazby matematických rovnic. Z tohoto důvodu existují i parametry, které ovlivňují způsob umístění vzorců na stránce. Výchozí umístění vzorců je na střed stránky a odpovídající číslo vzorce je umístěno na pravém okraji stránky. Při použití přepínače `fleqn` se vzorce zarovnají vlevo, parametr `leqno` nastaví zarovnání čísel vzorců k pravému okraji.

⁹při použití prostředí *abstract*

Posledním důležitým parametrem je `draft`. Při použití této volby označí \TeX černým rámečkem všechny řádky, které nebylo možné automaticky vyrovnat a které přesahují přes definovaný pravý okraj. Tato možnost slouží pro kontrolu automatického zarovnávání do bloku.

4.4 Sazba textu a odstavců

Pro sazbu textu je důležité vědět, jakým způsobem je zpracováván zdrojový soubor. Je-li v textu několik po sobě jdoucích mezer, sloučí je \LaTeX do jedné. Výchozí velikost mezery je $\frac{1}{3}$ čtverčiku, ale při blokové sazbě se může její velikost částečně měnit, aby text rovnoměrně vyplnil celý řádek.

\LaTeX podporuje všechny běžné druhy zarovnání. Ve výchozím nastavení zarovnává \LaTeX odstavce textu do bloku. Pro zarovnání na střed slouží prostředí `center`, pro zarovnání vlevo `flushleft` a pro zarovnání vpravo `flushright`¹⁰.

Odstavec je v \LaTeX u ukončen koncem řádku, nebo příkazem `\par`. Je-li ve zdrojovém dokumentu několik konců řádku po sobě, \LaTeX je sloučí do jednoho. Pro vložení samotného konce řádku bez vytvoření nového odstavce slouží příkaz `\\`.

Při odstavcové sazbě se používají dva způsoby oddělení odstavců. Při prvním způsobu není mezi odstavci zvětšená vertikální mezera, ale první řádek je mírně odsazen. Za vhodnou velikost odsazení je všeobecně považována šířka tří čtverčků. Při druhém způsobu je naopak mezi odstavci zvětšená vertikální mezera, ale nepoužívá se odsazení prvního řádku.

Pro definici velikosti vertikálního odsazení odstavců slouží proměnná `\parskip`. Tato hodnota udává požadovanou, minimální a maximální vzdálenost mezi odstavci. Při sazbě se \LaTeX snaží použít požadovanou hodnotu ale zároveň se snaží odstranit vdovy a sirotky a proto může použít hodnotu z povoleného intervalu.

Pro nastavení odsazení prvního řádku odstavce slouží proměnná `\parindent`. Hodnota proměnné určuje, o kolik bude odsazen první řádek všech odstavců. Potlačit sazbu tohoto odsazení je možné příkazem `\noindent`.

Posledním důležitým příkazem ovlivňujícím vzhled odstavců je `\linespread{číslo}`. Zadané číslo udává zvětšení či zmenšení mezirádkové mezery oproti standardnímu nastavení, výchozí hodnota je tedy 1.

¹⁰Pro zarovnání vlevo a vpravo se v typografii používají termíny sazba na levý a pravý praporek

4.5 Písmo

L^AT_EX obsahuje skupiny příkazů pro nastavení rodiny písma, řezu písma a stupně písma. Tyto příkazy je možné libovolně kombinovat k dosažení požadovaného vzhledu.

4.5.1 Rodina písma

L^AT_EX podporuje všechny tři základní rodiny písem, tedy písma patková, bezpatková a neproporcionální. Příkazy pro tato písma jsou uvedeny v následující tabulce.

patkové písmo	<code>\rmfamily</code>
bezpatkové písmo	<code>\sffamily</code>
neproporcionální písmo	<code>\ttfamily</code>

Tabulka 3: Rodiny písma

4.5.2 Velikost písma

Základní velikost písma bývá v L^AT_EXu 10 pt, 11 pt a 12 pt. Toto nastavení je platné pro celý dokument a od něj se odvíjejí definované velikosti písma. Těchto velikostí je v L^AT_EXu celkem deset a jejich úplný seznam je v následující tabulce. Uvedená tabulka odpovídá nastavení velikosti základního písma 10 pt.

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

Tabulka 4: Velikost písma

L^AT_EX umožňuje i ruční nastavení velikosti písma. K tomu slouží příkaz `\fontsize`, který má dva parametry. První parametr udává velikost písma a druhý velikost řádky. K přepnutí na nastavenou velikost písma slouží příkaz `\selectfont`.

4.5.3 Řez písma

Posledním způsobem ovlivnění vzhledu písma je nastavení jeho řezu. Některé příkazy pro změnu řezu je možné kombinovat, takže například spojením tučného řezu s kurzívou vznikne tučná kurzíva. Některé rodiny fontů však nemusí obsahovat všechny možné kombinace řezů. Následující tabulka uvádí všechny definované příkazy pro změnu řezu písma.

normální (střední) písmo	<code>\textmd</code>
tučné písmo	<code>\textbf</code>
normální (vzpřímené) písmo	<code>\textup</code>
<i>kurzíva</i>	<code>\textit</code>
<i>skloněné písmo</i>	<code>\textsl</code>
KAPITÁLKY	<code>\textsc</code>
<i>zvýrazněné</i>	<code>\emph</code>
běžné písmo dokumentu	<code>\textnormal</code>

Tabulka 5: Řezy písma v L^AT_EXu

Zajímavou vlastnost má příkaz `\emph`, jeho chování je totiž závislé na kontextu. Je-li příkaz použit uprostřed normálního řezu písma, zvýrazní zadanou část textu kurzívou. Pokud je naopak použit uprostřed textu sázeného kurzívou, je zvýraznění provedeno použitím základního řezu písma.

4.6 Uvozovky a pomlčky

V českém jazyce se pro sazbu uvozovek používají jiné znaky, než například v angličtině. V našem prostředí musí být pro uvození vždy použit znak „, a pro uzavření znak “. Při práci v L^AT_EXu s použitím balíku *babel* je definováno makro `\uv{text}`, které před a za uvedený text doplní odpovídající znaky.

Sazba rozdělovníku a dlouhé či krátké pomlčky je v L^AT_EXu velmi jednoduchá. Pro vytvoření rozdělovníku slouží znak `-`, pro krátkou pomlčku sekvence znaků `--` a pro dlouhou pomlčku `---`. Znaménko mínus se vkládá pomocí sekvence `$-$`, kde znaky `$` označují hranice matematického režimu.

4.7 Vnitřní členění dokumentu

Každá třída dokumentu obsahuje informace o možnostech vnitřního členění. Největší počet úrovní pro členění mají dokumenty tříd *report* a *book* a je následující:

1. `\part{...}`
2. `\chapter{...}`
3. `\section{...}`
4. `\subsection{...}`
5. `\subsubsection{...}`
6. `\paragraph{...}`
7. `\subparagraph{...}`

Příkazy `\part{}` a `\chapter{...}` v ostatních třídách dokumentů neexistují.

Pro každý z uvedených příkazů definuje L^AT_EX styl zobrazení a číslování a při každém překladu si ukládá pozice těchto příkazů do pomocného souboru. Tento soubor slouží při příštím zpracování k vytvoření obsahu. Z tohoto důvodu je tedy pro tvorbu obsahu potřeba vícenásobné přeložení vstupního souboru.

Tisk obsahu se zajistí příkazem `\tableofcontents`. Pokud si autor dokumentu nepřeje zahrnutí některé části do obsahu, použije pro její vytvoření takzvanou hvězdičkovou variantu příkazu, například `\subsection*{}`. Takto zapsaná podsektce se v obsahu neobjeví. Je-li název oddílu moc dlouhý, nebo je-li potřeba v obsahu uvést jiný název, zapíše se tento jako nepovinný parametr příkazu, tedy například `\section[Název v obsahu]{Název v textu}`.

4.8 Křížové odkazy a poznámky pod čarou

L^AT_EX podporuje jednoduchý způsob pro tvorbu odkazů na jiné části dokumentu, jako jsou kapitoly, obrázky, tabulky a podobně. Pro vytvoření odkazu je nejprve nutné označit část, kam bude z jiného místa dokumentu odkazováno. Pro toto označení se používá příkaz `\label{jméno návěští}`. Pro vytvoření odkazu v textu se používá dvojice příkazů `\ref{jméno návěští}` a `\pageref{jméno návěští}`. První z nich je při překladu nahrazen číslem odkazovaného prvku a ten druhý číslem stránky, na které se prvek nachází.

```

% definice návěští
\section{Jméno kapitoly}
\label{kap1}
...
...
...
% odkaz na návěští
... v~kapitole \ref{kap1} na straně \pageref{kap1}

```

Pro tvorbu křížových odkazů si \LaTeX při každém překladu vytváří pomocný soubor s čísly stránek, který je použit při následujícím překladu. Proto je při větších změnách vstupního dokumentu vhodné provést překlad několikrát, aby byl použit aktualizovaný pomocný soubor.

Další možností odkázat se na jinou část dokumentu je použití poznámky pod čarou. Těchto poznámek se často využívá k vysvětlení cizích termínů či zkratk. Pro tvorbu poznámky pod čarou slouží v \LaTeX u příkaz `\footnote{text poznámky}`. Na místě tohoto příkazu vysází \LaTeX číslo odkazu a vlastní text poznámky vysází s uvedeným číslem pod čarou ve spodní části stránky.

4.9 Prostředí

Pro změnu způsobu zpracovávání vstupních dat používá \LaTeX takzvaná prostředí. Prostředí je blok kódu, který je uvozen příkazem `\begin{prostředí}[parametry]` a ukončen příkazem `\end{prostředí}`. Uvnitř tohoto bloku pak mohou existovat nové příkazy, nebo se obecné příkazy \LaTeX u mohou chovat jinak.

4.9.1 Výčty

V \LaTeX u existují dvě základní prostředí pro výčet položek. Jedná se o prostředí *enumerate* pro číselované výčty a *itemize* pro nečíselované. Jednotlivé položky se uvozují příkazem `\item` a končí koncem řádku. \LaTeX podporuje až čtyři úrovně zanořených výčtů, číselované a nečíselované je možné v rámci zanoření vzájemně kombinovat.

```

\begin{itemize} % nečíselovaný výčet
  \item první nečíselovaná položka
  \item druhá nečíselovaná položka
\end{itemize}

```

```

\begin{enumerate} % číslovaný výčet
  \item první číslovaná položka
  \item druhá číslovaná položka
\end{enumerate}

```

Výsledek uvedeného kódu se vysází následovně:

- první nečíslovaná položka
 - druhá nečíslovaná položka
1. první číslovaná položka
 2. druhá číslovaná položka

L^AT_EX umožňuje nastavit symboly a typy číslování pro jednotlivé úrovně vnoření. Toto nastavení se provádí pomocí příkazu `\renewcommand{ }{ }`. První parametr je název upravovaného příkazu, druhý je nová hodnota tohoto příkazu. Příkazy pro změnu typu odrážek v nečíslovaných výčtech mají formát `\labelitem[číslo]`, kde číslo udává číslo úrovně a uvádí ho malými římskými číslicemi. Například pro nastavení znaku „x“ jako odrážky třetí úrovně slouží příkaz `\renewcommand{\labelitemiii}{x}`.

Změna stylu číslování u číslovaných výčtů probíhá podobně jako u nečíslovaných výčtů. I zde je potřeba použít příkaz `\renewcommand` s tím rozdílem, že měněný příkaz se jmenuje `\labelenum[číslo]`. Jako novou hodnotu je možné použít příkazy `\arabic`, `\roman`, `\Roman`, `\alph` a `\Alph`. Tyto příkazy zastupují arabské číslice, římské číslice a písmena¹¹. Například pro nastavení stylu číslování čtvrté úrovně na velké římské číslice slouží příkaz `\renewcommand{\labelenumiv}{\Roman}`.

4.9.2 Tabulky

Pro sazbu tabulek slouží prostředí *tabular*. Jako parametr tohoto prostředí se použije řetězec znaků, který nastaví zarovnání jednotlivých sloupců a oddělovač mezi nimi. Jendotlivé buňky se od sebe oddělují znakem `&`, konec řádku tabulky je označen příkazem `\\`. Pro vykreslení čáry oddělující jednotlivé řádky slouží příkaz `\hline`. Vše názorně vysvětluje následující příklad:

¹¹varianty začínající velkým písmenem používají pro značení velká písmena

```

\begin{tabular}{|l|r|} % začátek tabulky
  \textbf{plodina} & \textbf{osetá plocha} \\
  \hline \hline
  kukuřice & 100 ha \\
  \hline
  pšenice & 1000 ha \\
  \hline
\end{tabular} % konec tabulky

```

Výsledkem uvedeného kódu je tabulka 6 na straně 27. Parametr prostředí nastavuje písmenem `l` zarovnání prvního sloupce vlevo, písmeno `r` zarovnání druhého sloupce vpravo a znak `|` určuje, mezi kterými sloupci bude vykreslena oddělovací čára. Další použitelné řídicí znaky jsou `c` pro vycentrování obsahu buňky a `p{šířka}` pro tvorbu sloupce dané šířky s víceřádkovým textem. Pro určení jiného oddělovače slouží konstrukce `@{oddělovač}`. Tato konstrukce se často používá pro zarovnání čísel na desetinnou čárku. Při tomto použití se do prvního sloupce umístí celá část čísla, do druhého desetinná část a jako oddělovač se použije čárka. Specifikace tabulky pak vypadá například `\begin{tabular}{r@{,}l}`.

plodina	osetá plocha
kukuřice	100 ha
pšenice	1000 ha

Tabulka 6: Ukázková tabulka

4.9.3 Matematické vzorce

L^AT_EX vždy nacházel uplatnění při sazbě technických textů, které obsahovaly množství rovnic. Proto existuje hned několik způsobů vložení matematických výrazů do dokumentu. Při vkládání matematických výrazů přímo do textu se nejčastěji celý vzorec uzavírá mezi dvojicí znaků `$`. Dále je možné použít pro uzavření znaky `\(` a `\)`, nebo použít prostředí *math*.

Delší matematické výrazy je vhodné vkládat na samostatný řádek. Pro tento způsob sazby slouží buď oddělovače `\[` a `\]`, nebo prostředí *displaymath* a *equation*. Při použití prostředí *equation* jsou jednotlivé rovnice navíc číslovány. Detailní popis všech příkazů pro sazbu matematických vzorců je nad rámec této práce a lze jej najít například v literatuře [2]. Jako příklad možností matematické sazby je zde uveden vzorec pro Minkowského nerovnost (obrázek 4 na straně 28).

```

\begin{equation}
\label{minkowski}
\Big(\sum_{k=1}^n(x_k+y_k)^p\Big)^{\frac{1}{p}}
\leq
\Big(\sum_{k=1}^n x_k^p\Big)^{\frac{1}{p}}
+
\Big(\sum_{k=1}^n y_k^p\Big)^{\frac{1}{p}}
\end{equation}

```

$$\left(\sum_{k=1}^n(x_k + y_k)^p\right)^{\frac{1}{p}} \leq \left(\sum_{k=1}^n x_k^p\right)^{\frac{1}{p}} + \left(\sum_{k=1}^n y_k^p\right)^{\frac{1}{p}} \quad (1)$$

Obrázek 4: Minkowského nerovnost

4.9.4 Přímý výstup

\LaTeX umožňuje sázet úryvky textu i celé bloky přesně tak, jak jsou zadány ve zdrojovém souboru. To je praktické například pro vkládání částí počítačových programů a podobně. Takto vysázený text přesně zachovává řádkování a mezery. Zároveň v takto sázeném textu nejsou interpretovány žádné příkazy \LaTeX u. Pro sazbu většího bloku textu slouží prostředí *verbatim*, pro krátké části uvnitř odstavců příkaz `\verb`.

Použití prostředí *verbatim* je obdobné jako u ostatních prostředí. Příkaz `\verb` se ale chová trochu jinak. Nemá striktně daný oddělovač obsahu, ale jako oddělovač použije první znak za názvem příkazu. Stejný znak pak musí být použit i pro ukončení vkládaného textu. Jako oddělovač může sloužit jakýkoliv znak kromě písmen, mezery a znaku „*“. Záписы `\verb#text#` a `\verb|text|` jsou tedy rovnocenné.

4.10 Plovoucí objekty

Pro sazbu nedělitelných objektů slouží v \LaTeX u takzvaná plovoucí prostředí. Nedělitelné objekty jsou například obrázky a tabulky, které není možné rozdělit na více stránek. Když překladač narazí na podobný objekt, který už se na aktuální stránku nevejde, musel by vložit konec stránky a objekt přesunout na stránku následující. Tím by vznikaly prázdné konce stránek, které by nevypadaly nejlépe. Plovoucí prostředí řeší tento problém tím, že pokračují dále sazbou textu a zpracovávaný objekt umístí na nejbližší vhodné místo.

V \LaTeX u existují v zásadě dvě plovoucí prostředí, která jsou velmi podobná. Jedná se o prostředí *table* pro vkládání plovoucích tabulek a o prostředí *figure* pro ostatní plovoucí objekty. Obě prostředí mají nepovinný parametr pro určení způsobu umístění objektu na stránce.

Parametr pro umístění se skládá z řetězce znaků, který určuje, kam může \LaTeX objekt umístit. Tyto znaky jsou počátečními písmeny slov **here**, **top**, **bottom** a **page**. Znak **h** značí, že se má překladač pokusit vložit objekt přesně na místo, kde je uveden ve zdrojovém souboru. Pro umístění objektu na začátek stránky slouží znak **t**, pro umístění na konec znak **b**. Při použití znaku **p** \LaTeX umístí objekt na samostatnou stránku. Posledním definovaným znakem je **!**, který „vypne“ většinu omezení pro umístění plovoucích objektů, jako je například maximální počet plovoucích objektů na jedné stránce. Neurčí-li uživatel způsob umístění plovoucího objektu, je výchozí nastavení [**tbp**].

\LaTeX se vždy snaží umístit plovoucí objekt na aktuální stránku. Pokud to z nějakého důvodu není možné, přidá objekt na konec fronty neumístěných objektů a pokračuje na další stránku. Při vytvoření nové stránky se vždy nejdříve kontroluje, zda fronta neobsahuje dostatek objektů pro vytvoření samostatné stránky. Pokud tomu tak není, zpracovávají se postupně objekty ve frontě podle své specifikace pro umístění na stránce. Někdy se může stát, že některý objekt není možné umístit a fronta se „ucpe“. V tomto případě je nutné použít příkaz `\clearpage`, který donutí \LaTeX vysázet všechny objekty ve frontě.

Pro popis plovoucích objektů slouží příkaz `\caption[krátký popis]{popis}`. Uvedený popis se vysází pod plovoucím objektem, nepovinný krátký popis je použit v seznamu objektů. Pro výpis objektů definovaných prostředím *table* slouží příkaz `\listoftables`, pro seznam objektů prostředí *figure* `\listoffigures`. Každé prostředí má definovaný vlastní čítač.

Pro tvorbu odkazů na plovoucí objekty slouží i zde příkazy `\label`, `\ref` a `\pageref`.

4.11 Zvláštnosti českého jazyka

4.11.1 Nastavení prostředí

Výchozí nastavení \LaTeX u odpovídá pravidlům pro sazbu v angličtině, která má svá specifika. Většinu nastavení lze naštěstí provést použitím balíku *babel* s parametrem `czech`. Tím je nastaveno správné dělení slov, délka mezery za koncem věty, používají se správné uvozovky a podobně.

4.11.2 Dělení slov

V některých případech se může stát, že automatické dělení slov selže. Je možné buď definovat výjimky v preambuli, nebo nastavit či zakázat dělení konkrétního slova v textu.

K vytvoření nových pravidel pro dělení slov je určen příkaz `\hyphenation`. Jako parametr tohoto příkazu se používá seznam slov oddělených mezerou. U takto uvedených slov nezáleží na velikosti písmen. Místa pro dělení se označují znakem `-`.

Pokud je slovo možné dělit pouze na jednom místě, používá se pro označení tohoto místa značka `-|`. Pokud si autor některé slovo nepřeje dělit vůbec, stačí ho uvést v seznamu bez naznačeného dělení.

Dochází-li při automatickém dělení slov v některém konkrétním případě k chybě, je možné ve slově určit dělení příkazem `\-`. Označené místo je pak bráno jako jediné přípustné pro dělení.

4.11.3 Přetahování předložek

\LaTeX sám o sobě neumí přetahovat předložky na další řádek. K tomu slouží program *vlna* od Petra Olšáka. Tento program existuje v několika variantách pro různé způsoby volání. Všechny varianty pracují na principu nahrazení mezer za jednopísmennými předložkami a spojkami znakem `~` pro nedělitelnou mezeru.

Původní verze programu *vlna* byl binární soubor volaný z příkazové řádky. Výsledek byl ten, že změnil vstupní soubor a nahradil v něm mezery. Novější verze *encvlna* funguje jako balík pro \LaTeX , který mezery nahrazuje až při překladu, takže nedochází ke změně vstupního souboru. Pro svůj běh vyžaduje rozšíření \TeX u `encTeX`, které není vždy a všude dostupné.

\LaTeX dále umožňuje zamezit řádkovému zlomu mezi slovy příkazem `\mbox{}`. To je vhodné například při sazbě telefonního čísla, aby nebylo zalomeno mezi dvě řádky. Odpadá tím nutnost mezi trojice čísel doplňovat nedělitelné mezery.

5 Návrh řešení na využití \LaTeX u pro sazbu odborného časopisu

5.1 Definice požadavků

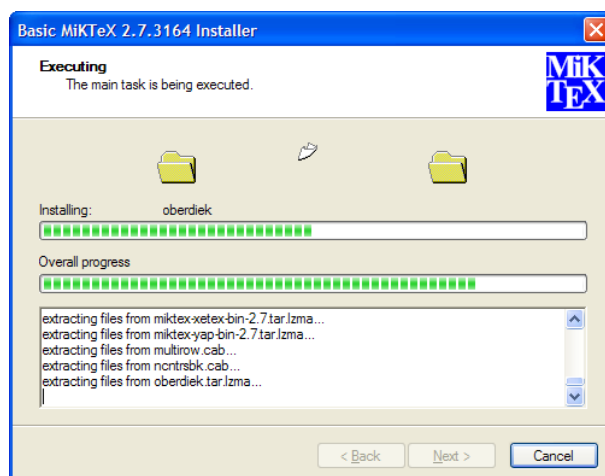
Existuje několik populárních tříd pro autory článků v odborných časopisech. Mezi nejznámější patří Elsevier a IEEE. Cílem této práce je vytvořit podobnou třídu, která bude splňovat následující požadavky:

- v záhlaví titulní strany je umístěno logo časopisu
- pod logem je tučný český titul a anglický titul normální tloušťky
- pod oběma tituly jsou jména autorů a jejich pracovní zařazení
- před článkem je český abstrakt s klíčovými slovy, který je z obou stran oddělen horizontálními čarami
- článek je sázen ve dvou sloupcích s možností použít plovoucí prostředí v rámci sloupce
- za článkem je anglický abstrakt s klíčovými slovy, který je z obou stran oddělen horizontálními čarami
- na konci poslední stránky článku je uvedena kontaktní adresa na autory článku, která je z obou stran oddělena horizontálními čarami.

5.2 Softwarové vybavení pro publikování v \LaTeX u

5.2.1 MiKTeX

Základním kamenem popisovaného publikačního prostředí je \TeX ový překladač. MiKTeX je implementace typografického systému \LaTeX pro operační systém Microsoft WindowsTM. Kromě samotného \TeX u v sobě zahrnuje i velké množství pomocných programů. Dále obsahuje sadu základních fontů, stylů a programových balíčků. Mezi hlavní výhody tohoto produktu patří jednoduchý grafický instalátor (Obr. 5, str. 32), který umožní instalaci i technicky méně zdatným uživatelům. Na domovské stránce autorů je dostupná aktuální verze, v současnosti se jedná o verzi 2.7.



Obrázek 5: Grafický instalátor MiKTeXu verze 2.7

Kromě grafického instalátoru obsahuje MiKTeX také programy pro správu a aktualizaci balíčků a konfigurátor celého prostředí. Nejedná se o jedinou dostupnou distribuci T_EXu pro Microsoft WindowsTM, ale svými možnostmi nejlépe vyhovuje začínajícím uživatelům, protože klade nejmenší nároky na jejich znalosti filozofie T_EXu.

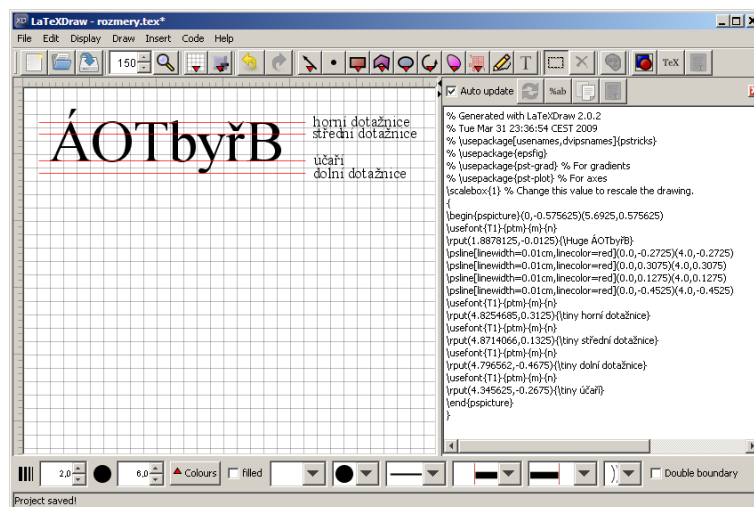
5.2.2 L^AT_EXDraw

Při přípravě odborných článků je často potřeba vytvořit různá schémata či jiné technické obrázky. Pro další zpracování v T_EXu je vždy vhodné ukládat tyto objekty do některého vektorového formátu, aby bylo možné podle potřeby měnit jejich velikost bez ztráty kvality. K těmto účelům dobře poslouží editor L^AT_EXDraw. Možnosti tohoto editoru sice nedosahují možností profesionálních programů, na druhou stranu je ale distribuován zdarma a pro tvorbu základních grafických prvků stačí.

Program L^AT_EXDraw je naprogramován v jazyce Java, takže je přenositelný na všechny běžné operační systémy pro pracovní stanice. Jediné co ke svému běhu potřebuje je Java runtime verze 1.5 a vyšší.

Program obsahuje základní i mírně pokročilé funkce pro práci s textem a vektorovými objekty [9]. Mezi výstupní formáty patří vektorové formáty SVG, EPS a PSTricks, ale i řada rastrových formátů. Pro zpracování v L^AT_EXu se nejlépe hodí formáty EPS a PSTricks. Hlavní rozdíl spočívá v tom, že formát EPS¹² je grafický

¹²Encapsulated PostScript



Obrázek 6: Grafický editor $\text{\LaTeX}Draw$

formát, zatímco PSTricks je textový soubor, který lze vložit do $\text{\LaTeX}u$ pomocí balíčku *pspicture* a lze ho i dále jednoduše editovat.

5.2.3 $\text{\TeX}nicCenter$

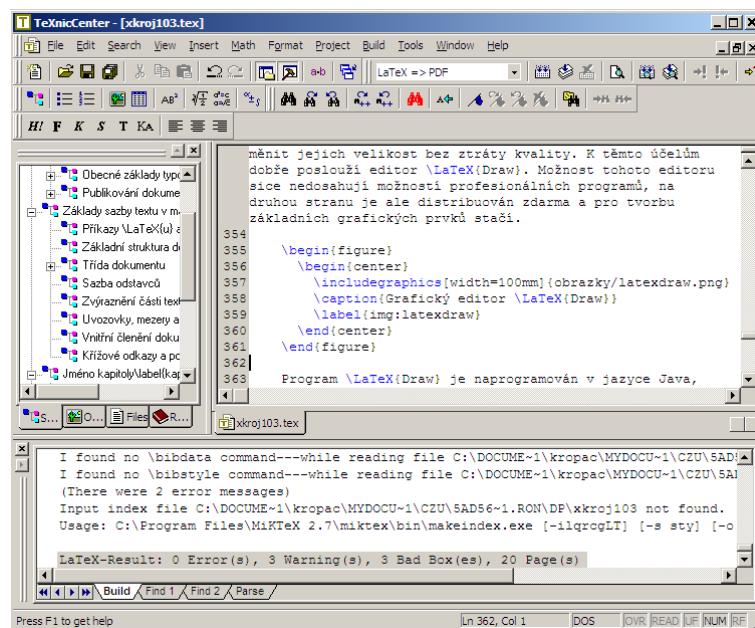
Práce se vstupním souborem pro \LaTeX sice vždy probíhá v textovém režimu, existují ale programy, které tuto práci ulehčí. Nejde totiž jen o přípravu vstupního souboru, ale následně je potřeba zavolat z příkazové řádky interpret a nechat ho vytvořit výstupní soubor. Tyto činnosti a i některé jiné autorům usnadňují speciální textové editory, jako je například $\text{\TeX}nicCenter$ [8].

Kromě základních funkcí textového editoru nabízí $\text{\TeX}nicCenter$ i mnoho užitečných funkcí určených přímo pro \LaTeX . Pro snazší orientaci v textu je v levé části pracovní plochy umístěno navigační okno *Navigator Bar*, které má čtyři záložky.

První záložka *Structure* obsahuje strom dokumentu. Tento strom je automaticky generován podle názvů kapitol a podkapitol a tato funkce pracuje i se soubory vloženými pomocí příkazů *input* a *include*. Jednotlivé názvy ve stromu slouží jako odkazy pro rychlý skok na korespondující řádku ve zdrojovém dokumentu.

Druhá záložka *Objects* obsahuje seznam všech objektů ve zdrojovém dokumentu. Tyto objekty jsou rozděleny do kategorií podle svého typu a jsou pojmenovány podle popisu zadaného pomocí příkazu *caption*.

Třetí záložka *Files* obsahuje seznam všech souborů, které jsou jakýmkoliv způsobem vkládány do zdrojového dokumentu. Tento seznam je rozdělen podle typu souboru a jako název je použita uvedená cesta k souboru.



Obrázek 7: Prostředí editoru T_EXnicCenter

V poslední záložce *References* jsou vypsány všechny položky ze seznamu literatury vytvořeného pomocí programu B_IB_TE_X.

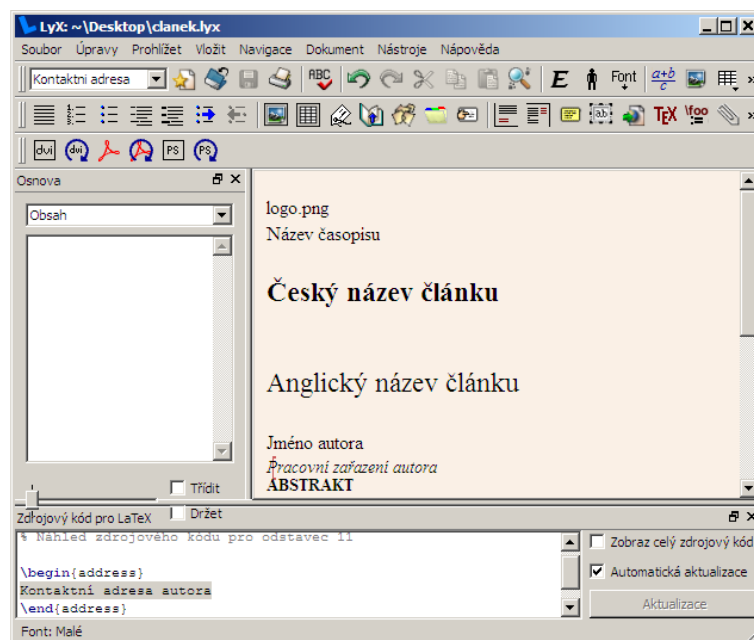
Spodní část pracovní plochy je vyhrazena oknu *Output Bar*. Toto okno slouží pro zobrazení výstupu překladače L^AT_EX_U a všech dalších programů, které jsou volány při překladu zdrojového souboru. Všechny chyby a varování jsou vypisovány do tohoto okna. Pomocí ikon v horním menu je možné zobrazovat postupně jednotlivé chyby a varování, zároveň se vždy v hlavním okně přesune kurzor na problematický řádek ve vstupním souboru.

Vývojové prostředí T_EXnicCenter dále umožňuje definice výstupních profilů, Výstupní profil je vlastně nastavení překladače a parametrů pro zpracování vstupního souboru. Těchto profilů může být definováno několik a není problém je podle potřeby průběžně přepínat. Po instalaci jsou přednastaveny profily pro výstup do PDF, PostScriptu a DVI.

5.2.4 LyX

Textový edit LyX slouží ke zjednodušení tvorby dokumentů v L^AT_EX_U. Pro každou třídu dokumentu je možné vytvořit takzvaný *layout*, který definuje vzhled prvků dané třídy při editaci. Cílem je, aby uživatel už při psaní dokumentu věděl, jak bude výsledný dokument vypadat. Možnosti LyXu zdaleka nedosahují možností L^AT_EX_U,

takže výsledný dokument vždy vypadá odlišně, v některých případech se však může lišit pouze v detailech.



Obrázek 8: WYSIWYM Editor LyX

Program má poměrně přehledné grafické rozhraní, některé činnosti však nejsou příliš intuitivní a chvíli trvá, než se je uživatelé naučí. Dokumentace k programu je velmi obsáhlá, ale autoři ji bohužel příliš neaktualizují. Některé části dokumentace tedy naneštěstí buď úplně chybí, nebo se vztahují k předchozím verzím.

5.3 Tvorba šablony pro \LaTeX

Doplnit do \LaTeX u nové příkazy a prostředí je možné několika způsoby. Nejjednodušší je vložit nové definice do *preambule*, to ale není moc elegantní. Takto vložené definice je potřeba vkládat do každého dokumentu podobného typu.

Druhou možností je vytvoření nového balíku, nebo vytvoření nové třídy dokumentu. Tato možnost je lepší, protože odděluje definice od samotného dokumentu, takže jakákoliv změna se po přeložení projeví ve všech dokumentech stejné třídy, nebo ve všech, které používají stejný balík.

Volba mezi třídou a balíkem závisí na tom, zda je vytvářená funkčnost použitelná pro všechny třídy dokumentu, nebo zda je omezena jen na jednu. Pokud je použitelná pro všechny standardní třídy, je vhodné ji uložit jako balík, aby se dala se všemi

třídami používat. Pokud je spojena s jednou třídou, pak je vhodné ji vložit také do třídy a zdědit vlastnosti vyžadované třídy.

Popisovaný ukázkový příklad je založen na upravení třídy *article*, proto budou vytvářené definice příkazů a prostředí uloženy jako třída. Definice třídy se skládá z několika částí, které nejsou vždy povinné. Maximální kostra definice třídy je:

1. Identifikace
2. Inicializace
3. Deklarace parametrů
4. Vykonání parametrů
5. Nahrání děděné třídy
6. Hlavní blok kódu

V následujících podkapitolách bude postupně popsána tvorba nové třídy po jednotlivých blocích.

5.3.1 Identifikace

Část „Identifikace“ slouží k nastavení základních parametrů nové třídy. V této části se definuje jméno třídy, její verze, požadovaná verze L^AT_EXu a podobně

```
%-----< identification >-----  
\NeedsTeXFormat{LaTeX2e}  
\ProvidesClass{clanek}[2009/03/01 Ukazkova trida dokumentu]
```

První použitý příkaz říká překladači, že třída pro svůj běh vyžaduje L^AT_EX 2_ε. Druhý příkaz definuje jméno třídy, datum vydání a stručný popis. Při deklaraci typu třídy v dokumentu je možné použít nepovinný parametr, který udává minimální verzi třídy. Jako hodnota tohoto parametru slouží datum vydání třídy ve formátu *rrrr/mm/dd*.

5.3.2 Inicializace

V části „Inicializace“ je možné nahrát balíky maker, které třída využívá. I zde je možné nepovinným parametrem určit minimální datum vydání, jako je tomu v případě balíku *geometry*.

```

%-----< initial code >-----
\RequirePackage{multicol}
\RequirePackage{graphicx}
\RequirePackage{calc}
\RequirePackage[a4paper]{geometry}[2008/12/21]
\RequirePackage{fancyhdr}
\RequirePackage[cp1250,latin2]{inputenc}
\RequirePackage[T1]{fontenc}
\RequirePackage[czech,english]{babel}

```

Balík *multicol* usnadňuje použití vícesloupcové sazby. Vícesloupcovou sazbu je možné zapnout parametrem `twocolumn` při deklaraci typu třídy dokumentu, balík *multicol* ale nabízí zajímavé rozšířené možnosti pro sloupcovou sazbu. Mezi nejdůležitější vylepšení patří možnost kombinace vícesloupcové a jednosloupcové sazby v rámci jedné stránky. Další zajímavou funkcí je možnost sazby do více než dvou sloupců. Poslední důležitou vlastností je automatické vyrovnání délky sloupců na poslední straně.

Balík *graphicx* rozšiřuje podporované formáty obrázků, se kterými mohou autoři dokumentů pracovat. \LaTeX bez rozšíření podporuje pouze formáty založené na PostScriptu, s tímto balíkem není problém vkládat většinu běžných obrazových formátů, jako jsou JPG, PNG a GIF.

Pro nastavení rozměrů stránky jsou použity balíky *geometry* a *calc*. První z nich výrazně zjednodušuje definici vlastních rozměrů stránky. Mezi jeho hlavní vlastnosti patří automatické dopočítávání nedefinovaných rozměrů podle těch uvedených. Druhý balík zase usnadňuje počítání s definovanými rozměry.

K vylepšení základních funkcí pro definování záhlaví a zápatí stránky je použit balík *fancyhdr*. Tento balík je brán ve svém oboru jako de facto standard a nabízí nevídané množství možností, z nichž je v tomto návrhu použita jen velmi malá část.

Poslední tři balíky jsou určeny k nastavení národního prostředí, vstupního kódování a fontů. V popisovaném případě je potřeba nahrát národní prostředí pro češtinu a angličtinu, protože tvořený dokument bude vícejazyčný. Vstupní kódování pro češtinu Windows-1250 naštěstí vyhovuje i angličtině, takže není potřeba řešit problém s kódováním. V případě odlišnějších jazyků by bylo vhodné použít například kódování UTF-8.

5.3.3 Deklarace parametrů

Při tvorbě nové třídy je většinou vhodné vycházet z jiné dříve definované. Té je možné předat i některé zadané parametry.

```
%-----< declaration of options >-----  
\PassOptionsToClass{onecolumn,10pt}{article}  
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}
```

Uvedené příkazy zajistí, že třídě *article* budou vždy předány parametry `onecolumn` a `10pt` a dále všechny parametry, které použije autor dokumentu při deklaraci jeho třídy.

5.3.4 Vykonání parametrů

Příkaz `\ProcessOptions` nastaví všechny parametry, které byly dříve nastaveny pomocí příkazů `\DeclareOptions`.

```
%-----< execution of options >-----  
\ProcessOptions\relax
```

5.3.5 Nahrání děděné třídy

Cílem příkazů této sekce je nahrání původní třídy či balíku `maker`. V ukázkovém příkladu jde o rozšíření třídy *article*, proto je nutné ji nahrát.

```
%-----< class loading >-----  
\LoadClass{article}
```

5.3.6 Hlavní blok kódu

Poslední a často nejobsáhlejší část definice nové třídy slouží k definování nových příkazů a prostředí, nebo ke změně již existujících. V této podkapitole budou postupně popsány všechny použité definice.

```
%-----< main code >-----  
%--- nastavení rozměrů stránky  
\geometry{nohead,margin=1.8cm,bindingoffset=0.3cm}
```

Na začátku hlavního kódového bloku je definice rozměrů stránky pomocí balíku *geometry*. Uvedené parametry by bylo samozřejmě možné zadat při přidávání tohoto balíku příkazem `\RequirePackage`, všechna nastavení by ale byla přepsána při nahrání třídy *article*. Uvedená definice nastavuje okraje stránky na 1,8 cm a ruší oblast pro záhlaví. Zároveň nastavuje změnu okrajů při oboustranném tisku tak, že se tělo stránky posune o 3 mm od vnitřního okraje. Podobné nastavení by bylo bez balíku *geometry* velmi komplikované.

V dalším kroku je definováno zápatí pro jednotlivé stránky pomocí maker balíku *fancyhdr*. Nejprve je nutné změnit styl stránek na *fancy* a zrušit všechna výchozí nastavení příkazem `\fancyhf{}`. Změnou tloušťky čar `\headrulewidth` a `\footrulewidth` na hodnotu `0pt` je dosaženo skrytí čar oddělujících záhlaví a zápatí.

```
%--- nastavení záhlaví a zápatí
\pagestyle{fancy}
\fancyhf{}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
\fancyfoot [RO,LE]{\small\thepage}
\newcommand{\journal}[1]{\fancyfoot [LO,RE]{\small#1}}
```

Samotné nastavení obsahu záhlaví a zápatí probíhá pomocí maker `\fancyfoot` a `\fancyhead`. První parametr určuje, o kterou část záhlaví či zápatí se jedná. Toto určení probíhá zadáním dvojic písmen. První písmeno může nabývat hodnot **L** pro levou část, **R** pro pravou část a **C** pro střed. Druhé písmeno určuje, zda se jedná o lichou—**O** či sudou—**E** stránku. Při jednostranném tisku se použije nastavení pro liché stránky.

```
\fancyfoot [RO,LE]{\small\thepage}
\newcommand{\journal}[1]{\fancyfoot [LO,RE]{\small#1}}
```

První z uvedených příkazů určuje, že v pravé části zápatí lichých stránek a v levé části zápatí sudých bude uvedeno malým písmem číslo aktuální strany. Druhý příkaz definuje makro `\journal` s jedním povinným parametrem. Tento parametr se vysází také malým písmem do zápatí levých lichých a pravých sudých stránek. Toto makro je určeno k vysázení jména časopisu do zápatí.

Dále je nutné definovat makra pro zadání cesty k logu časopisu a pro zadání názvu článku v anglickém jazyce. To mají na starost první dva příkazy v následujícím výpise.

```
%--- nová makra pro zadání loga a anglického titulu
\newcommand{\logo}[1]{\def\@logo{#1}}
\newcommand{\alttitle}[1]{\def\@alttitle{#1}}
\alttitle{}
\logo{}
```

Po definování uvedený kód nastaví hodnotu obou proměnných na prázdný řetězec. To je důležité pro následné testování, zda byla hodnota nastavena. Po uvedené inicializaci jsou obě hodnoty definované a prázdné, takže je možné je porovnávat s `@empty`.

Nyní je na řadě redefinice makra `\maketitle`, které v \LaTeX u standardně slouží k sazbě titulní strany. Třída *article* za titulní stranou nekládá konec stránky, takže v tomto ohledu není potřeba nic přenastavovat. Při změně základních maker \LaTeX u je dobrým zvykem uložit jejich původní hodnotu. K tomu je možné použít příkaz `\let`.

```
%--- záloha původního makra \maketitle
\let\oldtitle = \maketitle
```

Po uložení původní hodnoty `\maketitle` je tedy možné přejít k tvorbě nové definice. Jak už bylo řečeno, na titulní stránce bude logo, název a alternativní název. Definice příkazu `\maketitle` odpovídá zadání a skládá se z bloků pro sazbu jednotlivých částí titulní stránky.

První blok pouze přepne balík *babel* na sazbu v českém jazyce. Tento příkaz je nutný, protože je v dokumentu použito několik jazyků. Většina obsahu je v češtině, proto je vhodné provést na začátku dokumentu toto jazykové nastavení.

Druhý blok testuje, zda je určena cesta k souboru s logem. Pokud ano, vloží obrázek do stránky a nastaví jeho výšku na jeden palec. Šířka je dopočítána automaticky tak, aby byl zachován poměr stran.

Třetí blok zajišťuje sazbu hlavního nadpisu. Tento nadpis je uzavřen v prostředí *flushleft*, které nastaví sazbu na levý praporek. Při tomto způsobu sazby se \LaTeX nesnaží zarovnat všechny řádky na stejnou délku, ale jednoduše je zarovná k levému okraji. V tomto případě není potřeba testovat nastavení textu pro hlavní nadpis, protože to testuje \LaTeX při volání příkazu `\maketitle`.

Čtvrtý blok slouží k sazbě alternativního nadpisu v angličtině. Nejdříve proběhne test, zda autor zadal hodnotu tohoto nadpisu. Alternativní nadpis není povinný, takže pokud není zadán, nedojde k žádné chybě. Pokud byla makrem `\alttitle`

nastavena hodnota alternativního nadpisu, je změněno jazykové prostředí na anglické a tato hodnota je vysázena opět na levý praporek. Po vysázení alternativního nadpisu je jazykové prostředí změněno zpět na české.

```
%--- definice nového makra \maketitle
\renewcommand\maketitle{
  \selectlanguage{czech}

  \ifx\@logo\@empty\relax\else
    \noindent{\includegraphics[height=1in]{\@logo}}\par
  \fi

  \noindent
  \begin{flushleft}\Large\bfseries\@title\end{flushleft}

  \ifx\@alttitle\@empty\relax\else
    \selectlanguage{english}
    \par\par
    \noindent\begin{flushleft}\Large\@alttitle\end{flushleft}
    \selectlanguage{czech}
  \fi
}
```

Na titulní straně je kromě loga, nadpisu a nadpisu v angličtině ještě potřeba umístit jméno autora či autorů. Pro zadání jména autora existuje v \LaTeX u standardně makro `\author`. Pro pozdější využití této třídy v programu LyX je vhodnější definovat pro zadání jména autora samostatné prostředí, definice třídy tedy obsahuje nové prostředí *authors*.

```
%--- definice prostředí pro sazbu jména autora a jeho zařazení
\newenvironment{authors}{
  \par\noindent\bfseries
}{
  \vspace{0.1em}
}
```

Uvedená definice potlačuje odsazení první řádky odstavce, zvolí tučné písmo a nastavuje spodní odsazení. Podle definovaných požadavků musí být možné ke jménu autora zadat ještě firmu či oddělení, kde je zaměstnán. Pro zadání jména oddělení je definováno makro `\dept`:

```
%--- definice makra pro sazbu pracovního zařazení autora
\newcommand{\dept}[1]{
  \vspace{0.1em}
  \par\noindent\normalfont{\itshape{#1}}
}
```

Uvedená definice určuje, že text předaný makru `\dept` bude začínat na novém řádku bez odstavcového odsazení, bude vysázen kurzívou výchozího písma a bude shora odsazen o $\frac{1}{10}$ velikosti čtverčíku. Toto makro je určeno pro použití uvnitř prostředí *authors*.

Definice třídy dále obsahuje prostředí *abstrakt* a *abstract* pro sazbu abstraktů v českém a anglickém jazyce. Tato makra jsou si velmi podobná, liší se v nadpisu a v nastavení použitého jazyka pro anglický abstrakt. Díky vzájemné podobě obou prostředí jsou jejich hlavní vlastnosti definovány pomocí shodných maker `\abstropen` a `\abstrclose`.

```
%--- začátek abstraktu
\newcommand\abstropen[1]{
  \noindent\tabular{@{}p{\textwidth}@{}}
  \hline
  \small
  {\bfseries{#1:}}
}

%--- konec abstraktu
\newcommand\abstrclose{
  \\ \hline
  \endtabular
}
```

Makro `\abstropen` má jeden povinný parametr a tím je právě nadpis, kterým se český a anglický abstrakt liší. Toto makro založí novou tabulku na šířku celého textového bloku stránky. Protože abstrakt má podle zadání být oddělen od ostatního textu horizontálními čarami, je na začátku tabulky použit příkaz `\hline` pro vykreslení počáteční čáry. Potom je nastaven stupeň písma na malé a tučným řezem je vysázen zadaný nadpis.

Pro ukončení obou prostředí pro sazbu abstraktů je definováno makro `\abstrclose`. V definici uvedeného makra je pouze vykreslena druhá horizontální oddělovací čára a pak je ukončena tabulka založená makrem `\abstropen`.

Definice prostředí *abstrakt* pouze před sazbou zadaného textu použije makro `\abstropen` se zadaným nadpisem „ABSTRAKT“ a po sazbě zadaného textu zavolá makro `\abstrclose`.

```
%--- prostředí pro sazbu českého abstraktu
\newenvironment{abstrakt}{%
  \abstropen{ABSTRAKT}
}{
  \abstrclose
}
```

Příkaz `\abstract` již v \LaTeX u existuje, je tedy potřeba jeho definici změnit. Nová definice se od definice prostředí *abstrakt* liší zadaným nadpisem „ABSTRACT“ a změnou jazyka před sazbou zadaného textu na anglický a po sazbě zadaného textu zpět na český.

```
%--- prostředí pro sazbu anglického abstraktu
\renewenvironment{abstract}{
  \selectlanguage{english}
  \abstropen{ABSTRACT}
}{
  \abstrclose
  \selectlanguage{czech}
}
```

Oba abstrakty mohou obsahovat klíčová slova. Ta se liší pouze použitým nadpisem, takže je použit podobný způsob definice jako u abstraktů. Pro vložení klíčových slov třída definuje makra `\klicslova` a `\keywords`. Jako parametr těchto maker se zadávají vlastní klíčová slova.

```
%--- makra pro zadání klíčových slov (cs/en)
\newcommand\klicslova[1]{\kwrds{Klíčová slova}{#1}}
\newcommand\keywords[1]{\kwrds{Keywords}{#1}}
```

Obě uvedené definice předávají zadaný text a definovaný nadpis makru `\kwrds`, které pak zajišťuje samotnou sazbu. Celý blok klíčových slov je sázen kurzívou stejného stupně, jaký je definován pro poznámky pod čarou, nadpis navíc její tučnou variantou. Protože se klíčová slova používají uvnitř abstraktu, není nutné v jejich anglické variantě přepínat jazykové nastavení.

```
%--- makro pro sazbu klíčových slov
\newcommand{\kwrds}[2]{
  \\\footnotesize
  \itshape
  {\bfseries#1:}~#2
}
```

Po českém abstraktu má podle zadání přijít na řadu vlastní tělo článku, které je vysázeno ve dvou sloupcích. Pro zjednodušení práce s balíkem *multicol* třída definuje prostředí *clanek*, které tento způsob sazby nastavuje.

```
%--- prostředí pro sazbu ve dvou sloupcích
\newenvironment{clanek}{
  \begin{multicols}{2}
}{
  \end{multicols}
}
```

Posledním blokem v článku je adresa autora či autorů. Tento blok má být podle zadání umístěn na konci poslední stránky a má být oddělen z obou stran horizontálními čarami. Pro sazbu bloku s adresou je v definici třídy definováno prostředí *address*. První příkaz z definice `\vfill` vytvoří maximální možnou vertikální mezeru a tím umístí celý blok na konec stránky. Text bloku je dále uzavřen v tabulce, stejně jako tomu bylo u obou abstraktů. Text tohoto bloku je sázen malým písmem a je uveden nadpisem „Kontaktní adresa:“ v kurzívě.

```
%--- prostředí pro sazbu kontaktní adresy
\newenvironment{address}{
  \vfill
  \noindent\tabular{@{}p{\textwidth}@{}}
  \hline
  \vspace{0.1em}
  \small
  {\itshape{Kontaktní adresa:}}\\
}{
  \\
  \hline
  \endtabular
}
```

Definicí prostředí *address* končí část třídy *clanek*, která se zabývá nastavením rozvržení jednotlivých částí článku. Dále je ještě potřeba nastavit vzhled a chování jednotlivých prvků uvnitř textu. V první řadě je vhodné zrušit číslování jednotlivých sekcí a podsekcí, protože ve víceloupcové sazbě tato čísla zbytečně zabírají místo a navíc u článků není zvykem nadpisy jednotlivých částí číslovat. Makro `\secnumdepth` v tomto případě nastaví číslování kapitol do nulté úrovně, takže žádné nadpisy nebudou číslovány.

```
%--- číslování sekcí
\setcounter{secnumdepth}{0}
```

Aby bylo možné umisťovat plovoucí prostředí do jednotlivých sloupců, je nutné změnit jejich definice. Nejprve je vhodné uložit stávající definice měněných prostředí *table* a *figure*. K tomu slouží stejně jako v případě zálohování makra `\maketitle` příkaz `\let`.

```
%--- záloha původních prostředí
\let\oldtable=\table
\let\oldfigure=\figure
```

Při redefinici plovoucích prostředí je nutné použít makro `\@capytype`. \LaTeX umí se znakem `@` pracovat dvěma způsoby. Může ho považovat za písmeno, nebo za speciální znak. Pro použití makra `\@capytype` je nutné, aby byl znak `@` považován za písmeno. K tomu slouží příkaz `\makeatletter` před redefinicemi obou prostředí.

```
%--- Změna chování znaku @
\makeatletter
```

Nyní je možné přejít k definici prostředí pro plovoucí tabulky uvnitř sloupců. Samotná definice je velmi prostá, kromě deklarování nového typu plovoucího prostředí pouze nastavuje vertikální odsazení od okolního textu.

```
%--- Plovoucí tabulka v sloupcích
\renewenvironment{table} {
  \def\@capytype{table}
  \vspace{0.2em}
}{
  \vspace{0.2em}
}
```


Obdobně jako prostředí *tabular* je nutné redefinovat i prostředí *figure*. Definice je tedy naprosto stejná.

```
%--- Plovoucí figure v sloupcích
\renewenvironment{figure}{
  \def\@capttype{figure}
  \vspace{0.2em}
}{
  \vspace{0.2em}
}
```

Ve výchozím nastavení L^AT_EX považuje @ za speciální znak, proto je po definicích plovoucích prostředí nutné přepnout chování L^AT_EXu zpět příkazem `\makeatother`. Kdyby tomu tak nebylo, mohlo by docházet k problémům v ostatních použitých rozšiřujících balících maker.

```
%--- Změna chování znaku @
\makeatother
```

5.4 Uložení třídy a konfigurace MiK_TE_Xu

Vytvořená třída by měla být uložena v souboru s příponou, například `clanek.cls`. Třidu je sice možné mít v aktuálním pracovním adresáři, není to ale příliš vhodné. Nejlepší je umístit soubor do samostatného podadresáře k ostatním třídám dokumentů, které jsou nainstalovány v distribuci MiK_TE_X. Ve výchozím nastavení se jedná o adresář:

```
C:\Program Files\MiKTeX 2.7\tex\latex\clanek\clanek.cls
```

Tím ale celá práce nekončí, protože MiK_TE_X zatím o nové třídě neví. K „instalaci“ třídy je nutné ještě vykonat následující kroky:

1. Z příkazové řádky operačního systému spustit program `texhash`. Uvedený program aktualizuje interní databázi `ls-R`.
2. Z nabídky start spustit program `Settings`, který je ve složce MiK_TE_X 2.7. Z nabídky programu vybrat volbu `Refresh FNDB`.

Po uvedených krocích bude nová třída dostupná pro L^AT_EX nezávisle na aktuálním adresáři.

5.5 Použití vytvořené třídy v L^AT_EXu

Použití vytvořené šablony je velmi jednoduché. Nejdůležitější je v preambuli dokumentu použít třídu dokumentu *clanek*. Dále je v preambuli vhodné nastavit všechny hodnoty, které jsou použity při sazbě titulní strany.

```
\documentclass[a4paper]{clanek}

\logo{soubor s logem}
\journal{název časopisu}

\title{český titul}
\alttitle{anglický titul}
```

Uvedené příkazy zvolí třídu dokumentu *clanek* a nastaví všechny parametry titulní strany. Nyní je možné přejít k začátku samotného dokumentu, tedy k sazbě titulní strany, jmen autorů a českého abstraktu s klíčovými slovy.

```
\begin{document}

\maketitle

\begin{authors}
  Autor 1
  \dept{zaměstnání}
\end{authors}

\begin{authors}
  Autor 2
  \dept{zaměstnání}
\end{authors}

\begin{abstrakt}
  český abstrakt...
  \klicslova{česká klíčová slova}
\end{abstrakt}
```

Uvedený kód zajistí sazbu titulní strany a českého abstraktu. Po tomto bloku už přímo následuje obsah článku, který je sázen do dvou sloupců. K nastavení dvojsloupcové sazby je ve třídě *class* hlavně kvůli pozdějšímu použití v programu LyX definováno prostředí *clanek*.

```
\begin{clanek}
Článek...
\end{clanek}
```

Na konci článku je anglický abstrakt a adresa autora. Tím je článek ukončen a je možné ho přeložit do jakéhokoliv výstupního formátu.

```
\begin{abstract}
  Anglický abstrakt
  \keywords{anglická klíčová slova}
\end{abstract}

\begin{address}
Adresa
\end{address}

\end{document}
```

5.6 Tvorba šablony pro LyX

Program LyX používá speciální soubory šablon (*layout*). Tyto šablony nastavují pro každý prvek použité třídy parametry pro zobrazení při editaci. Vzhled editovaného dokumentu se pak blíží vzhledu finálního dokumentu. V této podkapitole bude postupně na příkladech popsána tvorba šablony pro třídu *clanek*. Nebudou zde uvedeny definice všech stylů, ale jen ty, ve kterých jsou použity některé důležité parametry.

Začátek definice šablony je velmi podobný definici třídy dokumentu, ke které se váže. Na druhém řádku je v hranatých závorkách uvedeno jméno třídy dokumentu. Ve složených závorkách je název, pod kterým bude šablona dostupná v LyXu.

```
## Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[clanek]{article (Casopis)}
```

Následující bloky příkazů definují vazby mezi styly v LyXu a příkazy v L^AT_EXu. U každé definice budou popsány pouze nové parametry, protože jejich význam se vždy opakuje a většinou je jasný už z jejich názvu.

Jméno stylu v LyXu je uvedeno za slovem „Style“ na řádku, kterým celá definice začíná. První blok tedy definuje styl, který se bude jmenovat „Journal“

```

Style Journal
  LatexType      Command
  LatexName      journal
  Category       Clanek
  InTitle        1
end

```

Parametr `LatexType` určuje, jaký kód se bude při výstupu generovat. Nejčastěji se používá s hodnotou `Command` pro L^AT_EXové příkazy a s hodnotou `Environment` pro prostředí. Následující parametr `LatexName` určuje jméno L^AT_EXového příkazu či prostředí. Popisovaný styl se tedy váže k L^AT_EXovému příkazu `\journal`.

LyX seskupuje všechny načtené styly v nástrojové liště do kategorií. Jméno kategorie je zadáno jako hodnota parametru `Category`. V tomto příkladu jsou všechny styly sdruženy do kategorie `Clanek`.

Posledním parametrem v uvedené definici je `InTitle`. LyX automaticky vkládá titulní stránku za poslední styl, který ovlivňuje její obsah. Pro třídu *clanek* je nutné před vytvořením titulní stránky nadefinovat logo, nadpis a anglický nadpis, všechny odpovídající styly tedy musí mít parametr `InTitle` nastaven na hodnotu „1“.

Definice stylu „Journal“ je ukončena příkazem `end` na posledním řádku uvedeného kódu.

Mezi zajímavé definice stylů patří styly pro český a anglický titul. Část jejich definice bude shodná s předchozími styly, některé parametry ale budou nové.

```

Style Title
  LatexType      Command
  LatexName      title
  Category       Clanek
  InTitle        1
  Align          left
  AlignPossible  left
  Font
    Series       Bold
    Shape        Up
    Size         larger
  EndFont
  TopSep        1
  BottomSep     1
end

```

Prvním novým parametrem je `Align`. Hodnota tohoto parametru určuje výchozí zarovnání stylu. Podobnou funkci má i parametr `AlignPossible`, který udává všechny povolené způsoby zarovnání. Pokud jich je více, jsou odděleny čárkou.

Další skupina nových parametrů je uzavřena v bloku mezi příkazy `Font` a `EndFont`. Všechny tyto parametry ovlivňují písmo u definovaného stylu.

Poslední dvojice parametrů `TopSep` a `BottomSep` určuje horní a spodní odsazení celého bloku.

Hodnoty parametrů často opakují, naštěstí ale není nutné je u všech příbuzných stylů definovat znovu. Parametr `CopyStyle` určuje, z jakého existujícího stylu má nově definovaný styl vycházet. Toho je využito při definici stylu „`AltTitle`“, který je téměř shodný se stylem „`Title`“.

```
Style AltTitle
  CopyStyle      Title
  LatexName      alttitle
  Font
    Series       Medium
  EndFont
end
```

V uvedeném stylu jsou tedy změněny pouze vlastnosti, ve kterých se liší od kopírovaného stylu.

Následující styl nepoužívá žádné nepopsané vlastnosti, na rozdíl od předchozích stylů se však neváže k příkazu, ale k prostředí. Při převodu do \LaTeX u tedy pro něj bude vygenerován kód ohraničený příkazy `\begin` a `\end`.

```
Style Autor
  LatexType      Environment
  LatexName      authors
  Category      Clanek
  Align          left
  AlignPossible  left
  Font
    Series       Medium
    Shape        Up
    Size         normal
  EndFont
end
```

Pouze do stylů pro prostředí je možné v LyXu jednoduše vkládat další vnořené styly, proto také bylo toto L^AT_EXové makro definováno jako prostředí.

Styl „Pracovní zarazení“ slouží k sazbě pracovního zařazení v seznamu autorů. Pokud se jméno stylu skládá z více slov, je nutné je v definici uvádět v uvozovkách. Stejně pravidlo platí i pro situaci, kdy je takto vytvořeno jméno kopírovaného stylu. Všechny ostatní parametry tohoto stylu byly již popsány dříve.

```
Style "Pracovní zarazení"  
  LatexType      Command  
  LatexName      dept  
  Category       Clanek  
  Align          block  
  AlignPossible  block  
  Font  
    Series       Medium  
    Shape        Italic  
    Size         small  
  EndFont  
end
```

LyX umožňuje před a za každý styl přidávat popisy. Toho bude využito pro doplnění slova „ABSTRAKT“ před blok s českým abstraktem. Parametr `LabelType` určuje umístění popisu, parametr `LabelString` jeho obsah.

```
Style "Abstrakt (CS)"  
  LatexType      Environment  
  LatexName      abstrakt  
  Category       Clanek  
  Align          block  
  AlignPossible  block  
  Font  
    Size         small  
  EndFont  
  LabelType      Top_Environment  
  LabelString    "ABSTRAKT"  
  LabelFont  
    Series       Bold  
  EndFont  
  BottomSep     2  
end
```

Použitá hodnota `Top_Environment` zajistí, že se popis umístí nad levý horní roh stylu. Pokud by se v dokumentu vyskytovalo více bloků stejného stylu za sebou, zobrazí se popis jen nad prvním z nich.

Blok parametrů mezi příkazy `LabelFont` a `EndFont` určuje nastavení písma pro popis. Tento popis není součástí výsledného dokumentu, ale pouze se zobrazuje při editaci. Samotná sazba tohoto slova je v tomto případě zajištěna v definici příkazu `\abstrakt` v třídě `clanek`.

V definici stylu pro klíčová slova je zajímavá kombinace hodnot parametrů `Margin` a `LabelType`. Pokud má `Margin` hodnotu `Dynamic` a `LabelType` hodnotu `Static`, nastaví se levý okraj prvního řádku v bloku podle velikosti zadaného popisu, který je na první řádek automaticky vložen.

```
Style "Klicova slova (CS)"
  LatexType      Command
  LatexName      klicslova
  Category       Clanek
  Align          block
  AlignPossible  block
  Font
    Shape        Italic
    Size         small
  EndFont
  Margin         Dynamic
  LabelType      Static
  LabelString    "Klicova slova: "
  LabelFont
    Series       Bold
  EndFont
end
```

Tím byly popsány všechny parametry použité při tvorbě stylu pro LyX. Kompletní definice je uvedena v příloze a také uložena na příloženém CD.

5.7 Zpřístupnění šablony v LyXu

Před použitím šablony je nutné ji uložit do souboru, například `clanek.layout`, a pak ji nahrát do správného adresáře. Při výchozím nastavení se jedná o adresář:

```
C:\Program Files\LyX 1.6.2\Resources\layouts\
```

Dále je nutné z prostředí programu spustit funkci „Rekonfigurovat“ z menu „Nástroje“. Po dokončení rekonfigurace je potřeba LyX restartovat a při dalším spuštění je již nová šablona dostupná pro tvorbu dokumentů.

6 Závěr

Výstup z typografického systému $\text{T}_{\text{E}}\text{X}$ je již na první pohled hezčí než výstup z tradičních textových editorů. Program sice standardně nenabízí velké množství rodin písem, dodávaná písma jsou ale velmi kvalitní a na výsledné sazbě to má nemalý podíl. Nedostatek rodin písma však nemusí být vždy na škodu, protože ke kvalitní sazbě často stačí jen základní řezy jedné patkové a jedné bezpatkové rodiny.

Možnosti $\text{T}_{\text{E}}\text{X}$ u pro sazbu matematických výrazů nemají ani v dnešní době konkurenci. Každý běžný textový procesor obsahuje editor rovnic, ale pro kvalitní sazbu komplikovaných vzorců se nehodí. Vhodnost $\text{T}_{\text{E}}\text{X}$ u pro sazbu technických textů dokazuje i to, že je používán pro sazbu většiny vysokoškolských skript.

Balík maker $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ zvyšuje uživatelskou přívětivost sazby v $\text{T}_{\text{E}}\text{X}$ u a zpřístupňuje tak tento výborný systém širší veřejnosti. K jednoduchosti sazby v běžných textových editorech má sice ještě daleko, ale i tak se jedná o výrazné zjednodušení. Každý odborně zdatnější uživatel by měl být schopen sázet text v definované třídě dokumentů a za použití běžných rozšiřujících balíků.

Tvorba definice třídy dokumentu už je nad rámec možností běžného uživatele. Naštěstí existuje řada kvalitních tříd od $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ových odborníků, které jsou dodávány s podrobnou dokumentací.

Editor vektorové grafiky $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ Draw svojí funkčností pokrývá požadavky většiny běžných uživatelů a je snadno ovladatelný. Ostatně všechny vektorové obrázky v této práci jsou vytvořeny právě pomocí tohoto editoru. Jeho možnosti jsou výrazně vyšší, to ale pro potřeby této práce nebylo potřeba. Pro tvorbu komplikovaných grafů se sice nehodí, obdobná grafika ale bývá většinou výstupem jiného programu.

Vývojové prostředí $\text{T}_{\text{E}}\text{X}$ nicCenter významně usnadňuje práci se zdrojovým dokumentem. Navigace pomocí postranního panelu je intuitivní a značně urychluje práci s dokumentem. Tlačítka pro vkládání nejběžnějších příkazů a prostředí také práci velmi usnadňují.

Kromě zjednodušení editace zdrojového souboru je vývojové prostředí $\text{T}_{\text{E}}\text{X}$ nicCenter uživateli nápomocno i při překládání zdrojového souboru a při vyhledávání a odstraňování chyb. Definované výstupní profily zjednodušují změnu typů výstupního souboru, která je s nimi otázkou několika kliknutí myši. Pro hledání chyb jsou zase velmi užitečné funkce, které při označení chyby automaticky přesunou kurzor na odpovídající řádek zdrojového souboru.

WYSIWYM editor LyX je nejrozporupnější částí popisované publikační platformy. Práce s ním je sice jednodušší než přímé psaní příkazů $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u, bez toho se ale uživatelé LyXu občas neobejdou. Pro práci s běžně používanými základními třídami

dokumentů je LyX vhodný, při práci se speciálními třídami je spíš zbytečným mezičlánkem a znalosti L^AT_EXu se uživatel stejně nevyhne.

Cíle práce se podařilo splnit, publikační platforma založená na volně dostupném programovém vybavení byla definována a popsána. Zároveň byla vytvořena nová třída dokumentů pro L^AT_EX a definice stylů pro LyX. Záleží jen na potřebách a odborných znalostech uživatelů, zda je pro ně výhodné používat editor LyX, nebo tento článek produkčního řetězce vynechat a editovat přímo zdrojový kód dokumentu v programu T_EXnicCenter.

Všechny soubory, které byly vytvořeny při vzniku této práce, jsou uloženy na přiloženém CD. Všechny použité programy a rozšiřující balíky pro L^AT_EX jsou volně dostupné na internetu.

7 Seznam literatury

- [1] ADAPTIC *Typografie*. [online] Adaptic, s.r.o., [cit. 2009-01-08]. <<http://www.adaptic.cz/znalosti/slovnicek/typografie.htm>>. .
- [2] OETIKER, T. – PARTL, H. – HYNA, I. – SCHLEGL, E. – KOČER, M. – SÝKORA, V. *Ne příliš stručný úvod do systému \LaTeX 2_ε* [online] Verze 2.2-Beta, verze překladu CZ-0.9-Beta. [cit. 2009-01-13]. <<http://www.penguin.cz/~kocer/texty/lshort2e/lshort2e-cz.pdf>>. .
- [3] RYBIČKA, J. *\LaTeX pro začátečníky*. 3. vyd. Brno: Konvoj, c2002. ISBN 80-7302-049-1 .
- [4] KOPKA, H. – DALY, P. W. *Guide to \LaTeX* . 4. vyd. Addison-Wesley Professional, c2003. ISBN 03-2117-385-6 .
- [5] MITTELBACH, F. – GOOSSENS, M. – BRAAMS, J. – CARLISLE, D. – ROWLEY, C. *\LaTeX Companion*. 2. vyd. Addison-Wesley Professional, c2004. ISBN 02-0136-299-6 .
- [6] GOOSSENS, M. – MITTELBACH, F. – RAHTZ, S. – ROEGEL, D. – VOSS, H. *The \LaTeX Graphics Companion*. 2. vyd. Addison-Wesley Professional, c2007. ISBN 03-2150-892-0 .
- [7] ŠVAMBERG, M. *Jak na \LaTeX* . [online] Root.cz, 11. 9. 2003 [cit. 2008-09-01]. URL: <<http://www.root.cz/serialy/jak-na-latex/>>. ISSN 1212-8309 .
- [8] TEXNICCENTER.ORG *TeXnicCenter – the Center of your \LaTeX Universe* [online] TeXnicCenter.org , [cit. 2009-03-27]. <<http://www.texniccenter.org/>>. .
- [9] BLOUIN, A. *The \LaTeX Draw features* [online]. c2005-2006, poslední revize 15.11.2006. <<http://latexdraw.sourceforge.net/generalFeatures.html>>. .

A Výsledný dokument třídy *clanek*



Využití sázecího systému \LaTeX pro odborný časopis

Typesetting system \LaTeX used in professional journal

Jan Kropáč

Application Support Specialist, Hewlett-Packard s. r. o., Vyskočilova 1, Praha 4, 149 00

ABSTRAKT: Tato diplomová práce se zabývá možnostmi profesionální sazby textu v prostředí operačního systému Microsoft WindowsTM. V úvodních kapitolách krátce popisuje vznik a vývoj typografie do její současné podoby. Dále specifikuje základní typografická pravidla nutná ke správnému používání typografických programů. Praktická část této práce definuje programovou publikační platformu sestavenou z volně dostupných programů. Všechny použité programy spolupracují s balíkem maker \LaTeX , který zajišťuje výslednou sazbu pomocí typografického programu \TeX . Je vyvinuta nová třída dokumentů pro \LaTeX , která usnadňuje sazbu článků pro odborný časopis. K této třídě je dále vytvořena definice rozvržení pro program LyX. V závěru práce je uvedeno kritické zhodnocení možností navržené publikační platformy, třídy dokumentu pro \LaTeX a definice rozvržení pro LyX.

Klíčová slova: typografie, sazba, \TeX , \LaTeX , \MiKTeX , \TeX nicCenter, LyX

Jestliže se minulá století nazývala století páry a století elektřiny, lze bez nadsázky současné století nazvat stoletím informací či informačních technologií. Informační technologie se staly nedílnou součástí našeho každodenního života a málokdo by si dnes dokázal život bez nich představit.

Mohlo by se zdát, že tvorba dokumentů pro stránkovou média postupně ztrácí smysl, ale není tomu tak. Informace totiž nemají jen svoji obsahovou část, ale mají také svoji formu. U elektronické pošty či aktuálních zpráv samozřejmě převažuje aktuálnost a informační hodnota nad vzhledem, ale například u knih tomu tak není. Tvorba textových dokumentů se dnes stala nedílnou součástí našeho života.

Typografie je věda, která existuje několik set let. Za dobu své existence se vyvinula a všechna její pravidla mají svá opodstatnění. Cílem této práce je popsat a vysvětlit základy typografie, navrhnout programovou platformu pro počítačové publikování a vysvětlit její použití. Jako příklad počítačové publikace bude sloužit odborný časopis.

Pro správné pochopení popisované problematiky jsou nejprve zavedeny některé odborné termíny z oblasti typografie a základní typografická pravidla. Tato pravidla jsou popisována bez vazby na konkrétní typografický program, protože se jedná o univerzálně platná pravidla. Jestliže se některé pravidlo týká jen sazby v českém jazyce, je na to v textu upozorněno.

Dále jsou popsány základy správné sazby v balíku maker \LaTeX . Tento popis obsahuje základy nutné pro sazbu textu. Dále obsahuje popis nejběžnějších prostředí, jako jsou *enumerate*, *itemize*, *tabular*, *verbatim* a podobně. Jedna podkapitola je věnována použití plovcích prostředí *table* a *figure* a tvorbě křížových odkazů.

Praktická část práce je zaměřena na definici programového vybavení vhodného pro publikování v \TeX u. Tato množina programů se skládá z distribuce \LaTeX u

\MiKTeX , vektorového grafického editoru \LaTeX Draw, vývojového prostředí pro práci se zdrojovými dokumenty \TeX nicCenter a WYSIWYM editoru LyX. Možnosti jednotlivých programů jsou popsány v samostatných podkapitolách.

V dalších podkapitolách praktické části je vytvořena nová třída dokumentů pro \LaTeX a k ní odpovídající styl (*layout*) pro LyX. Postup vytváření třídy dokumentů je doprovázen výkladem o použitých příkazech a doplňkových balících makech, které jsou v komentovaných definicích použité.

Závěr praktické části je věnován popisu vytvořeného stylu pro LyX. Stejně jako u třídy dokumentu i zde je celý postup doprovázen vysvětlením použitých příkazů.

Cíle práce se podařilo splnit, publikační platforma založená na volně dostupném programovém vybavení byla definována a popsána. Zároveň byla vytvořena nová třída dokumentů pro \LaTeX a definice stylů pro LyX. Záleží jen na potřebách a odborných znalostech uživatelů, zda je pro ně výhodné používat editor LyX, nebo tento článek produkčního fetezce vynechat a editovat přímo zdrojový kód dokumentu v programu \TeX nicCenter.

Použitá literatura

- [1] OETIKER, T. – PARTL, H. – HYNA, I. – SCHLEGL, E. – KOČER, M. – SÝKORA, V. *Ne přidiš stručný úvod do systému $\LaTeX 2_{\epsilon}$* [online]. Verze 2.2-Beta, verze překladu CZ-0.9-Beta. [cit. 2009-01-13]. <<http://www.penguin.cz/kocer/texty/lshort2e/lshort2e-cz.pdf>>.
- [2] RYBIČKA, J. *\LaTeX pro začátečníky*. 3. vyd. Brno: Konvoj, c2002. ISBN 80-7302-049-1

Obrázek 9: Dokument třídy *clanek*, strana 1

- [3] 3 KOPKA, H. – DALY, P. W. *Guide to L^AT_EX*.
4. vyd. Addison-Wesley Professional, c2003. ISBN
03-2117-385-6
- [4] 4 MITTELBACH, F. – GOOSSENS, M. –
BRAAMS, J. – CARLISLE, D. – ROWLEY,
C.
L^AT_EX Companion.
2. vyd. Addison-Wesley Professional, c2004. ISBN
02-0136-299-6
- [5] 5 GOOSSENS, M. – MITTELBACH, F. –
RAHTZ, S. – ROEGEL, D. – VOSS, H.
The L^AT_EX Graphics Companion.
2. vyd. Addison-Wesley Professional, c2007. ISBN
03-2150-892-0

ABSTRACT: The main goal of this master thesis is a description of typesetting possibilities on Microsoft WindowsTM operating system. At the beginning there is a short historical review of typesetting beginnings and its evolution till it's current state. The basic typographical terms and principles are specified, as it is essential for understanding and proper use of typesetting programs. Practical part of this thesis is focused on defining program platform for computer publishing, that is based on free software. All used programs collaborate together with the package of typesetting macros L^AT_EX, which uses T_EX software for the final output. New L^AT_EX document class is being developed in order to simplify professional journal article typesetting. For this new class There is also a new layout for LyX program being developed. The conclusion of this thesis contains the critical evaluation of proposed publication platform, of L^AT_EX document class and LyX document layout.

Keywords: *typography, typesetting, T_EX, L^AT_EX, MiK_TE_X, T_EXnicCenter, LyX*

Kontaktní adresa:

Jan Kropáč
Tatarkova 731, Praha 4 - Háje, 149 00
jan.kropac@hp.com
<http://www.kropac.net>

Diplomová práce

2

Obrázek 10: Dokument třídy *clanek*, strana 2

B Definice třídy dokumentu pro L^AT_EX

```
%-----< identification >-----
\NeedsTeXFormat{LaTeX2e}
\ProvidesClass{clanek}[2009/03/01 Ukazkova trida dokumentu]

%-----< initial code >-----
\RequirePackage{multicol}
\RequirePackage{graphicx}
\RequirePackage{calc}
\RequirePackage[a4paper]{geometry}[2008/12/21]
\RequirePackage{fancyhdr}
\RequirePackage[cp1250,latin2]{inputenc}
\RequirePackage[T1]{fontenc}
\RequirePackage[czech,english]{babel}

%-----< declaration of options >-----
\PassOptionsToClass{onecolumn,10pt}{article}
\DeclareOption*{\PassOptionsToClass{\CurrentOption}{article}}

%-----< execution of options >-----
\ProcessOptions\relax

%-----< class loading >-----
\LoadClass{article}

%-----< main code >-----
%--- nastavení rozměrů stránky
\geometry{nohead,margin=1.8cm,bindingoffset=0.3cm}

%--- nastavení záhlaví a zápatí
\pagestyle{fancy}
\fancyhf{}
\renewcommand{\headrulewidth}{0pt}
\renewcommand{\footrulewidth}{0pt}
\fancyfoot[RO,LE]{\small\thepage}
\newcommand{\journal}[1]{\fancyfoot[LO,RE]{\small#1}}
```

```

%--- nová makra pro zadání loga a anglického titulu
\newcommand{\logo}[1]{\def\@logo{#1}}
\newcommand{\altitle}[1]{\def\@altitle{#1}}
\altitle{}
\logo{}

%--- záloha původního makra \maketitle
\let\oldtitle = \maketitle

%--- definice nového makra \maketitle
\renewcommand\maketitle{
  \selectlanguage{czech}

  \ifx\@logo\@empty\relax\else
    \noindent{\includegraphics[height=1in]{\@logo}}\par
  \fi

  \noindent
  \begin{flushleft}\Large\bfseries\@title\end{flushleft}

  \ifx\@altitle\@empty\relax\else
    \selectlanguage{english}
    \par\par
    \noindent\begin{flushleft}\Large\@altitle\end{flushleft}
    \selectlanguage{czech}
  \fi
}

%--- definice prostředí pro sazbu jména autora a jeho zařazení
\newenvironment{authors}{
  \par\noindent\bfseries
}{
  \vspace{0.1em}
}

%--- definice makra pro sazbu pracovního zařazení autora
\newcommand{\dept}[1]{
  \vspace{0.1em}
  \par\noindent\normalfont{\itshape{#1}}
}

```

```

%--- začátek abstraktu
\newcommand\abstropen[1]{
  \noindent\tabular{@{}p{\textwidth}@{}}
  \hline
  \small
  {\bfseries{#1:}}
}

%--- konec abstraktu
\newcommand\abstrclose{
  \\ \hline
  \endtabular
}

%--- prostředí pro sazbu českého abstraktu
\newenvironment{abstrakt}{%
  \abstropen{ABSTRAKT}
}{
  \abstrclose
}

%--- prostředí pro sazbu anglického abstraktu
\renewenvironment{abstract}{
  \selectlanguage{english}
  \abstropen{ABSTRACT}
}{
  \abstrclose
  \selectlanguage{czech}
}

%--- makra pro zadání klíčových slov (cs/en)
\newcommand\klicslova[1]{\kwrds{Klíčová~slova}{#1}}
\newcommand\keywords[1]{\kwrds{Keywords}{#1}}

%--- makro pro sazbu klíčových slov
\newcommand{\kwrds}[2]{
  \\ \footnotesize
  \itshape
  {\bfseries#1:}~#2
}

```



```

%--- prostředí pro sazbu ve dvou sloupcích
\newenvironment{clanek}{
  \begin{multicols}{2}
}{
  \end{multicols}
}

%--- prostředí pro sazbu kontaktní adresy
\newenvironment{address}{
  \vfill
  \noindent\tabular{@{}p{\textwidth}@{}}
  \hline
  \vspace{0.1em}
  \small
  {\itshape{Kontaktní adresa:}}\\
}{
  \\
  \hline
  \endtabular
}

%--- číslování sekcí
\setcounter{secnumdepth}{0}

%--- záloha původních prostředí
\let\oldtable=\table
\let\oldfigure=\figure

%--- Změna chování znaku @
\makeatletter

%--- Plovoucí tabulka v sloupcích
\renewenvironment{table} {
  \def\@capytype{table}
  \vspace{0.2em}
}{
  \vspace{0.2em}
}

```

```
%--- Plovoucí figure v sloupcích
\renewenvironment{figure}{
  \def\@capttype{figure}
  \vspace{0.2em}
}{
  \vspace{0.2em}
}

%--- Změna chování znaku @
\makeatother
```

C Šablona pro LyX

```

%% Do not delete the line below; configure depends on this
# \DeclareLaTeXClass[clanek]{article (Casopis)}

# zakladni definice
Input stdclass.inc

Style Journal
  LatexType      Command
  LatexName      journal
  Category       Clanek
  InTitle        1
end

Style Logo
  LatexType      Command
  LatexName      logo
  Category       Clanek
  InTitle        1
end

Style Title
  LatexType      Command
  LatexName      title
  Category       Clanek
  InTitle        1
  Align          left
  AlignPossible  left
  Font
    Series       Bold
    Shape        Up
    Size         larger
  EndFont
  TopSep        1
  BottomSep     1
end
```

```

Style AltTitle
  CopyStyle      Title
  LatexName      alttitle
  Font
    Series       Medium
  EndFont
end

Style Autor
  LatexType      Environment
  LatexName      authors
  Category       Clanek
  Align          left
  AlignPossible  left
  Font
    Series       Medium
    Shape        Up
    Size         normal
  EndFont
end

Style "Pracovni zarazeni"
  LatexType      Command
  LatexName      dept
  Category       Clanek
  Align          block
  AlignPossible  block
  Font
    Series       Medium
    Shape        Italic
    Size         small
  EndFont
end

```

```

Style "Abstrakt (CS)"
  LatexType      Environment
  LatexName      abstrakt
  Category       Clanek
  Align          block
  AlignPossible  block
  Font
    Size         small
  EndFont
  LabelType      Top_Environment
  LabelString    "ABSTRAKT"
  LabelFont
    Series       Bold
  EndFont
  BottomSep     2
end

Style "Abstrakt (EN)"
  CopyStyle      "Abstrakt (CS)"
  LatexName      abstract
  LabelString    "ABSTRACT"
end

Style "Klicova slova (CS)"
  LatexType      Command
  LatexName      klicslova
  Category       Clanek
  Align          block
  AlignPossible  block
  Font
    Shape        Italic
    Size         small
  EndFont
  Margin         Dynamic
  LabelType      Static
  LabelString    "Klicova slova: "
  LabelFont
    Series       Bold
  EndFont
end

```

```

Style "Klicova slova (EN)
  CopyStyle      "Klicova slova (CS)"
  LatexName      keywords
  LabelString    "Keywords: "
end

Style "Telo clanku"
  LatexType      Environment
  LatexName      clanek
  Align          block
  AlignPossible  block
end

Style "Kontaktni adresa"
  LatexType      Environment
  LatexName      address
  Category       Clanek
  Align          left
  AlignPossible  left,block
  Font
    Size         small
  EndFont
  LabelType      Top_Environment
  LabelString    "Kontaktni adresa:"
  LabelFont
    Shape        Italic
  EndFont
end

```

D Seznam obrázků

1	Rozměry písma	10
2	Kerning	14
3	Ligatura	14
4	Minkowského nerovnost	28
5	Grafický instalátor MiKTeXu verze 2.7	32
6	Grafický editor L ^A T _E XDraw	33
7	Prostředí editoru T _E XnicCenter	34
8	WYSIWYM Editor LyX	35
9	Dokument třídy <i>clanek</i> , strana 1	57
10	Dokument třídy <i>clanek</i> , strana 2	58

E Seznam tabulek

1	Řezy písma	11
2	Třídy dokumentů	19
3	Rodiny písma	22
4	Velikost písma	22
5	Řezy písma v L ^A T _E Xu	23
6	Ukázková tabulka	27