

Mendelova univerzita v Brně  
Provozně ekonomická fakulta

---

# Nové možnosti tvorby tabulek v systému $\text{T}_{\text{E}}\text{X}/\text{L}\text{A}\text{T}_{\text{E}}\text{X}$

Diplomová práce

Vedoucí práce:  
Ing. Jan Přichystal, Ph.D.

Bc. Josef Jeniš

Brno 2015

Tady bude puvodni zadání BP...

Chtěl bych poděkovat mému vedoucímu práce Ing. Janu Přichystalovi Ph.D. za vedení této práce a za poskytnutí podkladů, které byly nezbytné pro její zpracování. Dále děkuji za poskytnutí cenných rad, pomoci a trpělivost při vedení práce. Mé poděkování patří také panu RNDr. Petru Olšákovi za jeho systém OPmac, který je veřejně přístupný a za jeho pomoc.

Také bych chtěl poděkovat mé rodině, že mi umožnili studovat univerzitu a podpořovali mě při tom. Hlavně mě ale občas omluvili a uvolnili ze sezónních prací, abych se mohl věnovat studiu.

Také bych chtěl poděkovat Hospodinu za to, že mi dodával energii a sílu pro studium a práci, kterou jsem musel zvládnout najednou.

### Čestné prohlášení

Prohlašuji, že jsem tuto práci: **Nové možnosti tvorby tabulek v systému  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$**

vypracoval samostatně a veškeré použité prameny a informace jsou uvedeny v seznamu použité literatury. Souhlasím, aby moje práce byla zveřejněna v souladu s § 47b zákona č. 111/1998 Sb., o vysokých školách ve znění pozdějších předpisů, a v souladu s platnou *Směrnicí o zveřejňování vysokoškolských závěrečných prací*.

Jsem si vědom, že se na moji práci vztahuje zákon č. 121/2000 Sb., autorský zákon, a že Mendelova univerzita v Brně má právo na uzavření licenční smlouvy a užití této práce jako školního díla podle § 60 odst. 1 Autorského zákona.

Dále se zavazuji, že před sepsáním licenční smlouvy o využití díla jinou osobou (subjektem) si vyžádám písemné stanovisko univerzity o tom, že předmětná licenční smlouva není v rozporu s oprávněnými zájmy univerzity, a zavazuji se uhradit případný příspěvek na úhradu nákladů spojených se vznikem díla, a to až do jejich skutečné výše.

V Brně dne 20.05.2015

.....

**Abstract**

Josef Jeniš, New options of table creation in  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  system, Diploma thesis, Brno:MENDELU PEF,2014

This diploma thesis deals with issue of table rates in systems  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , analysis of current state and current packages. Within this thesis was created the new package for table rates, which is based on analysis and requirements for such type of package, which is described in diploma thesis of Ing. Petra Talandová Ph.D. – Approaches to table processing in DTP systems.

**Abstrakt**

Josef Jeniš, Nové možnosti tvorby tabulek v systému  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , Diplomová práce. Brno:MENDELU PEF,2014

Tato diplomová práce se zabývá problematikou sazby tabulek v systémech  $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , analýzou současného stavu a dostupných balíčků. V rámci této práce je vytvořen nový balík pro sazbu tabulek, který vychází z analýzy a požadavků na takový balík, které popisuje Ing. Petra Talandová Ph.D. ve své diplomové práci Přístupy ve zpracování tabulek v systémech DTP.

## Obsah

<b>1</b>	<b>Úvod a cíl práce</b>	<b>7</b>
1.1	Úvod . . . . .	7
1.2	Cíl práce . . . . .	8
<b>2</b>	<b>Metodika</b>	<b>9</b>
<b>3</b>	<b>Teoretická část</b>	<b>10</b>
3.0.1	$\text{T}_{\text{E}}\text{X}$ . . . . .	10
3.0.2	$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . . . . .	10
3.0.3	$\text{T}_{\text{E}}\text{X}$ onWeb . . . . .	11
3.0.4	HTML (Hypertext markup language) . . . . .	11
3.1	Sazba tabulek . . . . .	11
3.1.1	Pořadová sazba . . . . .	12
3.1.2	Části tabulky . . . . .	12
3.1.3	Vedení a křížení linek . . . . .	12
3.1.4	Rozdělení tabulek . . . . .	12
3.2	Analýza současného stavu tvorby tabulek . . . . .	14
3.2.1	Prostředí tabbing . . . . .	14
3.2.2	Prostředí tabular . . . . .	16
3.2.3	Prostředí tabularx . . . . .	17
3.2.4	Balíky pro vícestránkové tabulky . . . . .	17
3.2.5	OPmac . . . . .	18
3.3	Příklad sazby tabulek v různých systémech . . . . .	18
3.3.1	LibreOffice Writer . . . . .	18
3.3.2	HTML . . . . .	19
3.3.3	$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ . . . . .	21
<b>4</b>	<b>Požadavky na nový balík</b>	<b>23</b>
4.1	Umístění tabulky . . . . .	23
4.2	Ohraničení tabulky . . . . .	23
4.3	Výška a šířka . . . . .	24
4.4	Práce se skupinami polí . . . . .	24
4.5	Mezery v tabulce . . . . .	24
4.6	Zarovnání v tabulce . . . . .	24
4.7	Barvy . . . . .	25
4.8	Ostatní vlastnosti . . . . .	25
<b>5</b>	<b>Analýza tvorby tabulek v systému Plain<math>\text{T}_{\text{E}}\text{X}</math></b>	<b>26</b>
5.1	Makro $\backslash$ leaders . . . . .	26
5.2	Makra pro tvorbu tabulky s pevnou šířkou sloupců . . . . .	27
5.3	Tvorba tabulky pomocí $\backslash$ halign . . . . .	28

<b>6</b>	<b>Popis implementovaného stylu</b>	<b>31</b>
6.1	Základní příkazy . . . . .	31
6.1.1	<code>\table</code> . . . . .	31
6.1.2	<code>\tablebody</code> . . . . .	33
6.1.3	<code>\endtable</code> . . . . .	35
6.2	Rozšiřující makra pro další nastavení tabulky . . . . .	36
6.2.1	<code>\bordertype</code> . . . . .	36
6.2.2	<code>\striped</code> . . . . .	37
6.2.3	Další příkazy pro nastavení . . . . .	38
6.3	Obsah těla tabulky . . . . .	39
6.3.1	<code>\title</code> . . . . .	39
6.3.2	<code>\tr</code> . . . . .	39
6.3.3	<code>\trwithline</code> . . . . .	40
6.3.4	<code>\trwithcolor</code> . . . . .	41
6.3.5	<code>\td</code> . . . . .	41
6.3.6	<code>\th</code> . . . . .	41
6.3.7	<code>\tdalign</code> . . . . .	42
6.3.8	<code>\colspan</code> . . . . .	43
6.3.9	<code>\rowspan</code> . . . . .	43
6.3.10	<code>\trcustom</code> . . . . .	44
6.4	Další definovaná makra . . . . .	46
6.4.1	<code>\longtable</code> . . . . .	46
6.4.2	<code>\spacesaroundtable</code> . . . . .	48
6.4.3	<code>\tablecaptions</code> . . . . .	48
<b>7</b>	<b>Ukázka použití</b>	<b>49</b>
7.1	Jednoduchá tabulka . . . . .	49
7.2	Tabulka s horizontálními linkami . . . . .	50
7.3	Pruhovaná tabulka . . . . .	50
7.4	Tabulka s definovanou šířkou . . . . .	52
7.5	Tabulka se sloučenými buňkami . . . . .	53
7.6	Dlouhá tabulka přes více stran . . . . .	54
<b>8</b>	<b>Diskuze a hodnocení</b>	<b>57</b>
8.1	Hodnocení práce a návrhy pro zlepšení . . . . .	57
8.2	Porovnání se stávajícími balíky . . . . .	58
<b>9</b>	<b>Závěr</b>	<b>60</b>
<b>10</b>	<b>Literatura</b>	<b>61</b>
	<b>Přílohy</b>	<b>62</b>
<b>A</b>	<b>Srovnání kódu</b>	<b>63</b>

OBSAH	8
<b>B Uživatelská dokumentace v anglickém jazyce</b>	<b>64</b>
<b>C Přiložené CD</b>	<b>72</b>



# 1 Úvod a cíl práce

## 1.1 Úvod

Technologie se neustále vyvíjejí. A i přesto, že tyto nové technologie jsou čím dál více sofistikované, jejich samotné ovládání se spíše zjednodušuje. Ovládání počítače a práce s ním už se tedy netýká pouze specialistů v tomto oboru. Skoro každý člověk dnes již umí ovládat počítač a mnoho lidí vlastní přímo svůj počítač, tablet nebo smartphone, nebo je alespoň využívá v zaměstnání. Vzniká tak obrovské množství dat. Mezi tyto data se řadí také dokumenty.

Různé dokumenty dnes existují a vznikají prakticky v každém zaměstnání a odvětví. Ať už se jedná například o definice pracovních postupů, různá hlášení a reporty, hospodaření skladů, nebo to mohou být například faktury a dodací listy. Jako dokumenty lze definovat také odborné práce, diplomové práce, knihy a další literaturu. A v převážné většině těchto dokumentů lze nalézt nějakou tabulku nebo výčet. Tyto jsou hojně používané pro zobrazování důležitých informací a výsledků hlavně díky své přehlednosti.

Jak již bylo řečeno, tvorbu těchto dokumentů už obstarávají stejní lidé, kteří jsou zodpovědní za vytvoření obsahové části dokumentu, tudíž jsou odborníci na jiný obor a mnohdy nemají přílišnou představu o pravidlech sazby dokumentů, respektive tabulek. Koncept, kdy obsahovou část dokumentu vytvořil odborník na daný obor a typografii a sazbu dokumentu obstarával někdo jiný, se již dlouho nepoužívá. Také možnosti sazby textu a tabulek pokročily. Dnes již existuje mnoho kancelářských balíčků pro tvorbu textu a některé z nich obsahují nástroje pro jednoduchou sazbu tabulek (např. MS Office, Open Office apod.). Kde snahou bylo co nejvíce zpříjemnit a zjednodušit uživatelům práci s textem. Většina těchto nástrojů umožňuje uživatelům graficky vytvářet a upravovat tabulky.

Jeden z využívaných systémů je systém  $\text{\TeX}$  a jeho různé varianty ( $\text{\LaTeX}$ ,  $\text{\XeTeX}$  apod.). Tento systém ze své podstaty nedisponuje možností graficky vytvářet a modifikovat tabulky. Existuje sice různý software, který obsahuje grafické rozhraní pro práci s tabulkami v těchto systémech (např.  $\text{\TeXnic}$ ), ale ve většině případů slouží pouze na vytvoření základní „šablony“ tabulky (definování určitého počtu sloupců a řádků a vysázení příkazů). Další práce s touto tabulkou už je na úrovni psaní příkazů a obsahu tabulky. Pro sazbu tabulek v různých distribucích systému  $\text{\TeX}$  je vytvořeno relativně hodně balíčků. Ale práce s nimi je mnohdy obtížná a navíc nemusí obsahovat příkazy pro všechny potřebné vlastnosti tabulek a je tedy zapotřebí zároveň s nimi použít další balíčky. V takových případech se ale může stát, že tyto balíčky nejsou spolu kompatibilní a využití jednoho znamená znemožnění využít jiný balíček.

## 1.2 Cíl práce

Cílem této práce je vytvoření nového komplexního balíku pro sazbu tabulek, který bude pokrývat základní požadavky na jejich sazbu co se funkcionality a typografických pravidel týče. Definovaná makra nově vytvořeného balíku musí mít snadné a intuitivní použití, které nemusí být podobné příkazům používaným ve stávajících balících. Dílčím cílem práce je analyzovat stav aktuálně dostupných balíků pro sazbu tabulek v systému  $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$  a vytyčit jejich výhody a nevýhody.

## 2 Metodika

Cílem práce je vytvořit balík pro sazbu tabulek, který bude snadný na používání s tím, že jeho použití a definovaná makra mohou být poněkud odlišné od současných a zavedených konvencí.

V první řadě je tedy zapotřebí analyzovat typografická pravidla pro sazbu tabulek a tyto pravidla zohlednit ve vytvořeném stylu. Tato analýza je jedna ze základních, ale také náročnějších částí práce, protože o typografii a sazbě tabulek neexistuje příliš mnoho literatury.

Dalším krokem je analýza současných balíků pro tvorbu tabulek v systému  $\LaTeX$ . Jejich funkcí, výhody a nevýhody, způsob použití a zápisu tabulek. Přehlednost zápisu tabulek je velice důležitý, protože jeden z hlavních cílů práce je usnadnit a zpřehlednit zápis tabulek a jejich vlastností.

Současně s analýzou existujících balíků pro systém  $\LaTeX$  a jejich způsobu vytváření tabulek musí proběhnout také analýza způsobu tvorby tabulek a zápisu v některých dalších systémech, které se používají pro sazbu textu, respektive tabulek a jejich porovnání. Pro tyto potřeby byl vybrán systém LibreOffice Writer, který je velice podobný systému Word od firmy Microsoft a jeho přístup k tabulkám bude velice podobný, a značkovací jazyk HTML, který je už více podobný systému  $\TeX$ .

Následuje analýza a vytvoření požadavků funkcí, které by měl nový balík umět, aby byl plnohodnotným balíkem pro sazbu tabulek a nepotřeboval již více úprav. Tato analýza bude postavena na diplomové práci Ing. Petry Talandové Ph.D. – Přístupy ve zpracování tabulek v systémech DTP, která se touto analýzou zabývá a má ji velice dobře zpracovanou.

Další důležitý krok je vybrat systém, na kterém bude balík postaven. Zde existuje možnost vybrat systém  $\TeX$ , respektive Plain  $\TeX$ , který obsahuje cca 300 základních primitiv se kterými se dá dále pracovat. Další možností je využít již existující balíky, které jsou vytvořeny pro systémy  $\LaTeX$ , XeTeX apod. Tato možnost se neukázala jako reálná, neboť současné balíky mohou mít problémy s kompatibilitou jednoho balíku s ostatními apod. Byl tedy vybrán systém Plain  $\TeX$ .

Následující krok vychází z předchozího a to je analýza možností sazby tabulek v PlainTeXu a celkově vytváření maker v PlainTeXu, se kterými nemá autor příliš zkušeností. Vyhodnocení těchto možností a výběr jedné, na které bude balík postaven. Následuje implementace a vytvoření stylu, dokumentace a zhodnocení výsledků.

## 3 Teoretická část

### 3.0.1 T<sub>E</sub>X

Typografický systém T<sub>E</sub>X vytvořil koncem 70. let americký profesor informatiky Donald E. Knuth a to proto, že nebyl spokojen se sazbou jedné ze svých knih. Rozhodl se proto vytvořit nový systém, který bude text sázet pořádně, včetně složitých matematických vzorců.

T<sub>E</sub>X se řadí do skupiny tzv. značkovacích jazyků a dal by se charakterizovat také jako programovací jazyk pro sazbu textů. Vstupním souborem je textový soubor, který obsahuje jak sázený text, tak příkazy ovlivňující sazbu.

T<sub>E</sub>X se zprvu striktně řadil mezi systémy, které nepatří do skupiny WYSIWYG<sup>1</sup>. Systémy, respektive textové editory skupiny WYSIWYG jsou takové, kde uživatel při editaci textu rovnou vidí, jak bude vypadat výsledný dokument, až ho například vytiskne, nebo uloží do PDF. Je to například systém Microsoft Word, OpenOffice.org apod. Uživatel v těchto systémech formátuje text nejčastěji pomocí grafického rozhraní a myši.

Oproti tomu sázení textu v systémech, které nejsou WYSIWYG, probíhá pomocí příkazů a značek. Existují sice editory, které dokáží ulehčit uživateli práci a ten si může pomocí myši „naklikat“ některé formátování, tyto ale pouze vypíší značky do textu za uživatele.

Původní verze T<sub>E</sub>X při zpracování vstupního dokumentu vytvořil soubor typu DVI<sup>2</sup>, který obsahoval jeho vysázenou podobu, nikoli však konkrétní tvary jednotlivých znaků. Tento soubor je abstraktní a obsahuje pouze informace typu „na souřadnicích (x,y) se nachází znak znak Q o velikosti  $v$  sázený písmem  $p$ “. To proto, aby byl systém T<sub>E</sub>X nezávislý na platformě a dokumenty vytvořené v tomto systému měly stejnou podobu například i po padesáti letech. V současnosti se ale využívá praktičtější přístup a to jsou různé implementace T<sub>E</sub>Xu, které na výstup generují přímo soubory ve formátu PDF.

Tvorba dokumentu pouze pomocí systému T<sub>E</sub>X, který má cca 300 vestavěných (tzv. primitivních) příkazů, by ale byla velice náročná. Vznikají tedy takzvaná makra, která dokáží ulehčit uživateli práci. Knuth vytvořil sadu takových maker pod názvem PlainT<sub>E</sub>X. (Satrapa, P, 2011)

### 3.0.2 L<sup>A</sup>T<sub>E</sub>X

L<sup>A</sup>T<sub>E</sub>X je sada maker pro T<sub>E</sub>X, kterou vytvořil Leslie A. Lamport. Snažil se tímto vyjít vstříc běžným potřebám při sazbě dokumentů. Tato sada proto obsahuje makra pro členění textu do kapitol, generování obsahu, vkládání obrázků apod. L<sup>A</sup>T<sub>E</sub>X spolu s PlainT<sub>E</sub>Xem v současné době patří mezi nejpoužívanější modifikaci původního systému. Oproti PlainT<sub>E</sub>Xu se ale L<sup>A</sup>T<sub>E</sub>X neustále vyvíjí a vznikají pro něj stále nové a nové balíky. (Satrapa, P, 2011)

<sup>1</sup>„What you see is what you get – co vidíš, to dostaneš“

<sup>2</sup>DeVice Independent

### 3.0.3 T<sub>E</sub>XonWeb

T<sub>E</sub>XonWeb je projekt vytvořený Ing. Janem Přichystalem Ph.D., který umožňuje využití systému T<sub>E</sub>X/L<sup>A</sup>T<sub>E</sub>X pouze za pomoci webového prohlížeče. Projekt vznikl za účelem zjednodušit a nabídnout nezkušeným uživatelům práci s kvalitním typografickým systémem tak, aby si nemuseli zdlouhavě, a někdy složitě instalovat a nastavovat fungující prostředí a zároveň dostupnost tohoto prostředí prakticky odkudkoliv.

Projekt je založen na TeXLive 2013 z distribuce Linux CentOS 5.2. Implicitní kódování je UTF-8 a pro vytvoření projektu byl použit skriptovací jazyk Perl spolu s JavaScriptem a knihovnou JQuery.

Pro registrované a přihlášené uživatele nabízí tento projekt mimo jiné také grafický návrhář tabulek podobný jako v systému Word. Zde si uživatel může definovat počet sloupců a řádků tabulky, ohraničení tabulky, barvy a obsah buněk. Návrhář poté vytvoří zdrojový kód této tabulky. Tento návrhář aktuálně využívá LaTeXový balík pro sazbu tabulek tabular. Tato funkce aktuálně umožňuje pouze převod z grafické podoby do textové. Jakmile uživatel využije návrhář pro vytvoření zdrojového kódu, už není možné tuto tabulku znovu zobrazit v grafické podobě, stejně tak, pokud uživatel přenesou tabulku z jiného zdroje. A to kvůli nepřehlednosti a náročnosti zdrojového kódu prostředí tabular. Proto je cílem této práce vytvořit takový balík, který by kód zjednodušil a tímto umožnil převod kódu do grafické podoby. (Přichystal,J)

### 3.0.4 HTML (Hypertext markup language)

Značkovací jazyk HTML byl primárně navržen pro sémantický popis vědeckých dokumentů, nicméně jeho obecný návrh a adaptace během dalších let rozšířil spektrum jeho použití také na popis mnoho typů dalších dokumentů.

Jazyk HTML vznikl v roce 1990 v zařízení zvaném CERN<sup>3</sup> a to kvůli jednoduché sazbě vědeckých dokumentů. Prvních pět let si prošel tento jazyk mnoha revizemi a změnami. Po zrození W3C<sup>4</sup> se vývoj relativně standardizoval. V praxi to ale vypadalo spíše tak, že vývoj jazyka byl závislý na vývojářích webových prohlížečů. V současné době je nejnovější verze HTML 5.

Jazyk HTML je ve své podstatě velice podobný systému T<sub>E</sub>X a to tím, že dokument obsahuje text a značky (tzv. tagy), pomocí kterých se tento text formátuje. (HTML 5.1 Nightly)

## 3.1 Sazba tabulek

Sazba tabulek je sazba textu a číslic přehledně upravená linkami, která navazuje na sazbu pořadovou. Samotná sazba tabulek je náročná a vyžaduje od sazeče všestrannou odbornost a navíc cit pro estetiku z hlediska celkové grafické úpravy. Základní

<sup>3</sup>výzkumné fyzikální centrum ve Švýcarsku

<sup>4</sup>World Wide Web Consortium – konsorcium pro webové standardy pro World Wide Web

hledisko úpravy všech tabulek je ale přehlednost a optimální rozdělení „světla“ v jednotlivých částech tabulky. (Pop, P, 1989)

### 3.1.1 Pořadová sazba

Pořadová sazba se využívá pro uspořádání textu do přehledného pořadí pod sebou v samostatných řádcích nebo do sloupců vedle sebe. Pokud je text na více řádcích, tyto řádky se pro lepší přehlednost a estetický vzhled odrážejí čtverčíkem. Sázejí se tímto knižní obsahy, jmenné, místní pojmové seznamy, apod. (Pop, P, 1989)

### 3.1.2 Části tabulky

Součástí většiny tabulek je nějaký nadpis, který může obsahovat např. důležité informace o obsahu tabulky, nebo může osvětlovat, o čem obsah tabulky vypovídá apod. Toto se nazývá *záhlaví tabulky*. Šířka základního textu je shodná se šířkou tabulky. Někdy bývají tyto nadpisové řádky uzavřeny linkovým rámcem a tvoří s tabulkou jednotný celek. (Pop, P, 1989)

Samotná tabulka se poté skládá z *rámečku tabulky*, jehož součástí je také *základní linka*. Dále je to *hlavička* – horní část tabulky s textem, který je rozdělen svislými nebo příčnými linkami na tzv. oka.

*Hlavička tabulky* je oddělena od dolní části tabulky vodorovnou linkou zvanou *linka pod hlavičkou*. Část tabulky pod hlavičkou se nazývá *nohy* tabulky a obvykle je rozdělena svislými linkami na *kolonky* (někdy také *rubriky*).

V nohách tabulek určených pro vpisování jsou zpravidla jednotné mezery na vpisované řádky členěny vodorovnými (příčnými) linkami – tato část tabulky se poté nazývá *příčnick*, viz obr.1.(Pop, P, 1989)

### 3.1.3 Vedení a křížení linek

Vedení a křížení linek se v tabulkách řídí přesnými pravidly a to: „Linky obrazem tlustší přetínají linky tenčí ve vodorovném i svislém směru“. Pořadí linek je tedy: tečkovaná, jednoduchá, tupojemná, dvojitá, polotučná a tučná. V případě, že se kříží linky stejného obrazu, přednost má linka vodorovná (základní linka, linka pod hlavičkou, součtová linka a patní linka u uzavřené tabulky).

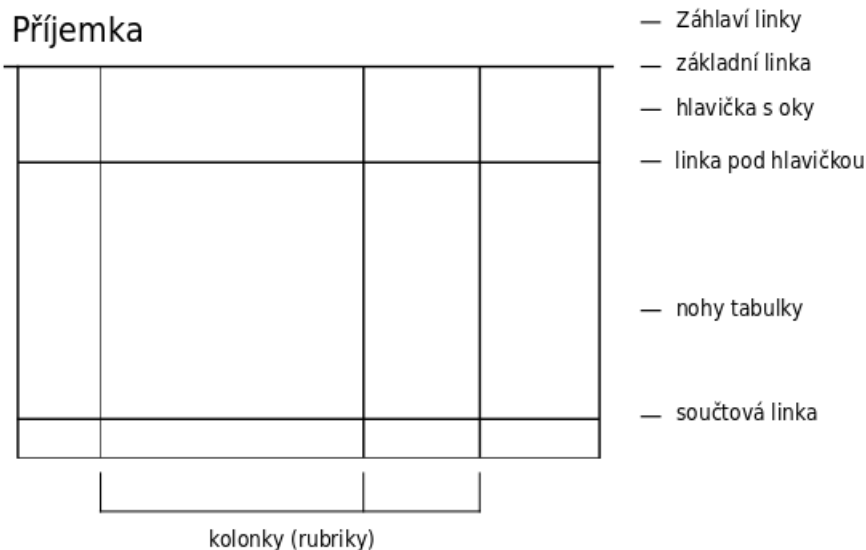
V nohách tabulky mají vždy přednost linky svislé před vodorovnými, viz. obr.2 (Pop, P, 1989)

### 3.1.4 Rozdělení tabulek

Tabulky se mohou dělit podle úpravy, účelu a technického zpracování.

#### Dělení tabulek podle úpravy

- Tabulky pořadově sázené – vyskytují se převážně v novinách, časopisech apod. Sází se ze stejného řezu i stupně, jako je základní text.



Obr. 1: Popis tabulky (Pop, P, 1989)

- Tabulky otevřené – nejsou uzavřeny svislými linkami po stranách, ale pouze linkou v záhlaví. Používají se převážně pro tisk na spadávání<sup>5</sup> viz 3.
- Tabulky uzavřené – používají se výhradně ve všech hospodářských tiskovinách, knihách, časopisech apod. Tyto tabulky jsou ohraničeny linkami stejné tloušťky na všech stranách tabulky viz 4.

(Pop, P, 1989)

**Dělení tabulek podle účelu použití**

- Hospodářskoadministrativní – jsou to tabulky formulářové, kartotékové, výkazové apod. Tyto se mohou dělit dále na textové (převažuje v nich text), číslicové (převážně jen čísla) a linkové (kombinace předchozích dvou),
- knižní a časopisecké – tyto tabulky jsou svou úpravou podřízeny celkové úpravě knihy, nebo časopisu,
- speciální – tyto tabulky jsou určeny například pro počítačové stroje. Důležitá je u nich přesnost.

(Pop, P, 1989)

**Dělení tabulek z hlediska technického zpracování**

- Tabulky sázené ručně,
- tabulky sázené na stroji,

<sup>5</sup>sazba od levého okraje papíru k pravému



Obr. 2: Křížení linek (Pop, P, 1989)

- tabulky kombinované,
- tabulky s příčnícem vsazeným,
- tabulky s příčnícem zvlášť,
- tabulky sázené na fotosázecích strojích

(Pop, P, 1989)

## 3.2 Analýza současného stavu tvorby tabulek

Analýza současného stavu se věnovala prostředím a balíkům pro systém  $\text{\LaTeX}$ , který je v současnosti asi nejvíce využívaná varianta.

### 3.2.1 Prostředí `tabbing`

Pomocí tohoto prostředí může uživatel nastavit tabulační zarážky a sázet text do sloupců a využívá se zejména k sazbě seznamů a přehledů. Toto prostředí lze také využít pro sazbu jednoduchých tabulek, ale užití tohoto prostředí pro tabulky není uživatelsky moc přívětivé. Kód, který musí uživatel vytvořit pro vysázení tabulky je nepřehledný a makra, která musí použít, nejsou příliš intuitivní a dávají prostor k chybám. (Talandová P, 2006)




Obr. 3: Tabulka otevřená (Pop, P, 1989)


Obr. 4: Tabulka uzavřená (Pop, P, 1989)

Existuje také novější varianta tohoto prostředí s názvem *Tabbing*. Tento balík vychází z původního prostředí, ale z důvodu snadnějšího zápisu znaků s diakritikou nahrazuje některé příkazy původního prostředí. Použití toho balíku je tedy příjemnější pro uživatele například z Francie, České Republiky, Slovenské Republiky apod. (Drucbert, 1998) Ukázka kódu:

```
\begin{Tabbing}
  Město Aaneco \TAB= Prostějovoneco \TAB= Vzdálenost[Km] \TAB= \kill
  Město A~     \TAB> Město B         \TAB> Vzdálenost[Km] \TAB> \\
  Brno        \TAB> Praha           \TAB> 204             \TAB> \\
  Brno        \TAB> Prostějov       \TAB> 61              \TAB>
\end{Tabbing}
```

Výsledná tabulka z tohoto kódu:

Město A	Město B	Vzdálenost[Km]
Brno	Praha	204

Brno                      Prostějov                      61

### 3.2.2 Prostředí tabular

*Tabular* je základní prostředí pro sazbu tabulek a slouží pro umístění tabulky do textu. Syntaxe definování tabulky je následující:

```
\begin{tabular}[y]{sloupce}
```

,kde *y* je volitelný parametr, pomocí kterého lze nastavit způsob připojení tabulky k okolnímu textu (horním okrajem – t, dolním okrajem – b). Defaultní nastavení je zařazení tabulky středem.

Dále následuje definice sloupců, který se definuje pomocí písmen: l,r,c,p. Každé písmeno určuje, jak bude sloupec zarovnán. Tedy k levému okraji, k pravému okraji, na střed a nebo do položky do bloku o šířce, která je zadaná jako následující parametr (hodí se pro delší obsah buňky). Počet sloupců se rovná počtu písmen (např. pro pět sloupců zarovnaných na střed oddělené vertikální čarou bude parametr příkazu c c c c c).

Toto prostředí umožňuje také oddělit jednotlivé řádky a sloupce vertikální a také horizontální čarou. Vertikální čára se definuje pomocí znaku | a zadává se při definici sloupců. Pro horizontální čáru se používá příkaz `\hline` nebo také `\cline{a-b}` pro čáru, která nebude přes celou šířku tabulky, ale jenom pro určité sloupce (a – b). V prostředí *tabular* má uživatel mnohem více možností si tabulku přizpůsobit. Například místo čáry jako mezi-sloupcového oddělovače může použít jiné znaky, je možné nastavit velikost mezi-sloupcové mezery apod. Pomocí dalších balíčků, které jsou většinou kompatibilní s příkazy tohoto prostředí lze spojovat sloupce a řádky, nastavit barvu pozadí apod.

Stejně jako v předchozím případě i tady existuje varianta a to je prostředí *tabular\**. Toto prostředí je prakticky shodné s původním, ale navíc má parametr pro nastavení celkové šířky tabulky. (LaTeX v kostce)

Zdrojový kód, který vygeneruje tabulku není v tomto případě příliš odlišný od předchozího. Prakticky pouze definuje jiné názvy příkazů pro nový řádek a pro ukončení buňky. Oproti předchozímu balíku ale dává uživateli více možností, například horizontální a vertikální čáry. Nevýhoda balíku je absence různých funkcí, pro které je zapotřebí importovat další balíky.

Následuje příklad předchozí tabulky vytvoření pomocí prostředí *tabular*, která je ohraničená čarami a ukázka kódu:

```
\begin{tabular}{|l|l|l|l|}
\hline
Město A~& Město B & & Vzdálenost [Km] \\ \hline
Brno & & Praha & 204 \\ \hline
Brno & & Prostějov & 61 \\ \hline
\end{tabular}
```

Město A	Město B	Vzdálenost[Km]
Brno	Praha	204
Brno	Prostějov	61

### 3.2.3 Prostředí `tabularx`

Syntaxe prostředí `tabularx` je definované stejně jako prostředí `tabular*` a přijímá stejné argumenty. Podstatný rozdíl oproti prostředí `tabular*` spočívá v sazbě tabulky na definovanou šířku. Prostředí `tabular*` rozšiřuje mezi-sloupcové mezery pro výslednou šířku tabulky, zatímco v `tabularx` je nutno definovat sloupec písmenem X. Tento sloupec se zarovná do bloku a bude mít takovou šířku, aby tabulka měla celkovou definovanou šířku. Takových sloupců může být i více. Celková šířka se mezi ně rozdělí. (Carlisle, D, 2014)

Způsob zápisu tabulky v tomto balíku je prakticky stejný jako v předchozím případě, takže není zapotřebí dále rozebírat.

```
\begin{tabularx}{250pt}{|X|X|l|}
\hline
Město A~& Město B      & & vzdálenost[Km] \\ \hline
Brno    & Praha              & & 204              \\ \hline
Brno    & Prostějov          & & 61               \\ \hline
\end{tabularx}
```

Město A	Město B	Vzdálenost[Km]
Brno	Praha	204
Brno	Prostějov	61

### 3.2.4 Balíky pro vícestránkové tabulky

Většina výše uvedených balíčků neumí sazbu tabulky na více stránek. Pomocí prostředí `tabbing` je to možné, ale jeho nastavení je příliš zdlouhavé a pro některé uživatele složité. Existují proto další balíky, pomocí kterých je toto možné a jsou to převážně balíky `supertabular` a `longtable`. Všechny tyto balíky používají stejné příkazy pro definici tabulky jako výše uvedené a není tedy zapotřebí je více specifikovat. Podrobnější popis včetně výhod a nevýhod jsou popsány níže.

**Supertabular** Tento balík zavádí nové prostředí `supertabular`. Principiálně funguje tak, že pro sazbu tabulek využívá prostředí `tabular` respektive `tabular*` a pokud se objeví příkaz pro ukončení řádku tabulky (`\`), zkontroluje se množství místa zbraného tabulkou. Pokud je tabulka již na konci stránky, ukončí se prostředí `tabular` a poté se na nové stránce toto prostředí vytvoří znovu. Uživatel může definovat hlavičku a patu tabulky společnou pro všechny stránky a zvlášť pro první stranu a jestli se vůbec zobrazí. (Braams, J, 2004)

Balík a prostředí má své chyby a zvláštnosti. Jelikož se pro každou stránku vytváří nové prostředí, může se například stát, že jednotlivé sloupce mají na každé stránce jinou šířku. Prostředí se také nehodí pro použití uvnitř plovoucího prostředí,

protože toto vyvolává snahu umístit celou tabulku na jednu stránku. Další z chyb, která se vyskytuje nastává, pokud poslední řádek tabulky způsobí přechod na novou stránku. Tato stránka následně obsahuje prázdnou tabulku (pouze hlavičku a zakončení). (Talandová P, 2006)

**Longtable** Tento balík definuje nové (stejnojmenné) prostředí, které je podobné prostředí (*super*)*tabular* a obsahuje většinu funkcí jako toto prostředí. Oproti prostředí *supertabular* ale používá pro zalomení stránek standardní algoritmus pro zalomení stránky. Stejně tak také sdílí některé vlastnosti prostředí *table*. (např. stejné počítadlo, příkaz `\caption`, `\listoftables`, který vytváří seznam tabulek v dokumentu, atd.). (Carlisle, D, 2014a)

Použití prostředí *longtable* je velice obdobné jako použití ostatních prostředí pro sazbu tabulek. Sloupce mohou být zarovnané vlevo, vpravo, na střed a do bloku. Tabulka se implicitně sází na střed stránky, uživatel ale může pomocí volitelného parametru toto chování změnit.

Jak již bylo napsáno výše, prostředí obsahuje většinu funkcí jako prostředí (*super*)*tabular*. Uživatel může například vysázet multisloupce, poznámky pod čarou, hlavičku která se objeví na všech stránkách, speciální hlavičku pro první stránku tabulky, patu tabulky atd. (Talandová P, 2006)

### 3.2.5 OPmac

OPmac je soubor doplňujících maker vytvořený RNDr. Petrem Olšákem v roce 2012. Tato makra procházejí stálým vývojem podle aktuální potřeby uživatelů, podle nalezených chyb apod. Makra jsou vytvořena pro systém Plain $\TeX$ a obsahují mnoho funkcionalit podobně jako systém  $\LaTeX$ . Patří mezi ně například změna velikosti písma, automatická tvorba obsahu a rejstříku apod. Mimo jiné obsahuje také velice kvalitní makra pro tvorbu tabulek. Balík je volně k dispozici a je možné využít jeho části a modifikovat. RNDr. Olšák také dává jeho uživatelům a programátorům, kteří by chtěli eventuálně balík využít, popřípadě modifikovat, možnost se s ním spojit pomocí emailu s doplňující otázkou, hlášením chyby apod. Pokud je to otázka nebo připomínka k nějaké chybějící vlastnosti nebo funkcionalitě, většinou se tato funkcionalita v brzké době objeví také v balíku. (Olšák, P, 2014)

## 3.3 Příklad sazby tabulek v různých systémech

V této části je malé porovnání sazby tabulek v různých systémech. Porovnání se vztahuje na tvorbu tabulky, uživatelské prostředí a vzhled tabulky.

### 3.3.1 LibreOffice Writer

První systém je LibreOffice Writer. Je to kancelářský balík podobný Microsoft Wordu. Má velice podobné vlastnosti, chování a ovládání. Velikým plusem tohoto balíku je to, že patří mezi freeware a je dostupný na všechny platformy (Linux, Windows a Macintosh), takže je zaručena přenositelnost dokumentů vytvořených tímto

systémem. Toto se například o Wordu říci nedá, protože na platformu Linux neexistuje a zpracování dokumentu ve formátu .doc může být na ostatních kancelářských balících problém.

Tvorba tabulky v tomto systému je jednoduchá a intuitivní pro každého, kdo někdy používal podobný software. Uživatel si může prakticky všechno naklikat pomocí myši. Určit si písmo a font, barvy řádků nebo buněk, zarovnání textu apod. Co se týče tvorby tabulek, je tento systém pro uživatele zaručeně nejpřívětivější. O přívětivosti a sazby textu v tomto systému ale tato práce už nepojednává.

Zdrojový kód této tabulky bohužel neporovnáme s ostatními. LibreOffice sice umožňuje exportovat tabulku do jazyka XML, ale tento kód je natolik dlouhý, že není možné ho sem vložit.

Vizuální podoba tabulky viz obr 5

	Sloupec 1	Sloupec 2	Sloupec 3
Řádek 1	0,125	Text ve dvou horizontálně sloučených buňkách	
Řádek 2	1,466	Buňka 2;2	Buňka 2;3
Řádek 3	13,542	Buňka 3;2	Buňka 3;3
Řádek 4	1235,546	Nějaký delší text, který je zapotřebí zalomit	Text ve dvou vertikálně sloučených buňkách
Řádek 5	1,456	Buňka 5;2	

Obr. 5: Tabulka vytvořená v LibreOffice

### 3.3.2 HTML

Vytváření tabulek v jazyku HTML už je o něco náročnější než v kancelářském softwaru a je velice podobné sazby tabulek v systému  $\text{\TeX}$ . Jak je již zmíněno výše, značkovací jazyk HTML obsahuje také značky a sazba se přibližuje spíše programování.

S určitými znalostmi jazyka HTML ale není sazba tabulek nějak extrémně náročná a definice pravidel pro tabulku také ne. Pro horizontální i vertikální sloučení buněk jsou v HTML relativně jednoduché příkazy, stejně tak pro barvy a nastavení fontu. Pro toto nastavení (vzhledu, zarovnání textu apod.) se v praxi definují spíše styly, ale pro tento příklad a porovnání bylo lepší tyto značky vložit přímo do definice tabulky samotné. Zdrojový kód tabulky vypadá následovně:

```
<table border="2" style="border-collapse: collapse" cellpadding="5" cellspacing="4">
<colgroup span="1">
<colgroup span="1" align="char" char=",">
  <thead>
    <tr style="height: 40px; background-color: #cccccc ">
      <th style="width: 120px"></th>
      <th style="width: 120px">Sloupec 1</th>
      <th style="width: 200px">Sloupec 2</th>
```

```

        <th style="width: 200px">Sloupec 3</th>
    </tr>
</thead>
<tbody>
    <tr style="background-color: #e6e6ff">
        <td>Řádek 1</td>
        <td style="text-align: right">0,1250</td>
        <td colspan='2'>Text ve dvou horizontálně sloučených buňkách</td>
    </tr>
    <tr>
        <td>Řádek 2</td>
        <td style="text-align: right">1,4665</td>
        <td>Buňka 2;2</td>
        <td style="text-align: right">Buňka 2;3</td>
    </tr>
    <tr style="background-color: #e6e6ff">
        <td>Řádek 3</td>
        <td style="text-align: right">13,5420</td>
        <td>Buňka 3;2</td>
        <td style="text-align: center">Buňka 3;3</td>
    </tr>
    <tr>
        <td>Řádek 4</td>
        <td style="text-align: right">1235,5400</td>
        <td>Nějaký delší text, který je zapotřebí zalomit</td>
        <td rowspan='2'>Text ve dvou vertikálně sloučených buňkách</td>
    </tr>
    <tr>
        <td>Řádek 5</td>
        <td style="text-align: right">1,4560</td>
        <td>Buňka 5;2</td>
    </tr>
</tbody>
</table>

```

Vizuální vzhled tabulky viz obr 6

	Sloupec 1	Sloupec 2	Sloupec 3
Řádek 1	0,1250	Text ve dvou horizontálně sloučených buňkách	
Řádek 2	1,4665	Buňka 2;2	Buňka 2;3
Řádek 3	13,5420	Buňka 3;2	Buňka 3;3
Řádek 4	1235,5400	Nějaký delší text, který je zapotřebí zalomit	Text ve dvou vertikálně sloučených buňkách
Řádek 5	1,4560	Buňka 5;2	

Obr. 6: Tabulka vytvořená v HTML

### 3.3.3 L<sup>A</sup>T<sub>E</sub>X

Nakonec ta stejná tabulka vytvořená v systému L<sup>A</sup>T<sub>E</sub>X.

Vytvoření jednoduché tabulky v systému L<sup>A</sup>T<sub>E</sub>X s použitím současných balíků není nikterak náročné a zvládne to každý, kdo má alespoň základy programování. Vytvořit ale složitější tabulku, ve které chce uživatel například jednu buňku ve sloupci zarovnat jinak než je zarovnaný sloupec (v praxi se příliš nevyskytuje, ale tato možnost může nastat) nebo horizontálně, či vertikálně sloučit buňky už je poněkud náročnější. Je potřeba použít různé balíky pro použití a definice barev.

Zdrojový kód:

```
%COLOURS DEFINITION
\definecolor{Gray}{gray}{0.85}
\definecolor{Even}{HTML}{E6E6FF}
\usepackage{dcolumn}
\usepackage{multirow}
\usepackage{multicol}
\usepackage{xcolor,colortbl}

\begin{tabular}{|l|r|p{5cm}|p{5cm}|}
\hline
\rowcolor{Gray}
& \textbf{Sloupec 1}
& \textbf{Sloupec 2}
& \textbf{Sloupec 3}
\\ \hline
\rowcolor{Even}
& Řádek 1
& 0,1250
& \multicolumn{2}{|l|}{Text ve dvou horizontálně sloučených buňkách}
\\ \hline
& Řádek 2
& 1,4665
& Buňka 2;2
& \multicolumn{1}{r|}{Buňka 2;3}
\\ \hline
\rowcolor{Even}
& Řádek 3
& 13,5420
& Buňka 3;2
& Buňka 3;3
\\ \hline
& Řádek 4
& 1235,5400
& Nějaký delší text, který je zapotřebí zalomit
& \multirow{3}{5cm}{Text ve dvou vertikálně sloučených buňkách}
\\ \cline{1-3}
& Řádek 5
& 1,4560
& Buňka 5;2
&

```

```

\\ \hline
\end{tabular}

```

Vizuální vzhled tabulky viz obr 7.

	Sloupec 1	Sloupec 2	Sloupec 3
Řádek 1	0,1250	Text ve dvou horizontálně sloučených buňkách	
Řádek 2	1,4665	Buňka 2;2	Buňka 2;3
Řádek 3	13,5420	Buňka 3;2	Buňka 3;3
Řádek 4	1235,5400	Nějaký delší text, který je zapotřebí zalomit	Text ve dvou vertikálně sloučených buňkách
Řádek 5	1,4560	Buňka 5;2	

Obr. 7: Tabulka vytvořená v systému  $\text{\LaTeX}$



## 4 Požadavky na nový balík

Analýza požadavků na nový balík pro sazbu tabulek, který by obsahoval všechny potřebné funkce a splňoval tak požadavky uživatelů na takovýto balík, vychází z diplomové práce Ing. Petry Talandové Ph.D. – Přístupy ve zpracování tabulek v systémech DTP, která je zaměřená právě na tuto problematiku.

Autorka v této práci provádí analýzu stávajících balíků a prostředí, typografická pravidla sazby tabulek, různý software pro sazbu textu a tabulek apod. Požadavky na nový balík vycházející z této práce a analýzy funkčnosti stávajících balíků jsou následující:

### 4.1 Umístění tabulky

- Horizontální – horizontální umístění tabulky na stránce by mělo být defaultně na střed stránky, ale uživatel musí mít možnost toto nastavení změnit buďto na umístění vlevo nebo vpravo.
- Vertikální – Tabulka by měla být umístěna v plovoucím prostředí tak, jak je to zvykem v systému  $\text{T}_{\text{E}}\text{X}$ . Uživatel by ale měl mít možnost definovat, zda se tabulka umístí přednostně na horní, či dolní okraj stránky, samostatnou stránku nebo do místa, kde bude do dokumentu vložena.  
Balík také musí být schopen rozdělit tabulku na více stran, pokud je příliš dlouhá, a v tomto případě vysázet hlavičku této tabulky na každé nové straně, pokud toto uživatel zvolí.

### 4.2 Ohraničení tabulky

- Ohraničení – Uživatel si může vybrat, jestli tabulka bude otevřená nebo uzavřená. Pokud bude tabulka otevřená, může uživatel definovat pouze horizontální linky. V případě uzavřené tabulky si uživatel může definovat jednak ohraničení tabulky a dále horizontální i vertikální linky.
- Styl linek – základní typ linek bude linka plná o tloušťce půl bodu a jeden bod a linka dvojitá. V případě že by uživatel chtěl použít jinou, nestandardní tloušťku linek, bude mít možnost toto definovat (v bodech). Linky jiného obrazu (např. čárkované) se využívají pouze okrajově, není tedy nutná podpora.
- Tisk linek – Uživatel musí být schopen nastavit délku linky. Defaultní nastavení bude tedy přes celou šířku tabulky, ale v případě potřeby pouze na některé sloupce. Další požadavek je na správné křížení linek, kde dvojitě linky budou mít přednost před jednoduchými.

### 4.3 Výška a šířka

- Výška řádků i celé tabulky by se měla řídit pouze jejich obsahem. Balík by ovšem měl dávat uživateli možnost nastavit také toto.
- Oproti tomu ale šířku sloupců i celé tabulky si může uživatel přizpůsobit nebo také ponechat na nastavení podle obsahu.
  - Nastavení šířky podle obsahu je standardní způsob tvorby tabulek, ovšem má svá úskalí. Pokud je obsah příliš dlouhý, tak tabulka svou šířkou může přesáhnout šířku papíru.
  - Další možností bude nastavení celkové šířky tabulky s tím, že tato šířka se rozdělí rovnoměrně mezi sloupce.
  - Uživatel bude mít také možnost nastavit každý sloupec zvlášť.
  - U posledních dvou nastavení se automaticky budou zalamovat řádky v buňkách, ve kterých bude text příliš dlouhý.
  - Šířka se bude zadávat v bodech.

### 4.4 Práce se skupinami polí

- Balík musí umožnit uživateli sloučit buňky. Ať už v horizontálním směru nebo vertikálním směru, neboť toto je velice používané.
- Balík by měl být schopen pracovat se skupinami polí – například tedy s celým sloupcem, řádkem, hlavičkou apod. Například nastavit barvu pro celý řádek, nastavit zarovnání obsahu pro sloupec apod.

### 4.5 Mezery v tabulce

- Mezera mezi obsahem buňky a jeho okrajem by měla být alespoň dva body a uživatel by měl mít možnost toto nastavení změnit.
- Mezi-sloupcové a meziřádkové mezery jsou běžně nastaveny stejné pro celou tabulku. Balík by opět měl dávat uživateli možnost toto změnit.

### 4.6 Zarovnání v tabulce

Zarovnání textu v tabulce, respektive v buňkách se má řídit charakterem údajů a množstvím světla. Zarovnání textu je tedy výhradně v režii uživatele.

- Vertikální zarovnání – možné varianty tohoto zarovnání budou nahoru, dolů nebo na střed. Defaultní hodnota bude na střed.
- Horizontální zarovnání – možné varianty budou vlevo, vpravo, na střed a v případě čísel na desetinnou čárku.

## 4.7 Barvy

Balík musí uživateli umožnit použít barvy a to pro obarvení textu, pozadí buněk (celých řádků a sloupců) popřípadě také obarvení linek.

## 4.8 Ostatní vlastnosti

- Název tabulky – Tabulky bude možné opatřit názvem daným uživatelem. Ten si také bude moci definovat, jestli tento název bude umístěn pod a nebo nad tabulkou.
- Písmo – typografická pravidla určují, že obsah tabulky má být vysázen písmem o jeden až dva stupně menším, než je základní text. Defaultně tedy bude text sázen písmem o velikosti dva body menším, než je základní text a stejného fontu. Uživatel ale bude mít možnost toto přenastavit.

## 5 Analýza tvorby tabulek v systému Plain $\TeX$

### 5.1 Makro `\leaders`

Slovo `leaders` se dá přeložit jako vedení. V Plain $\TeX$ u je toto makro použito ve významu vést oči v obsahu knihy od názvu kapitoly k číslu strany. Obvykle se k tomu používají tečky, toto makro ale umí stanovený prostor vyplňovat čímkoli.

`\leaders` vytváří pružný doplněk, který obvykle něčím vyplní. Použití makra je následující:

```
\leaders <box or rule> \textless glue specification>
```

kde `<box or rule>` je konstrukce boxu nebo linka” `\hrule` nebo `\vrule`. Tento parametr tedy určuje, co se bude (třeba opakovaně) ve výpňku sázet. Druhý parametr `<glue specification>` udává charakter výplňku, kde může být například `\vskip`, `\hskip`, `\hfil` a další. Použití může být například:

```
\leaders \hbox{.}\hfil - vysází tečky po celou délku výplňku \hfil
\leaders \hrule\hskip5cm - vznikne linka 0.4pt vysoká a~5 cm široká
```

Pokud je v prvním parametru použit primitiv `\hrule` nebo `\vrule` a makro je použito v horizontálním módu, tak se ignoruje šířka linky a použije se pouze šířka podle výsledného rozměru pružného výplňku (druhého parametru).

Pokud je v prvním parametru použit `box`, pak se mezera odpovídající šířce použitého pružného výplňku vyplní opakovaně daným boxem. V tomto případě se ale samozřejmě může stát, že šířka pružného výplňku není přesným násobkem šířky použitého boxu a nastane situace, kdy se do výplňku nevejde celý poslední box. (Olšák,P, 2001)

Tento problém může  $\TeX$  řešit třemi způsoby, v závislosti na tom, jakou variantu primitivu `\leaders` uživatel použije. Primitiv `\leaders` se chová tak, že se podívá na levý okraj vnějšího boxu, ve kterém je použit. Odtud začne pomyslně opakovat stanovený box až do chvíle, kde je pokryt celý text vlevo od místa, kde začíná `\leaders`. Poté začne skutečně sázet boxy a to do chvíle, než je vyplněna požadovaná mezera tak, že už se další box nevejde.

Hodí se proto pro vkládání teček například do obsahu. Kde tečky začne sázet prakticky okamžitě po ukončení textu před nimi a ukončí sázení prakticky těsně před dalším znakem, což je číslo strany. Tečky na řádcích budou navíc přesně pod sebou nehledě na délku názvu kapitoly, protože mezera mezi levým okrajem a začátkem teček se vypočítává se stejným základem a to je šířka boxu s tečkou. (Olšák,P, 2001)

Primitiv `\cleaders` vloží do požadované mezery maximální počet definovaných boxů těsně vedle sebe a prakticky je vycentruje. Mezeru, která se rovná nevyčerpanému zbytku, kam by se nevešel další box rozdělí na poloviny a vysází po levé a po pravé straně boxů.

Poslední varianta primitivu `\xleaders` vloží také maximální počet boxů, zbylou mezeru ovšem rozdělí mezi všechny vysázené boxy a také před první a za poslední box. (Olšák,P, 2001)

Tento primitiv se pro sazbu tabulek příliš nehodí, protože sází pouze stejně velké boxy se stejným textem.

## 5.2 Makra pro tvorbu tabulky s pevnou šířkou sloupců

### Primitiv `\hbox to dimen`

V případě, že je předem známá šířka každého sloupce lze vysázet tabulka pomocí primitivu `\hbox to <dimen>` a dalších několika primitiv (např. `\vrule` pro vertikální linky, `\hrule` pro horizontální linky, atd.). Definice tabulky poté může vypadat nějak takto:

```
\bgroup
\offinterlineskip
\vrule height4.5mm depth1.5mm \hbox to4cm{První pole \hfil}
\vrule height4.5mm depth1.5mm \hbox to4cm{Druhé pole \hfil}
\vrule height4.5mm depth1.5mm \hbox to2cm{Posledni}
\par \hrule width11cm
\vrule height4.5mm depth1.5mm \hbox to4cm{Druhy radek \hfil}
\vrule height4.5mm depth1.5mm \hbox to4cm{Druhy radek jedna \hfil}
\vrule height4.5mm depth1.5mm \hbox to2cm{Druhy}
\par \hrule width11cm
\egroup
\vskip 5mm
```

Tabulka by pak vypadala takto:

První pole	Druhé pole	Posledni
Druhy radek	Druhy radek jedna	Druhy

Samozřejmě by bylo možné definovat makra, která by toto poněkud usnadnila a kód by byl přehlednější a použitelnější, ale jak už bylo řečeno, tento primitiv se hodí pouze pro případy, kdy uživatel předem ví, jakou šířku budou mít sloupce. Využití tohoto primitivu je tedy poněkud omezené.

### Prostředí `\tabalign to dimen`

Dalším primitivem, kterým můžeme vytvořit jednoduchou tabulku se sloupečky s pevně definovanou šířkou je primitiv `\tabalign`. Tento primitiv je velice podobný LaTeXovému balíku *Tabbing*, to znamená, že vytváření tabulek je velice podobné psacímu stroji.

Uživatel může pro definování tabulky použít primitivy `\+` který se používá na začátku řádku. Po tomto primitivu uživatel definuje obsah řádku, který se skládá z textu, který má být vysázen, a tzv. tabulátory, které se značí symbolem `&` a na konec ukončí řádek primitivem `\cr`. (Olšák, P, 2001)

Definice tabulky pak může vypadat například takto:

```
\+ První řádek a~první buňka & druhá buňka & poslední v~prvním řádku \cr
\+ Druhý řádek, první & druhý řádek, druhá buňka & poslední \cr
```

Primitiv funguje tím způsobem, že na prvním řádku se vytvoří definované tabulátory podle aktuálních pozic znaků `&` a tyto pozice jsou potom stejné pro všechny další řádky, nehledě na to, jestli je text delší nebo ne. Může se tedy stát, že na některém řádku bude například text z druhého sloupce přepisovat konec textu z prvního sloupce.

Pro toto prostředí existují další primitiva, která dovolují předefinovat tabulátory a hýbat s nimi. Je to primit `\cleartabs`, který ignoruje všechny nastavené tabulátory, které následují po jeho použití. Tyto tabulátory může poté uživatel nadefinovat znova. Tento primitiv se používá přímo uvnitř obsahu řádku. (Olšák, P, 2001)

Další makro, kterým můžeme upravovat tabulátory je `\settabs`. Toto makro se používá mimo obsah řádku a vymaže polohy všech tabulátorů a umožní tak jejich nové nastavení.

Toto makro je možno použít více způsoby. Jeden ze způsobů použití je formát:

```
\settabs\+ nějaké dlouhé jméno & nějaké dlouhé příjmení & nazev predmetu \cr
\+ Jan & Hus & Dobrý \cr
\+ Bartoloměj & Skočdolesa & Nedostatečný \cr
```

Při tomto použití se nastaví tabulátory podle pozice na řádku (podle délky textu), ale tento text se netiskne. Je tedy možné takto nastavit tabulátory podle nějakého delšího textu. Jako například v ukázce výše. Pokud by nebyl použit první řádek s primitivem `\settabs`, ale rovnou řádek s prvním žákem, což je Jan Hus, tak hodnoty pro dalšího žáka by nebyly čitelné.

Další možné použití tohoto makra je `\settabs x\columns`. Tato definice vytvoří tabulku o „x“ sloupcích s tím, že poslední sloupec je přesně na pravém okraji zrcadla určeného podle `\hsize` a všechny ostatní sloupce jsou rovnoměrně daleko od sebe přesně na danou šířku, např:

```
\settabs 4\columns
\+\hss Jméno& \hss Příjmení& \hss Matematika& \hss Český jazyk&\cr
\+ Jan & Hus & Dobrý & Výborný \cr
```

Makro `\hss` v prvním sloupci přidává před text mezeru a tím centruje text v sloupci.

### 5.3 Tvorba tabulky pomocí `\halign`

Makro `\halign` je pro své vlastnosti a množství možností nejčastěji používané makro pro tvorbu tabulek. Používá ho i naprostá většina LaTeXových balíčků pro tvorbu tabulek, jako například `\tabular`.

Struktura zápisu tohoto makra vypadá následně:

```
\halign{<deklarace řádku> \cr
<položky tabulky prvního řádku> \cr
<položky tabulky druhého řádku> \cr
....
<položky tabulky posledního řádku> \cr
}
```

Vše, co je zapsáno mezi závorkami `{}` za primitivem `\halign` se označuje jako tělo tabulky. Toto tělo se dělí na dvě části a sice: deklarace řádku a datová část tabulky. Deklarace řádku je ukončena prvním výskytem primitivu `\cr`.

Deklarace řádku se používá pro deklarování sloupců, jejich počtu, zarovnání textu v daném sloupci, apod. Uživatel může v této deklaraci použít také přímo text, který se má objevit v každém sloupci. Například to může být sloupec s cenou (x Kč) nebo mírou (x cm, x mm), apod. Pro označení obsahu buňky se používá znak `#` (těchto znaků musí být v této části právě tolik, kolik je sloupců) a pro ukončení sloupce `&`. Pro ukončení řádku se používá primitiv `\cr`. (Olšák, P, 2001)

Deklarace například pro tabulku ohraničenou vertikálními linkami se třemi sloupci, kde první sloupec je zarovnaný vlevo, druhý nastřed a třetí vpravo může, vypadat takto:

```
\vrule\quad#\hfil\quad&\vrule\quad\hfil#\hfil\quad&%
\vrule\quad\hfil#\quad\cr
```

kde:

`\vrule` - je primitiv pro vertikální čáru

`\quad` - mezera mezi obsahem buňky a ohraničením buňky

`\hfil` - mezera, v tomto případě slouží pro nastavení horizontálního zarovnání.

`#\hfil` zarovnávní vlevo

`\hfil#\hfil` zarovnaní na střed

`\hfil#` zarovnaní vpravo

Datová část tabulky už obsahuje přímo řádky tabulky a jejich data. Zde se opět používá znak `&` pro oddělení jednotlivých buněk. Těchto znaků musí být v řádku právě o jeden méně, než je počet sloupců. V případě, že jich bude méně, zbývající položky budou prázdné a to včetně nadefinovaného vzoru z části deklarace řádku. Pokud by řádek obsahoval více těchto znaků, než je počet sloupců, `TeX` vygeneruje chybu.

Pro ukončení řádků se používá primitiv `\cr`. Tento primitiv je povinný i za posledním řádkem tabulky. V opačném případě `TeX` vygeneruje chybu. Pokud tedy programátor vytváří nějaké makro pro sazbu tabulky, tak nemůže spoléhat na uživatele, že správně ukončí tabulku, a v tomto případě se používá primitiv `\crcr`, který

se chová jako běžné `\cr`, ale v případě že se objeví hned za primitivem `\cr`, tak je ignorován.

Další primitiv, který se hojně využívá je `\noalign <vertikální seznam>`. Tento primitiv způsobí, že  $\TeX$  zatím neotvívá novou položku, ale akceptuje `<vertikální seznam >`, který uloží pod právě ukončený řádek. Primitiv se vyskytuje nejčastěji v kombinaci s primitivem `\hrule` a vytvoří tak horizontální čáru pod řádkem. Pokud se tento primitiv použije hned za prvním `\cr`, které ukončuje deklarační část, tak se tento `<vertikální seznam>` vloží nad první řádek tabulky.

Dalším užitečným primitivem, který se dá použít při tvorbě tabulky v prostředí `\halign` je primitiv `\omit`, který dává  $\TeX$ u povel, aby ignoroval všechny vlastnosti buňky definované v deklarační části a vyplní buňku pouze materiálem z datové části buňky. Tento primitiv musí být použit hned po znaku `&`.

Jeden z mocných a používaných primitivů je pak `\span`, kterým se dá nahradit separátor typu `&`. Tiskový materiál buněk, které sousedí s tímto primitivem mají následně společný box. V případě, že není použit také `\omit`, mají buňky také společnou deklaraci. Proto bývá obvyklé používat tento primitiv společně s `\omit`. Pro sloučení například tří sloupců v řádku by se použila sekvence – `\omit \span \omit \span \omit`. Tento zápis je ale dlouhý a nepřehledný, proto v `plain $\TeX$`  existuje makro `\multispan n`, který vygeneruje sekvenci těchto primitivů potřebnou pro sloučení `n` buněk.

Při čtení obsahu jedné položky  $\TeX$  otevře nový `\hbox` a zpracovává primitiva a texty, které jsou obsaženy v deklaraci řádku pro daný sloupec. Tyto zpracovává až po znak `#` (obsah buňky), poté zpracuje obsah dané buňky a nakonec zpracuje zbylou část maker v deklaraci řádku pro danou buňku, tedy část po pravé straně od znaku `#`. Pro každou položku v datové části se vytvoří speciální `\hbox`.

Primitiv `\halign` sestavuje tabulku ve dvou průchodech, kdy v jednom průchodu vytvoří výše uvedeným způsobem boxy jednotlivých položek, při čemž se na tyto boxy neklade žádný požadavek na stažení nebo roztažení. Poté se vypočítá maximální šířka boxů pro každý sloupec. Ve druhém průchodu se již sestavují řádky tabulky. (Olšák,P, 2001)



## 6 Popis implementovaného stylu

Vytvořený styl je založen na primitivu `\halign`. Po analýze všech primitiv a možností, na kterých by bylo možné postavit makra pro tvorbu tabulek, se tato možnost jevila jako nejlepší. Balík je z velké části inspirován balíkem `OPmac` a ukázkami kódu z knihy `TeXbook` naruby.

Styl se skládá z jednoho souboru pojmenovaném `tablePackage.tex`, který obsahuje definovaná makra a dokumentace. Definice maker, komentáře i základní dokumentace k balíku je napsaná v anglickém jazyce. Tento jazyk byl zvolen kvůli svému rozšíření, pokud by balík chtěl využívat, popřípadě modifikovat, někdo z jiné země. Toto jde velice obtížně, pokud je kód (názvy proměnných, komentáře) napsán v jiném jazyce než anglickém. Snahou bylo také napsat kód čitelně včetně odsazování, názvů proměnných, apod.

### 6.1 Základní příkazy

Tvorba tabulky pomocí tohoto balíčku by měla být co nejvíce intuitivní a pro uživatele jednoduchá. Některá nastavení jsou tedy předdefinována přímo v balíku a uživatel by v podstatě neměl mít potřebu je měnit. Existují proto některé základní příkazy, se kterými si uživatel vystačí pro jednoduché tabulky, a v případě, že tabulku musí nějakým způsobem upravit, má k dispozici další příkazy, které mu to umožňují.

#### 6.1.1 `\table`

První základní příkaz, který uživatel musí v každém případě použít je příkaz:

```
\table[nofloat]{zarovnání}{popisek tabulky}
```

Tento příkaz je zároveň první, který musí uživatel použít pro vygenerování tabulky. Příkaz ve své podstatě neprovádí nic kromě resetování základních proměnných, které se používají v dalších makrech, jako je například počet sloupců tabulky, pozice aktuálního sloupce, výška řádku, šířka linek apod. Vloží do textu vertikální mezeru, aby tabulka nebyla příliš blízko textu, který je nad ní a podle zadaného parametru vytvoří vertikální box zarovnaný vlevo, na střed nebo vpravo, do kterého se dále bude tabulka sázet. Dále vysází definovaný popisek tabulky a vloží primitiv `\bgroup`, aby se všechny změny parametřů, definic apod projevíly pouze lokálně v této tabulce. Příkaz má tři parametry. První z nich `[nofloat]` je nepovinný. Základní nastavení balíku je takové, že pro tabulku využívá plovoucí prostředí. To znamená, že v případě, že uživatel chce vložit tabulku do určitého místa v dokumentu, a `TeX` zjistí, že tabulka se do tohoto místa nevejde kvůli konci stránky, vloží do tohoto místa text, který následuje za tabulkou a tabulku samotnou vysází při první příležitosti na další stránce místo toho, aby za cenu toho že dodrží přesný tok textu nechal nějaké místo na stránce prázdné.

Toto je základní chování prakticky všech balíčků pro vkládání objektů jako jsou tabulky, obrázky apod. Proto i v tomto balíku je toto chování nastaveno a pouze v případě, že uživatel nechce použít plovoucí prostředí, musí použít tento parametr. Ten musí být zapsán přesně v této podobě *[nofloat]*.

Druhý a třetí parametr *{zarovnání}{popisek tabulky}* jsou povinné. Druhý parametr slouží pro horizontální zarovnání tabulky a může obsahovat možnosti: *left, right, center*. Třetí a poslední slouží pro nastavení popisku tabulky, který bude vysázen nad tabulkou.

```

\def\table{
  \futurelet \nextchar \isFloat
}

\def\isFloat{ % Checking if the optional parameter - no float is filled
  \ifx [\nextchar \let\next=\tableNOfloat
  \else
    \let\next=\tablefloat
  \fi \next
}

\def\tablefloat#1#2{ % Begin of the table env - settings of some variables
  \vertSpace
  \midinsert
  \global\float=1
  \def\left{left}
  \def\right{right}
  \def\center{center}
  \def\param{#1}
  \caption{#2}
  \vskip 4pt
  \ifx\param\left
    \vbox\bgroup
  \fi
  \ifx\param\right
    \hfill\vbox\bgroup
  \fi
  \ifx\param\center
    \hfil\vbox\bgroup
  \fi
  \global\colnum=0
  \global\actcol=0
  \global\strip=0

```

```

\global\tabline=0
\global\strutheight=11pt
\global\horLineHeight=0.2pt
\global\verLineWidth=0.2pt
\global\verbortype=0
\global\horbortype=0
}

```

Použití příkazu může vypadat například takto:

```
\table[nofloat]{left}{Tabulka příjmů za první čtvrtletí 2015}
```

### 6.1.2 \tablebody

Další základní a povinný příkaz je:

```
\tablebody[šířka]{zarovnaní sloupců}{tělo tabulky}
```

Příkaz má také tři parametry z nichž jeden je nepovinný. Prvním parametrem má uživatel možnost definovat celkovou šířku tabulky. Tuto šířku může definovat jakýmkoli způsobem, který přijímá  $\TeX$  (např: 8cm, 30em, 300pt). Celková šířka tabulky poté bude mít tento rozměr.

Druhý parametr slouží pro nastavení počtu sloupců tabulky a jejich zarovnaní. Toto se provádí stejně jako je zvykem v dalších balíčcích pomocí písmen – l,c,r. Význam těchto písmen je left (vlevo), center (na střed) a right (vpravo). Počet sloupců se potom rovná počtu těchto použitých písmen.

Poslední parametr bude obsahovat tělo tabulky, které uživatel definuje pomocí dalších maker. Tato část je vysvětlena níže.

Makro v první řadě kontroluje, jestli byl použit první nepovinný parametr. Pokud ano, zavolá se makro `\tableto`, pokud nebyl vyplněn, zavolá se makro `\tablebodyNormal`.

```

\def\tablebody{
  \futurelet \nextchar \isDefinedWidth
}

\def\isDefinedWidth{ % Checking if the optional parameter is filled
  \ifx [\nextchar \let\next=\tableto
  \else
    \let\next=\tablebodyNormal
  \fi \next
}

```

Makro `\tablebodyNormal` pouze volá další makro `\tableA`.

Toto makro obsahuje příkaz `\offinterlineskip`, který je potřebný proto, aby byly boxy ve vertikálním módu sázeny pod sebe s nulovou meziřádkovou mezerou. Dále toto

makro vymaže aktuální obsah proměnné `\tabdata` a zavolá makro `\scantabdata` s prvním parametrem. Toto makro znak po znaku prohledává první parametr, který je vlastně deklarací sloupců, kontroluje jeho obsah (jestli neobsahuje nepovolený znak) a podle použitých znaků volá další makra, která do proměnné `\tabdata` ukládají postupně deklaraci řádku. Poté zavolá makro `\halign`, které už slouží pro vytvoření samotné tabulky, ve kterém použije již zmiňovanou proměnnou `\tabdata`, obsah druhého parametru, což je vlastně tělo tabulky a nakonec přidá makro `\trend`, které obsahuje povinný primitiv `\crr`, a v případě, že uživatel definoval tabulku s horizontálními linkami, také tuto linku.

```

\def\tablebodyNormal{
  \tableA
}

\def\tableA#1#2{\offinterlineskip \tabdata={}\scantabdata#1\relax
  \halign\expandafter{\the\tabdata\crr#2\trend}
}

%%%% FUNCTIONS FOR TABLE DECLARING %%%
\def\scantabdata#1{\let\next=\scantabdata
  \ifx\relax#1 \ifnum \verbortype=1
    \addtabdata{\vrule width\verLineWidth}
  \fi \let\next=\relax
  \else\expandafter\ifx\csname tabdeclare#1\endcsname \relax
    \warning{tab-declare letter #1 unknown, ignored}%
  \else\expandafter \addtabitem\expandafter
    {\csname tabdeclare#1\endcsname}%
  \fi\fi \next
}

\def\addtabitem#1{ % Adding table declaring into toks - tabdata
  \ifnum\colnum>0 \addtabdata{&}\addto\ddlinedata{&\dditem}\fi
  \global\advance\colnum by1 \let\tmpa=\relax
  \expandafter\addtabdata\expandafter{#1\strut}}
\def\addtabdata#1{\tabdata\expandafter{\the\tabdata#1}
}

\gdef\trend{ % Default new line on the end of table
  \ifnum\horbortype=1 \the\newLineWithBorder \else \the\newLine \fi
}

```

Pokud tedy uživatel definuje také nepovinný parametr se šířkou tabulky, aktivuje se makro `\tableto`. Toto makro v podstatě nanečisto vytvoří tabulku s definovanými daty. Poté zkontroluje šířku vytvořené tabulky a šířku definovanou uživatelem. Roz-

díl těchto šířek poté podělí dvojnásobkem počtu sloupců a výsledné hodnoty použije pro mezery v buňkách (`\tabitemr`, `\tabiteml`). Takto se tabulka roztáhne na požadovanou šířku.

```
\def\tableto[#1]#2#3{%
{\def\tabiteml{}\def\tabitemr{}\setbox0=\tablewidth{#2}{#3}
  \global\colnum=0\global\actcol=0%
  \tmpnum=0 \countcols#2\relax \tabw=#1\advance\tabw by-\wd0
  \divide\tabw by\tmpnum \def\tabiteml{\kern\tabw}
  \def\tabitemr{\kern\tabw}\tablewidth{#2}{#3}}
}

\def\countcols#1{\ifx#1\relax\else
  \ifx#1|\else\advance\tmpnum by2 \fi \expandafter\countcols \fi
}

\def\tablewidth{
  \vbox\bgroup \tablew
}

\def\tableW#1#2{
  \offinterlineskip \def\tmpa{}\tabdata={}\scantabdata#1\relax
  \halign\expandafter{\the\tabdata\crr#2\trend}\egroup
}
```

Použití makra je velice jednoduché. Například pro tabulku se čtyřmi sloupci zarovnanými vlevo, vlevo, na střed a vpravo bude definice vypadat takto:

```
\tablebody{llcr}{tělo tabulky}
```

Pro tabulku širokou 12 cm se třemi sloupci, kde všechny jsou zarovnané na střed by použití bylo:

```
\tablebody[12cm]{ccc}{tělo tabulky}
```

### 6.1.3 `\endtable`

Další povinný příkaz, tentokrát pro ukončení tabulky. Tento příkaz primitivem `\egroup` ukončí vytvořený úsek, ve kterém se lokálně projeví změny některých parametrů a definic. V případě, že uživatel nepoužil parametr pro potlačení plovoucího prostředí, tak se tady ukončí a nastaví se hodnoty některých proměnných. Nakonec se vysází vertikální mezera mezi tabulkou a následujícím textem.

```
\def\endtable{ % End of the table env
  \egroup
  \vertSpace}
```

```

\ifnum\float=1
  \endinsert
\fi
\global\strip=0
\global\strutheight=11pt
\global\horLineHeight=0.2pt
\global\verLineWidth=0.2pt
}

```

## 6.2 Rozšiřující makra pro další nastavení tabulky

Jak již bylo řečeno, uživatel může vytvořit tabulku pouze za pomoci několika málo maker. Většina uživatelů je ale nejspíš více náročná a proto existují další makra, která uživatel může použít pro přizpůsobení tabulky. Tato makra se zapisují mezi makra `|table` a `textit|tablebody`. Některá z nich uživatel využije při každé definici tabulky, některá nemusí použít vůbec.

### 6.2.1 `\bordertype`

Toto makro slouží pro definování typu tabulky. Syntaxe zápisu vypadá následně:

```
\bordertype[rozměr]{typ čar}
```

První parametr je opět nepovinný a slouží k nastavení šířky čar (horizontálních i vertikálních). Uživatel opět může použít jakýkoliv druh zápisu, který  $\TeX$  přijímá (např. 0.5pt, 0.1cm, ...). Základní nastavení šířky je 0,2 pt.

Druhý parametr je povinný a přijímá čtyři možnosti – *none*, *cross*, *vertical*, *horizontal*. Při možnosti *none* bude tabulka neohraňovaná a nevysázejí se ani horizontální, ani vertikální linky. Pokud uživatel použije možnost *cross*, je to pravý opak a budou automaticky použity jak vertikální, tak horizontální linky. V případě použití možnosti *horizontal* nebo *vertical*, vysází se automaticky pouze jeden typ čáry v závislosti na použité možnosti.

Makro opět nejprve zkontroluje, jestli byl použit nepovinný parametr s definicí šířky linek. Pokud ano, zavolá se makro `|borderTypeWithWidth`, pokud ne tak se použije `|borderTypeWithoutWidth`. Tato dvě makra mají společnou část `|borType`, kde se nastavují hodnoty proměnných `|verbortype` a `|horbortype`, které jsou potřeba pro další použití. Tyto proměnné mají nabývají hodnot 0 nebo 1, jako false nebo true.

```

\def\bordertype{          % Function for setting up of border lines
  \futurelet \nextchar \isBorderWidth
}

```

```

\def\isBorderWidth{
  \ifx [\nextchar \let\next=\borderTypeWithWidth %YES

```

```

\else
  \let\next=\borderTypeWithoutWidth          %NO
\fi \next}

\def\borderTypeWithWidth[#1]#2{
  \global\horLineHeight=#1
  \global\verLineWidth=#1
  \borType{#2}
}

\def\borderTypeWithoutWidth#1{
  \borType{#1}
}

\def\borType#1{
  \def\none{none}
  \def\cross{cross}
  \def\vertical{vertical}
  \def\horizontal{horizontal}
  \def\param{#1}
  \ifx\param\none
    \global\verbortype=0
    \global\horbortype=0
  \fi
  \ifx\param\cross
    \global\verbortype=1
    \global\horbortype=1
  \fi
  ...
  ...
}

```

Použití tohoto makra může vypadat například takto:

```

\bordertype[3pt]{cross} - pro ohraničenou tabulku s~šířkou linek 3 pt
\bordertype{none} - pro neohraničenou tabulku

```

Toto makro není povinné a v případě, že je uživatel nepoužije, použije se základní nastavení proměnných `\verbortype` a `\horbortype`, které je 0 jako false.

### 6.2.2 `\striped`

Toto makro slouží pro nastavení tabulky, která bude mít každý druhý řádek obarvený určitým stupněm šedé barvy. Syntaxe makra vypadá takto:

```
\striped[0-1]
```

Parametr makra je nepovinný a slouží pro nastavení stupně šedé barvy. Použití je možné číslo v intervalu od 0 – 1, kde 0 znamená bílá barva a 1 znamená černá barva. Makro je definováno tak, že zkontroluje existenci parametru. V případě, že je definován stupeň šedé uživatelem, zavolá se makro `\definedStriped`, ve kterém se nastaví proměnná `\strip` na hodnotu 1, tedy jako true (tato proměnná se poté kontroluje při vytváření nových řádků) a proměnná `\definedColor` se nastaví na stupeň šedé definovanou uživatelem. V případě, že uživatel chce využít základní nastavení stupně šedé a nepoužije nepovinný parametr, zavolá se makro `\defaultStriped`, kde se provedou stejné kroky jako v předchozím makru, ale hodnota proměnné `\definedColor` se nastaví na již definovanou barvu `\LightGrey` (stupeň šedé je roven 0,2).

```
\def\striped{
  \futurelet \nextchar \isDefinedColor
}

\def\isDefinedColor{ % Checking if the optional parameter is filled
  \ifx [\nextchar \let\next=\definedStriped
  \else
  \let\next=\defaultStriped
  \fi \next
}

\def\definedStriped[#1]{
  \global\strip=1
  \gdef\definedColor{\setcmykcolor{0 0 0 #1}}
}

\def\defaultStriped{
  \global\strip=1
  \gdef\definedColor{\LightGrey}
}
```

### 6.2.3 Další příkazy pro nastavení

Sem patří pár dalších maker pro různá nastavení. Tyto makra nedělají nic jiného, než že nastaví hodnoty proměnných, které jsou použity při vytváření tabulky. Patří sem:

- `\cellspaces{x}` – pomocí tohoto makra lze nastavit velikost mezery před a za textem v buňkách.  
Příklad: `\cellspaces{5em}`
- `\rowsheight{x}` – toto makro slouží pro nastavení velikosti řádků v tabulce. Defaultní hodnota je nastavena na 11 pt.  
Příklad: `\rowsheight{16pt}`



- `\vertlinewidth{x}` – tímto příkazem může uživatel nastavit tloušťku vertikálních čar v tabulce, pokud by chtěl nastavit jinou šířku pro vertikální a horizontální čáry a nepoužil tak nepovinný parameter v příkazu `\bordertype`.  
Příklad: `\vertlinewidth{3pt}`
- `\-horlinewidth{x}` – tento příkaz slouží pro nastavení tloušťky horizontálních linek.  
Příklad: `\-horlinewidth{1pt}`

## 6.3 Obsah těla tabulky

Nyní přicházejí na řadu makra definovaná přímo pro vytvoření tabulky, které se používají ve druhém povinném parametru makra `\tablebody`. Jak již bylo řečeno, cílem projektu bylo definovat taková makra, která budou jednoduchá na použití a budou se podobat tagům používaných k tvorbě tabulky v HTML.

### 6.3.1 `\title`

Toto makro může uživatel použít pro nadpis tabulky. Syntaxe je velice jednoduchá:

```
\title{Nadpis tabulky}
```

Makro automaticky vloží nový řádek do tabulky, buď s linkou nebo bez linky v závislosti na tom, jaký je použit styl tabulky, sloučí všechny buňky v řádku a vypíše daný nadpis. Tento nadpis je vycentrovaný doprostřed řádku a je vysázený polotučným písmem.

```
\def\title#1{ % Title of the table
  \tr \multispan\colnum \strut \hskip-0.33em \ifnum \verbortype=1
  \vrule width\verLineWidth \fi
  \tabiteml\hfil\fontbf{#1}\hfil\tabitemr \ifnum \verbortype=1
  \vrule width\verLineWidth \fi
}
```

### 6.3.2 `\tr`

Název makra vychází z HTML (table row), kde se tento tag používá pro nový řádek tabulky. V tomto balíku je funkce stejná. Makro tedy vytvoří nový řádek tabulky. Makro nepřijímá žádný parametr, zápis je tedy velice snadný. V těle makra proběhne kontrola proměnné `\strip` a `\horbortype`. V závislosti na hodnotě těchto proměnných (1 nebo 0) se zavolají příslušné proměnné, které obsahují přímo definice nového řádku.

V případě, že uživatel použil makro `\striped` a hodnota proměnné `\strip` je 1, zavolá se navíc makro `\colorline`. Toto makro zkontroluje, jestli je právě tisknutý řádek lichý. Pokud ano, zavolá se makro `\colortabline`, které má za úkol podbarvit řádek. Toto probíhá tak, že na již vytvořeném řádku se vysází nový obarvený obdelník

pomocí primitivu `\hrule` s výškou a hloubkou stejnou jako je výška a hloubka řádku a šířkou jako je šířka tabulky. Nakonec ještě musí proběhnout kontrola, jestli se má vysázet vertikální linka v řádku. Pokud ano, vrátí se barva zpět na černou a vysází se tato linka znovu, neboť původní se vysázela v šedé barvě a není zřetelná.

```

\def\tr{%
  \ifnum\strip=0
  \ifnum\horbortype=1 \the\newLineWithBorder
  \else \the\newLine\fi
  \else
  \ifnum\horbortype=1 \the\newLineWithBorder\colorline
  \else
  \the\newLine\colorline \fi
  \fi
}

\def\colorline{ % Checks if the line is odd -> grey background
  \ifodd\tabline \colortabline \fi
  \global\advance\tabline by1
}

\newtoks\newLine
\newLine={\crr}

\newtoks\newLineWithBorder
\newLineWithBorder={\crr \noalign{\hrule height\the\horLineHeight}}

\def\colortabline{ % Line in the table with color background
  \noalign{\expandafter\localcolor\definedColor
  \hrule height\ht\strutbox depth\dp\strutbox \kern-\ht\strutbox
  \kern-\dp\strutbox \ifnum\verbortype=1
  \expandafter\localcolor\Black \vrule width\verLineWidth \fi}

```

### 6.3.3 `\trwithline`

Funkce tohoto makra je zřejmá z názvu. Je to nový řádek s horizontální linkou. V tomto makru neprobíhá kontrola definovaného typu tabulky, ale horizontální linka se vysází bez ohledu na tuto proměnnou. Použití tohoto makra se hodí v případě, že uživatel chce mít horizontální linky pouze na některých místech (nejčastěji základní linka, linka pod hlavičkou a součtová linka nebo linka na konci tabulky). Použije tedy v `\bordertype` parameter *none* a na výše uvedených místech použije makro `\trwithline`.

```
\gdef\trwithline{                % New line with border
  \the\newLineWithBorder
}
```

### 6.3.4 `\trwithcolor`

Další alternativa k makru `\tr`. Tentokrát makro pro sazbu podbarveného řádku.

```
\trwithcolor{0-1}
```

Makro má jeden povinný parametr a sice stupeň šedé pro podbarvený řádek. Makro opět kontroluje jaký styl tabulky uživatel definoval a podle toho vysází podbarvený řádek s určeným stupněm šedé barvy.

```
\def\trwithcolor#1{
  \ifnum\horbortype=1 \the\newLineWithBorder \colorthisline{#1}
  \else \the\newLine\colorthisline{#1} \fi
}
```

```
\def\colorthisline#1{ % Line in the table with color background
  \noalign{\expandafter\localcolor\setcmykcolor{0 0 0 #1}
  \hrule height\ht\strutbox depth\dp\strutbox \kern-\ht\strutbox
  \kern-\dp\strutbox \ifnum\verbortype=1
  \expandafter\localcolor\Black \vrule width\verLineWidth \fi}
}
```

### 6.3.5 `\td`

Toto makro slouží pro vkládání obsahu jednotlivých buněk. Název makra vychází opět z HTML tagu. Použití je opět velice jednoduché. Makro má jeden povinný parametr, což je obsah buňky.

```
\td{obsah buňky}
```

Makro funguje tak, že kontroluje pozici aktuálního sloupce a pokud sloupec není poslední, vloží za text separátor sloupců (&). Pokud je poslední, tento separátor se nevloží a proměnná se vrátí zpět na nulu.

```
\def\td#1{                % Table cell
  \global\advance\actcol by1
  \ifnum\actcol<\colnum #1 \separ \else #1 \global\actcol=0\fi
}
```

### 6.3.6 `\th`

Varianta pro předchozí makro. Kdo alespoň několikrát tvořil tabulky v HTML, tak určitě pochopí i funkci makra. Název je zkratka pro – table head. Makro funguje

na stejném principu jako předchozí `\td`, ovšem text vycentruje a vysází polotučným písmem. Použití je stejné jako u předchozího makra. Má tedy jeden parametr, kam uživatel zapíše obsah buňky.

Použití: `\th{Obsah buňky}`

Implementace:

```
\def\th#1{ % Table head - center and bold
  \global\advance\actcol by1
  \ifnum\actcol<\colnum \hfil\fontbf{#1}\hfil \separ
  \else
  \hfil\fontbf{#1}\hfil \global\actcol=0\fi
}
```

### 6.3.7 `\tdalign`

Další varianta pro vytváření buňky. Tuto variantu využije uživatel v případě, že v dané buňce požaduje jiné zarovnání, než je definováno ve sloupci. V  $\LaTeX$ ovém balíku `\tabular` je toto vyřešeno pomocí makra pro sloučení sloupců `\multicolumn`, kde ale uživatel do prvního parametru, který má obsahovat požadovaný počet sloučených buněk zadá číslo jedna a poté typ zarovnání. Toto ale není příliš elegantní řešení a proto bylo v tomto balíku vytvořeno makro se syntaxí:

```
\tdalign{zarovnání}{obsah buňky}
```

Parametry makra jsou dva a oba jsou povinné. První parametr slouží pro definici zarovnání a přijímá klasické možnosti – l,c,r. Do druhého parametru se vloží text, který má buňka obsahovat.

Makro používá stejně jako v balíku `\tabular` primitiv `\multispan` s číslem 1, což znamená prakticky sloučení jedné buňky. Dále probíhá kontrola typu tabulky a po případě vysázení vertikální linky. Následuje kontrola druhého parametru. V případě, že uživatel zadal jiné písmeno než l,c nebo r se vypíše chybová hláška. Jinak proběhne požadované zarovnání, kontrola aktuálního sloupce, zda není poslední, pro vysázení oddělovače sloupců. Pokud je buňka v posledním sloupci, proběhne opět kontrola nastavení typu tabulky a případné vysázení vertikální linky.

```
\def\tdalign#1#2{
  \multispan1\strut\hskip-0.66em\ifnum\verbortype=1
  \vrule width\verLineWidth\fi
  \if#1l \alignl{#2} \else \if#1c \alignc{#2}
  \else \if#1r \alignr{#2}
  \else\expandafter\ifx\csname align#1\endcsname \relax
  \warning{Colspan align letter #1 unknown, ignored} \fi\fi\fi\fi
  \global\advance\actcol by1 \ifnum\actcol<\colnum \separ \fi
  \ifnum\actcol=\colnum \ifnum \verbortype=1
  \vrule width\verLineWidth
```

```

\fi \global\actcol=0\fi
}
\def\alignc#1{          % Definition of center align in the table
  \tabiteml\hfil#1\hfil\tabitemr
}

\def\alignl#1{         % Definition of left align in the table
  \tabiteml#1\hfil\tabitemr
}

\def\alignr#1{         % Definition of right align in the table
  \tabiteml\hfil#1
}

```

### 6.3.8 `\colspan`

Toto makro slouží pro sloučení buněk v řádku. Pracuje prakticky stejně jako předchozí. Jediná odlišnost je v počtu parametrů, kde jako první parametr se udává počet buněk, které chceme sloučit, druhý parametr řídí zarovnání této výsledné buňky (opět l,c,r) a do třetího parametru se zapisuje požadovaný obsah buňky. Jediná odlišnost v kódu je, že při volání `\multispan` se tomuto primitivu předá první parametr, který určí počet sloučených buněk. Proto není potřeba zde tento kód ukazovat. Pouze způsob použití. Například pro sloučení tří buněk, zarovnání na střed:

```
\colspan{3}{c}{Nějaký text}
```

Po použití tohoto makra (pro sloučení tří buněk) se bude v daném řádku vyskytovat o dva příkazy `|td` méně.

### 6.3.9 `\rowspan`

Další makro pro slučování buněk, tentokrát pro sloučení buněk v jednom sloupci. Použití makra je podobné jako v předchozím případě. Makro má tři povinné parametry. První parametr očekává počet buněk ve sloupci, které se mají sloučit, druhý parametr nastavuje vertikální zarovnání. V tomto případě makro přijímá opět tři písmena, tentokrát ale písmena – u,c,d. Význam písmen znamená – up, center, down. Do třetího parametru se udává požadovaný obsah buněk:

```
\rowspan{počet buněk}{vertikální zarovnání}{obsah buňky}
```

Makro pracuje tak, že podle uvedeného počtu buněk, které se mají sloučit, podle definovaného zarovnání a podle výšky řádku v tabulce, se vypočítá hodnota posunutí boxu, která se uloží do proměnné `\displacement`. Proměnná `\actcol`, ve které je uložen aktuální sloupec, se zvýší o jedno a do tabulky se vloží `\vbox`, který obsahuje `\hbox` s požadovaným textem. Tento box je posunutý směrem nahoru o vypočítá-

nou hodnotu `\displacement`. Nakonec se zkontroluje `\actcol` a popřípadě se vloží oddělovač sloupců.

```
\def\rowspan#1#2#3{
  \global\displacement=0pt
  \if#1d \displacement=\the\strutheight \fi
  \if#1c \displacement=\the\strutheight \multiply\displacement by #2
  \advance\displacement by-5pt \fi
  \if#1u~\displacement=\the\strutheight \multiply\displacement by #2
  \pom=\the\strutheight \divide\pom by 2 \advance\displacement by \pom \fi
  \global\advance\actcol by1
  \leavevmode\vtop{\vskip-\the\displacement
  \vbox to \the\strutheight{\vfill \hbox{#3}\vfill}\vss}
  \ifnum\actcol<\colnum \separ \else \global\actcol=0\fi
}
```

Z důvodu, že makro pracuje na principu posunutí boxu směrem nahoru, je nezbytné, aby buňky ve stejném sloupci, které mají být sloučené byly prázdné. Makro bude fungovat i v případě, že prázdné nebudou, text se ale bude překrývat, takže každého uživatele posléze napadne buňky vyprázdnit. Uživatel ale musí definovat prázdné buňky příkazem `\td`.

Použití může vypadat například takto:

```
...
...
\tr
  \td{}
  \td{notebook}
  \td{14 000 Kč}
\tr
  \td{}
  \td{tablet}
  \td{4 500 Kč}
\tr
  \rowspan{c}{3}{Leden}
  \td{mobilní telefon}
  \td{3 000 Kč}
\tr
...
...
```

### 6.3.10 `\trcustom`

Toto makro s největší pravděpodobností využije uživatel, který použije předchozí makro pro sloučení buněk ve sloupci. Slouží k vysázení horizontálních linek pouze

pro definované sloupce. Makro se používá následovně:

```
\trcustom{definice čar}
```

Do těla parametru se zapisují příkazy `\cline` nebo `\noline` v závislosti na tom, zda si uživatel přeje vysázet horizontální čáru pro daný sloupec. V případě, že by uživatel chtěl vysázet horizontální čáry například pouze pro první a třetí sloupec, použije makro následovně:

```
\trcustom{\cline\noline\cline}
```

Makro pracuje na jednoduchém principu, kdy se do tabulky vloží nový řádek, ovšem v každé buňce je použit primitiv `\omit` pro ignorování definovaného obsahu a v případě, že v této buňce má být horizontální linka, vloží se také `\hrulefill`, které vyplní buňku touto linkou. Díky použití `\omit` má potom tento řádek pouze velikost této linky což je 0.4 pt. Makro `\trcustom` vyprázdní obsah proměnné `\customLines`, do které se ukládají požadavky uživatele. Poté následuje zpracování parametru makra, při kterém se postupně naplní tato proměnná primitivy pro sazbu linek v požadovaných sloupcích a poté se zavolá makro `\custLines`, které vysází nový řádek, proměnnou s čárami a opět nový řádek. Nakonec vynuluje proměnnou obsahující číslo aktuálně zpracovávaného řádku pro další použití. Makra `\cline` a `\noline` kontrolují aktuálně zpracovávaný sloupec a přidávají do proměnné `\customLines` primitivy pro ignorování definice sloupce a popřípadě pro vysázení horizontální linky. V případě že zpracovávaný řádek není poslední, přidávají do této proměnné také znak pro oddělení sloupců.

```
\newtoks\customLines \customLines{}
\def\cline{
  \global\advance\actcol by 1
  \ifnum\actcol<\colnum \addLines{\omit\vrule\hrulefill\vrule \separ}
  \else \addLines{\omit\vrule\hrulefill\vrule} \global\actcol=0
  \fi
}

\def\noline{
  \global\advance\actcol by 1
  \ifnum\actcol<\colnum \addLines{\omit \separ}
  \else \addLines{\omit} \global\actcol=0
  \fi
}

\def\addLines#1{
  \global\customLines\expandafter{\the\customLines#1}
}
```

```

\def\trcustom#1{
  \global\customLines{}\global\actcol=0#1\custLines
}

\def\custLines{
  \cr\the\customLines\cr \global\actcol=0
}

```

## 6.4 Další definovaná makra

Balík obsahuje několik dalších maker, která nejdou zařadit ani mezi makra tvořící tabulku, ani makra pomocí kterých si uživatel může nastavit právě tvořenou tabulku.

### 6.4.1 `\longtable`

Jedním z těchto maker, která by uživatel mohl potřebovat je `\longtable`. Makro slouží, jak již naznačuje jméno, pro vytvoření dlouhé tabulky, která bude delší než jedna strana. Syntaxe vypadá následovně:

```
\longtable\{zarovnání sloupců}{popis tabulky}{tělo tabulky}
```

Pro toto makro se nepoužívají příkazy `\table` a `\endtable`. Zarovnání sloupců se definuje stejně jako v ostatních tabulkách. Tělo tabulky má jistá omezení – v těle tabulky je možné použít pouze příkazy `\tr` a `\td`, některá další makra také fungují (např. `\tdalign`) ale mohou rozhodit vzhled tabulky nebo některá nemusí fungovat správně. V tabulce se také nesmí vyskytnout horizontální linky definované uživatelem. Tabulka má definované horizontální linky na začátku tabulky, pod titulem a na konci tabulky nebo stránky. Na začátku každé stránky se znovu opakuje titulek tabulky, což je v tomto případě první řádek tabulky.

V první řadě se definuje nový strut (výplněk definující velikost řádku) pro titulek tabulky, který je ohraničen horizontálními linkami. Při běžné velikosti by tento titulek vypal stísněně.

Nejprve se sestaví celá tabulka do `\box0`. Ten se následně rozebere na další boxy a v opačném pořadí než jsou, vloží do `\box2`. Do `\box4` uloží titulek, který se má opakovat na stránkách. Poté proběhne první kontrola místa na stránce, tedy jestli je na stránce dostatek místa pro titulek tabulky. Pokud ne, vloží se konec stránky a tabulka se začne sázet na nové stránce. Dále se pomocí makra `\printboxes` odeberají boxy zezadu z `\box2` a kladou se na stránku pomocí `\center` aby byly všechny vycentrovány. Po každém takto vloženém boxu makro `\whatfree` zkontroluje, jestli se na stránku vloží další box. V záporném případě vysází horizontální linku (pomocí `\crule`), vloží konec stránky (pomocí makra `\break`) a na začátek další stránky vloží horizontální linku, titulek, opět horizontální linku a začne sázet další boxy.

```

\def\strutT {      % Strut for a~title
  \vrule height1.1em depth.8em width0pt

```



```

}

\def\longtable#1#2#3{
  \caption{#2}\global\colnum=0 \global\actcol=0\hskip1em
  \bordertype{none}\goodbreak \bgroup
  \setbox0=\tablewidth{#1}{\strutT\relax#3}
  \setbox1=\vbox{\unvbox0 \setbox2=\vbox{}}\revertbox}
  \setbox2=\vbox{\unvbox2 \global\setbox4=\lastbox\unskip}
  \whatfree4 \advance\tmpdim by-.8pt \ifdim\tmpdim<\baselineskip
  \vfil\break \fi
  \offinterlineskip \crule\center4\crule \printboxes
  \crule \egroup \goodbreak
}

\def\whatfree#1{
  \tmpdim=\vsize \advance\tmpdim by-\pagetotal
  \advance\tmpdim by-\prevdepth \advance\tmpdim by-.4pt
  \advance\tmpdim by-\ht#1 \advance\tmpdim by-\dp#1
  \advance\tmpdim by-\ht5
}

\def\revertbox{
  \setbox0=\lastbox\unskip
  \ifvoid0 \else \global\setbox2=\vbox{\unvbox2\box0} \revertbox\fi
}

\def\printboxes{
  \setbox2=\vbox{\unvbox2 \global\setbox0=\lastbox\unskip}
  \ifvoid0 \else \whatfree0
  \ifdim\tmpdim<0pt \crule\vfil\break \crule\center4\crule \fi
  \center0 \nobreak \printboxes \fi
}

\def\crule{
  \centerline{\hbox to\wd4{\hrulefill}}\nobreak
}

\def\center#1{
  \centerline{\copy#1}\nobreak
}

```

### 6.4.2 `\spacesaroundtable`

Toto makro slouží pouze pro definování vertikální mezery nad a pod každou tabulkou. Toto nastavení se změní od okamžiku, kdy je uživatel použije. Ideální je tedy použít je na začátku dokumentu.

```
\def\spacesaroundtable#1{
  \global\verticalSpace=#1\relax
}
```

### 6.4.3 `\tablecaptions`

Toto makro slouží pro nastavení popisu tabulek. Tímto se myslí začátek popisku jako například (*Tabulka*, *Tab*, *Table*, apod). Makro je opět vhodné použít na začátku dokumentu. Číslování těchto tabulek je samozřejmě automatické.

```
\def\tablecaptions#1{
  \global\cap={#1 }
}
```

## 7 Ukázka použití

V této kapitole následuje ukázka několika tabulek vytvořených v nově vytvořeném balíku. Makro je určeno pro plain $\TeX$ , takže do dokumentu budou vloženy tabulky jako obrázky.

### 7.1 Jednoduchá tabulka

První ukázka kódu a porovnání s prostředím  $\backslash tabular$  bude pro jednoduchou tabulku se třemi sloupci zarovnanými vlevo, nastřed a vpravo, a čtyřmi řádky. V tabulce bude také použito makro pro hlavičku tabulky.

```
\table{center}{Jednoducha tabulka}
\borderstyle{cross}
\tablebody{lcr}{
  \tr
    \th{Mesic}
    \th{Zbozi}
    \th{Cena}
  \tr
    \td{Leden}
    \td{Notebook}
    \td{13000 Kc}
  \tr
    \td{Unor}
    \td{Skateboard}
    \td{2200 Kc}
  \tr
    \td{Duben}
    \td{Mobilni telefon}
    \td{4200 Kc}
}
\endtable
```

Tab 1. Jednoducha tabulka

Mesic	Zbozi	Cena
Leden	Notebook	13000 Kc
Unor	Skateboard	2200 Kc
Duben	Mobilni telefon	4200 Kc

Obr. 8: Jednoduchá tabulka

## 7.2 Tabulka s horizontálními linkami

Nyní je zde zobrazen zdrojový kód a tabulka, která se dá definovat pomocí maker `\trwithline`, tedy tabulka pouze s několika horizontálními linkami.

```
\table{right}{Tabulka s~horizontalnimi linkami}
\borderstyle{none}
\tablebody{ccr}{
  \trwithline
    \th{Mesic}
    \th{Zbozi}
    \th{Cena}
  \trwithline
    \td{Leden}
    \td{Notebook}
    \td{13000 Kc}
  \tr
    \td{Unor}
    \td{Skateboard}
    \td{2200 Kc}
  \tr
    \td{Duben}
    \td{Mobilni telefon}
    \td{4200 Kc}
  \trwithline
}
```

Tab 2. Tabulka s horizontálními linkami

Mesic	Zbozi	Cena
Leden	Notebook	13000 Kc
Unor	Skateboard	2200 Kc
Duben	Mobilni telefon	4200 Kc

Obr. 9: Tabulka s horizontálními linkami

## 7.3 Pruhovaná tabulka

V tomto příkladě je vytvořena tabulka, která má automaticky každý druhý pruh šedou barvou. Odstín šedé je nastaven na číslo 0.3 pomocí nepovinného parametru v příkazu `\striped`.

```

\table{left}{Pruhovana tabulka}
\striped[0.3]
\bordertype{none}

\tablebody{ccr}{
  \trwithline
    \th{Mesic}
    \th{Zbozi}
    \th{Cena}
  \tr
    \td{Leden}
    \td{Notebook}
    \td{13000 Kc}
  \tr
    \td{Unor}
    \td{Skateboard}
    \td{2200 Kc}
  \tr
    \td{Duben}
    \td{Mobilni telefon}
    \td{4200 Kc}
  \tr
    \td{Srpen}
    \td{Pracka}
    \td{8400 Kc}
  \tr
    \td{Prosinec}
    \td{Tablet}
    \td{7000 Kc}%
}
\endtable

```

Tab 3. Pruhoavana tabulka

Mesic	Zbozi	Cena
Leden	Notebook	13000 Kc
Unor	Skateboard	2200 Kc
Duben	Mobilni telefon	4200 Kc
Srpen	Pracka	8400 Kc
Prosinec	Tablet	7000 Kc

Obr. 10: Pruhoaná tabulka

## 7.4 Tabulka s definovanou šířkou

V tomto příkladě je vytvořena tabulka s definovanou šířkou 12 cm. V tabulce je také použito makro pro větší výšky řádku a nepovinný parametr v příkazu `\bordertype` pro definování větší šířky linek. Také je zde použit nadpis tabulky pomocí makra `\title`.

```
\table[left]{Tabulka s~definovanou sirkou}
\bordertype[1pt]{cross}
\rowheight{1.5em}
\tablebody[10cm]{ccr}{
  \title{Nadpis tabulky}
  \tr
    \th{mesic}
    \th{zbozi}
    \th{cena}
  \tr
    \td{leden}
    \td{notas a~}
    \td{14 kc}
  \tr
    \td{unor}
    \td{skejt}
    \td{2 kKc}
  \tr
    \td{cervenec}
    \td{jachticka}
    \td{3,4 kc}%
}
\endtable
```

Tab 4. Tabulka s definovanou sirkou

Nadpis tabulky		
mesic	zbozi	cena
leden	notas a	14 kc
unor	skejt	2 kKc
cervenec	jachticka	3,4 kc

Obr. 11: Tabulka s definovanou šířkou

## 7.5 Tabulka se sloučenými buňkami

Následuje jednoduchá tabulka se sloučenými buňkami v řádku a také ve sloupci.

```

\table{center}{Ruzne sloucene bunky}
\borderstyle{cross}

\tablebody{ccrr}{
  \tr
    \td{Dalsi}
    \td{Neco}
    \colspan{2}{c}{Colspan 2}
  \tr
    \td{Text}
    \td{Jeste neco jineho}
    \td{Neco}
    \td{Dalsi}
  \tr
    \td{}
    \td{skejt}
    \td{Nejak text}
    \td{Dalsi text}
  \trcustom{\noline \cline \cline \cline}
    \rowspan{c}{2}{Rowspan}
    \td{jachticka}
    \td{3,4 kc}
    \td{neco}
}
\endtable

```

Tab 5. Ruzne sloucene bunky

Dalsi	Neco	Colspan 2	
Text	Jeste neco jineho	Neco	Dalsi
Rowspan	skejt	Nejak text	Dalsi text
	jachticka	3,4 kc	neco

Obr. 12: Tabulka se sloučenými buňkami

## 7.6 Dlouhá tabulka přes více stran

```
\longtable{ccc}{A~tady je dlouha tabulka}{
  \th{Titulek 1}
  \th{Titulek 2}
  \th{Titulek 3}
\tr  % titulek
  \td{Jina data}
  \td{Dalsi data}
  \td{Ostatni data}
\tr
  \td{data}
  \td{data}
  \td{data}
\tr
  \td{data}
  \td{data}
  \td{data}
\tr
  \td{data}
  \td{data}
  \td{data}
\tr
  \td{data}
  \td{data}
  \td{data}
...
...
...
}
```



urna. Praesent faucibus eu purus sit amet accumsan. Vivamus viverra non mauris sed fermentum. Nulla sit amet purus pellentesque,

Tab 3. A tady je dlouha tabulka

<b>Titulek 1</b>	<b>Titulek 2</b>	<b>Titulek 3</b>
Jina data	Dalsi data	Ostatni data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data

---

<b>Titulek 1</b>	<b>Titulek 2</b>	<b>Titulek 3</b>
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data
data	data	data

---

Nunc mauris lacus, lacinia et gravida nec, faucibus eget arcu. Nunc placerat semper enim, rhoncus rhoncus lectus cursus et. Quisque mollis faucibus finibus. Fusce

Obr. 14: Dlouhá tabulka na poslední stránce

## 8 Diskuze a hodnocení

### 8.1 Hodnocení práce a návrhy pro zlepšení

Během této práce vznikl balík pro sazbu tabulek v systému plain $\text{T}_{\text{E}}\text{X}$ . Aktuálně balík nefunguje v systému  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ , to by mohla být příležitost pro další vylepšení. Tím, že balík převzal část sémantiky z jazyka HTML a další příkazy jsou postaveny na této sémantice, je jeho použití intuitivní i pro uživatele, kteří nemají přílišné zkušenosti se sazbou tabulek v systému založeném na  $\text{T}_{\text{E}}\text{X}$ , ale mají například nějaké zkušenosti s prací v HTML. Naopak, pro uživatele, kteří jsou zvyklí na definování tabulek pomocí všech nejznámějších balíku nebo přímo pomocí primitivu `\hline` bude tento způsob zápisu poněkud neobvyklý.

Balík splňuje většinu požadavků definovaných v kapitole 4. Uživatel má možnost definovat horizontální umístění tabulky na střed, vlevo i vpravo na stránku podle potřeby. Tabulka je defaultně umístěna v plovoucím prostředí. Uživatel má ale možnost toto nastavení zrušit. Co uživatel bohužel nemůže definovat je umístění plovoucího prostředí, tedy jestli se tabulka umístí na horní či dolní okraj stránky. Balík také umožňuje vysázení dlouhé tabulky přes více stran. Není implementována automatická kontrola, zda je tabulka přes více stran nebo ne a uživatel musí sám použít makro pro vícestránkovou tabulku. Tato kontrola by mohla být implementována dodatečně.

Uživatel má možnost definovat ohraničení tabulky. Toto definování je velice jednoduché a v případě že uživatel nevyžaduje vysázení nějaké atypické tabulky, balík se automaticky postará o vysázení linek. Balík obsahuje také příkaz pro vysázení horizontálních linek pouze v určitých místech, neobsahuje ale tuto možnost pro vertikální linky. Uživatel má také možnost definovat šířku těchto linek. Balík obsahuje pouze jeden styl linky a to je jednoduchá linka. To by také mohla být příležitost pro vylepšení balíku. Tedy definovat styl pro sazbu dvojitých linek. Další typy linek (čárkované apod.) se příliš nepoužívají a bylo by zbytečné je implementovat. Uživatel má také možnost definovat linky pouze pro určité buňky, což je velice výhodné při slučování řádků.

Výška a šířka sloupců v tabulce se vypočítávají automaticky. Uživatel má možnost definovat výšku řádků a šířku celé tabulky. V balíku chybí možnost definovat šířku jednotlivých sloupců a zalamování řádků přímo v buňce.

Slučování buněk je k dispozici jak ve vertikálním, tak horizontálním směru a makra pro tuto funkci jsou velice jednoduchá na zápis. Balík také obsahuje makro pro obarvení celého řádku (např. součtového řádku nebo hlavičky), sazbu nadpisu stránky a makro pro zarovnání textu na střed a sazbu tohoto textu polotučným písmem pro hlavičku tabulky.

Mezery v tabulce jsou pevně nastaveny tak, aby byly údaje v tabulce dobře čitelné. Uživatel má ovšem také možnost si je předefinovat. Stejně tak jako vertikální mezeru před a za tabulkou, která ji dělí od dalšího textu.

Uživatel si při definici počtu sloupců definuje také horizontální zarovnání v těchto sloupcích. Toto probíhá pomocí stejné definice jako ve všech nejvíce používaných balících. Toto zarovnání poté platí pro celý sloupec. V balíku je implementováno makro pro zarovnání pouze jedné buňky, která by měla být zarovnána jinak než daný sloupec. Nastavení vertikálního zarovnání je možné pouze pro buňku, která vznikne sloučením více buňek v jednom sloupci.

V balíku je implementováno několik základních barev. Aktuálně balík používá pouze šedou barvu pro obarvení řádků. Toto je další námět pro vylepšení balíku. Možnost použít další barvy pro obarvení řádků, popřípadě implementovat možnost obarvení jednotlivých buněk.

Součástí vytvořené tabulky je také popis tabulky, který je umístěn nad tabulkou. Tento popis obsahuje také čítač, který se automaticky zvyšuje s každou definovanou tabulkou.

## 8.2 Porovnání se stávajícími balíky

V porovnání s ostatními balíky, které existují a jsou nejvíce známé a používané má tento balík několik významných výhod. Mezi ty patří určitě jeho intuitivní a jednoduché ovládání, pomocí kterého může tabulky sázet i méně zkušený uživatel. Makra, která slouží pro různá nastavení i pro sazbu tabulky jsou pojmenována tak, aby bylo z jejich názvu jasná funkce tohoto makra. Další výhodou je, že balík přímo v sobě obsahuje prakticky všechnu důležitou funkcionalitu pro potřebu sazby tabulek. Uživatel tedy už nemusí hledat a připojovat další balíky (například `\multirow`, `\multicol`, `\xcolor` ...) pro další funkcionalitu, jako je slučování buněk ve sloupci, slučování buněk v řádku, balíky pro barvu apod. Pro sazbu tabulky na více stránek musí uživatel dokonce použít úplně jiný balík, než pro klasické tabulky.

Výhodou je určitě také z velké části automatizovaná sazba linek podle toho, jakou tabulku uživatel definoval. Nemusí tedy za každý konec řádku vkládat příkaz pro vytvoření horizontální linky. Zároveň ale tuto možnost má, a v případě, že chce mít linky pouze na určitých řádcích, je mu to umožněno. Co nový balík postrádá oproti stávajícím, je sazba vertikálních linek pouze za určitými sloupci. Toto by se mohlo zdát jako nevýhoda.

Další příjemná vlastnost je automatické vkládání znaku pro oddělení sloupců. Uživatel sice stále musí použít příkaz `\td{}` pro vytvoření každé buňky, nemusí ale řešit, zda je buňka první nebo poslední v řádku a jestli za ni, respektive před ni patří tento oddělovač. Tímto se zápis tabulky stává přehlednějším. Stejně tak je velká výhoda automatické sázení podbarvených řádků v případě, že uživatel chce mít každý druhý řádek podbarvený a nemusí tedy počítat řádky a určovat, který z nich je sudý (nebo lichý) a který bude mít podbarvení. Toto řešení aktuálně obsahuje v novém balíku chybu a pruhovaná tabulka může být vysázena pouze na levé straně stránky.

V novém balíku má také uživatel, pouhým použitím nepovinného parametru, možnost definovat celkovou šířku tabulky. Při použití stávajícího řešení musí použít

balík `\tabular*` nebo `\tabularx`. Tyto balíky mají ale šířku tabulky jako povinný parametr a uživatel tedy musí použít různé balíky pro sazbu tabulky bez a s definovanou šířkou tabulky. V čem nový balík zaostává je možnost určit šířku jednotlivých sloupců a také možnost zalomit delší text do více řádků v buňce.

Balík obsahuje několik maker, která slouží pouze pro zjednodušení práce uživateli. Je to například makro `\title{}`, které slouží pro sazbu záhlaví tabulky. V praxi tedy nedělá nic jiného, než že sloučí všechny bunky v řádku a vysází daný text. Nebo například makro `\th{}`, pomocí kterého může uživatel vysázat hlavičku tabulky a text v buňkách je vysázen na střed a polotučně. Další takový příklad je makro `\tdalign{}`, pomocí kterého uživatel může zarovnat text v buňce jinak, než je definováno pro daný sloupec.

Všech těchto věcí je uživatel schopen dosáhnout i pomocí stávajících balíčků, musel by si ovšem definovat vlastní makra, která by mu umožnila jednoduchý zápis, nebo použít definovaná makra  $\LaTeX$ u (např. `\textbf`, `\multicolumn` nebo `ltext`) pro sazbu polotučného písma a změnu zarovnání jedné buňky, a zápis tabulky by se tímto zneprůhlednil a prodloužil.

Jedna z nevýhod je také možnost použití balíku pouze v systému Plain $\TeX$ , který používá méně nezkušených uživatelů, než například  $\LaTeX$  a převod balíku pro použití v tomto systému by byla určitě dobrá volba. Balík zároveň ještě není dostatečně otestován a nejspíše se postupem času vyskytnou nějaké chyby. Pomocí dokumentace a hlavně této práce by ale neměl být problém tyto chyby odstranit.

## 9 Závěr

V rámci této práce byla analyzována a popsána typografická pravidla pro sazbu tabulek, typy tabulek a jejich sazba v dokumentech. Pomocí diplomové práce Ing. Petry Talandové Ph.D., Přístupy ve zpracování tabulek v systémech DTP, byly definovány vlastnosti a funkce, které by měl obsahovat plnohodnotný balík pro sazbu tabulek tak, aby dále nebylo zapotřebí využívat nějaký další. Také proběhla analýza několika stávajících balíků pro sazbu tabulek (*tabular*, *tabbing* apod.). Práce obsahuje také porovnání tvorby a zápisu tabulek v různých systémech a softwaru, které jsou zaměřeny na sazbu textu. Byly to následující – LibreOffice Writer, HTML, L<sup>A</sup>T<sub>E</sub>X.

Na těchto analýzách byl následně vytvořen balík pro sazbu tabulek v systému plainT<sub>E</sub>X, který se snaží o přiblížení a ulehčení práce v systému také uživatelům, kteří nemají se sazbou textu, respektive tabulek v plainT<sub>E</sub>Xu mnoho zkušeností a to tím, že tvorba tabulek je přívětivá a intuitivní a definovaná makra jsou velice podobná tagům, které se používají v jazyku HTML. Vytvořený balík obsahuje téměř všechnu funkcionalitu, která byla nadefinována pro takto určený balík. Práce obsahuje velice podrobný popis funkcí a definovaných maker, takže v případě dalšího vývoje balíku a doplnění funkcí může případný vývojář nastudovat makra a dále je vyvíjet.

Balík může nalézt uplatnění v projektu T<sub>E</sub>XonWeb, kde by měl pomoci v rozvoji funkcionality grafického definování tabulek a jejich zpětné načtení do grafické podoby z kódu. Jelikož je k balíku vytvořena také anglická dokumentace a balík samotný je psán v anglickém jazyce, může být také umístěn do archivu balíku pro systém T<sub>E</sub>Xa systémy založené na něm CTAN (Comprehensive T<sub>E</sub>XArchive Network), který v současné době obsahuje několik tisíc různých balíčků (<https://www.ctan.org/>).

## 10 Literatura

- BRAAMS, J a JURRIENS, T.2004. *The supertabular environment\** [online]. [cit. 2015-05-14]. Dostupné z: <http://ftp.cvut.cz/tex-archive/macros/latex/contrib/supertabular/supertabular.pdf>
- CARLISLE, D 2014 *The tabularx package\** [online]. [cit. 2015-05-14]. Dostupné z: <http://mirrors.nic.cz/tex-archive/macros/latex/required/tools/tabularx.pdf>.
- CARLISLE, D. 2014a. *The longtable package* [online]. [cit. 2015-05-14]. Dostupné z: <http://mirrors.nic.cz/tex-archive/macros/latex/required/tools/longtable.pdf>.
- DRUCBERT, J. P. F. 1998 *The Tabbing package* [online]. Toulouse [cit. 2015-05-14]. Dostupné z: <http://mirrors.nic.cz/tex-archive/macros/latex/contrib/Tabbing/Tabbing.pdf>.
- HTML 5.1 Nightly: 1 Introduction. 2015. *World Wide Web Consortium (W3C)* [online]. [cit. 2015-05-13]. Dostupné z: <http://www.w3.org/html/wg/drafts/html/master/introduction.html>.
- KNUTH, D. E. 1986. *The TEXbook*. Boston: Addison-Wesley, ix, 483 s. ISBN 02-011-3448-9. .
- LaTeX v kostce - vytváření tabulek*. 2001 SOPUCH, P LaTeX v kostce [online]. [cit. 2015-05-14]. Dostupné z: <http://www.it.cas.cz/manual/latex/i.html>.
- OLŠÁK, P. 2001 *TeXbook naruby*. 2. vyd. Brno: KONVOJ. ISBN 80-7302-007-6. Dostupné také z: <ftp://math.feld.cvut.cz/pub/olsak/tbn/tbn.pdf>.
- OLŠÁK, P. 2014. *OPmac - rozšiřující makra plain TeXu*. OPmac [online]. [cit. 2015-05-14]. Dostupné z: <http://petr.olsak.net/opmac.html>.
- POP, P, POP, V a FLÉGER, J *Sazba I: ruční sazba pro 1. a 2. ročník střední průmyslové školy grafické a střední odborné učiliště polygrafické*. 2. vyd. Praha: SPN, 1989, 185 s. .
- PRICHYSTAL, J *eXonWeb* [online]. [cit. 2015-05-13]. Dostupné z: <https://tex.mendelu.cz/navod.pl>.
- SATRAPA, P 2011 *LaTeX pro pragmatiky* [online] Verze 1.1. [cit. 2015-05-13]. Dostupné z: <ftp://ftp.tex.ac.uk/tex-archive/info/czech/latex-pro-pragmatiky/latex-pro-pragmatiky.pdf>.
- TALANDOVÁ, P 2006 *Přístupy ve zpracování tabulek v systémech DTP*. Brno. Diplomová práce. Mendelova zemědělská a lesnická univerzita v Brně, Provozně ekonomická fakulta. Vedoucí práce Doc. Ing. Jiří Rybička, Dr.,... .

## **Přílohy**



## A Srovnání kódu

Kód prostředí *tabular*

```
\begin{tabular}{|l|c|r|}
\hline
\multicolumn{3}{|c|}{\textbf{Vzdálenosti měst}} \\ \hline
\textbf{Město A} & \textbf{Město B} & \textbf{Vzdálenost [Km]} \\ \hline
Brno & Praha & 204 \\ \hline
Brno & Prostějov & 61 \\ \hline
\end{tabular}
```

Kód nového balíku

```
able{center}{Jednoduchá tabulka}
\borderstyle{cross}
\tablebody{lcr}{
  \title{Vzdálenosti měst}
  \tr
    \th{Město A}
    \th{Město B}
    \th{Vzdálenost [Km]}
  \tr
    \td{Brno}
    \td{Praha}
    \td{204}
  \tr
    \td{Brno}
    \td{Prostějov}
    \td{64}
}
\endtable
```

Vzdálenosti měst		
Město A	Město B	Vzdálenost [Km]
Brno	Praha	204
Brno	Prostějov	64

## **B Uživatelská dokumentace v anglickém jazyce**

# Table package – diploma thesis

Jenis Josef

20. května 2015

## Introduction

There are lots of packages for creating tables in  $\text{T}_{\text{E}}\text{X}$ / $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  systems now. But lots of them are difficult for using. The usual  $\text{T}_{\text{E}}\text{X}$ / $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  way, or if you want macros/commands, for this are very specific for these systems. And I know from my own experiences and from experiences of friends of mine, that it's a little bit difficult to get used to this way at the beginning.

So the vision was to create a new package with different commands, which are similar to commands in some other programming language, or better text markup language. And there is one clear choice – HTML. Almost everybody knows this language and almost everybody has some experiences with it. Even economics students, agriculture or some general high schools has some experiences with HTML from the classes. So I decided to build up this package based on HTML tags for tables. This package is independent on some other  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  packages and You can use it just in Plain $\text{T}_{\text{E}}\text{X}$ .

## Using

### Basic commands

This package should be designed to be user-friendly. That means, creation of tables should be as simple as possible. The first command You have to use is

```
\table[nofloat]{align}{caption}
```

First parameter is optional and only one option is allow – [nofloat]. Default environment is the floating one. By this parameter you can change this settings. The second parameter is mandatory and it's for settings of horizontal alignment. You can use values – left, right and center. The last parameter is also mandatory and it's a caption of the table.

Example of this command: `\table[nofloat]{center}{Table of yields}`

The second basic command is

```
\tablebody[width]{columns align}{body of the table}
```

First parameter is optional. You can define the width of the table. Second parameter should contain characters like - l/r/c depends on what alignment you want to use (left/right/center). How many columns you want in the table, is how many characters do you need in this parameter. Third parameter contains the body of the table. Structure of this body is described below.

For example for a table with four columns aligned to the left, left, right and center:

```
\tablebody{llrc}{tablebody}
```

Last from basic commands is

```
\endtable
```

This command is important for closing the floating environment etc.

### Extended macros for the table customization

You can use some other commands between basic commands `\table` and `\tablebody` for more customization of a table. Some of them are more important and you will use them more or less in every table definition and some of them just in some special cases. These commands are:

- `\bordertype[x]{border type}` – you can define the style of the table. The first parameter is optional, the second one is mandatory. You can define the width of the border lines in the table by using the first parameter. In the second parameter you have to define the type of the lines. You can use four values: none/cross/verical/horizontal. None is table without all lines, cross means horizontal and vertical lines, options vertical and horizontal are just horizontal or vertical lines.  
Example: `\bordertype[3pt]{cross}`
- `\striped[0–1]` – if you will use this option, every second row in the table will have a grey background. Parameter of this macro is optional. If you will use the parameter, the shade of grey will be defined like a value of this parameter. You can use values between 0 and 1, where 0 is totally white, and 1 is totally black color. Default value is 0.2. You have to use this command before the macro – `\bordertype`
- `\cellspaces{x}` – you can define a spaces before and after the text in cells by using this command.  
Example: `\cellspaces{5em}`

- `\rowshheight{x}` – this command is used to define the height of the rows in the table.  
Example: `\rowshheight{16pt}`
- `\vertlinewidth{x}` – command is used to define the width of vertical lines.  
Example: `\vertlinewidth{3pt}`
- `\horlinewidth{x}` – command is used to define the width of horizontal lines.  
Example: `\horlinewidth{1pt}`

### Content of table body

Body of the table is the second parameter of the command

```
\tablebody{columns align}{body of the table}
```

As mentioned, the vision of the package was to use similar commands like the HTML ones. There are defined the following commands:

- `\title{title}` – this macro automatically merges all cells in the row and typesets the content of the parameter into this cell in bold text and aligned in center.
- `\tr` – macro for new row in table. It automatically adds also a line according on settings from command `\borderline`
- `\trwithline` – macro for a new row in a table with a line.
- `\trwithcolor{0–1}` – new table row with a grey background. Shade of grey color depends on the number in parameter – 0 is totally white, 1 is totally black.
- `\th{content of the cell}` – it's like a table head. Text in this cell is bold and centered.
- `\td{content of the cell}` – table cell.
- `\tdalign{l/r/c}` – table cell aligned to the left/right/center.
- `\colspan{x}{l/r/c}{content of the cell}` – You can merge x cells with this macro. Second parameter is for alignment of the cell. If You will use this command for merging for example three cells, you have to use two commands `\td` less in that row.
- `\rowspan{x}{u/c/d}{content of the cell}` – this command is for merging cells in one column. Macros has also three mandatory parameters. First one is for declaring how many cells do you want to merge. You can define the vertical alignment by the second parameter and the third one should be the content of the cell.  
Example: `\rowspan{3}{c/}{Content}`. For using this command (for merging three cells in the column) you have to use command `\td{}` with empty content in two rows above.

- `\trcustom{definition of horizontal lines}` – You can use this command if you want to typeset horizontal lines just for some columns. In the parameter can be command `– \cline` for columns when You want to have a line or `\noline` for columns where You don't want to have a horizontal line.  
Example: `\trcustom{\cline \noline \cline}` for having horizontal line just in first and third column.

### Some other commands

Package provides some more commands.

```
\longtable\{columns align\}{caption}\{body of the table\}
```

It's a command for a table longer than one page. All parameters are mandatory. This table has specific environment and configuration so no more commands are needed there. Just commands for definition of table body (`\tr`, `\th`).

```
\spacesaroudtable{x}
```

You can define spaces above and below the table. Default value is `– 0.5 cm`.

```
\tablecaptions{caption name}
```

You can define the caption of the tables. For example `Tab.` `Table.` or whatever you want.

## Examples

```
\table{center}\{Elementary table\}
\bordertype{cross}
```

```
\tablebody{ccr}\{
  \tr
    \title{My purchases}
  \tr
    \th{Month}
    \th{Goods}
    \th{Price}
  \tr
    \td{January}
    \td{Laptop}
    \td{15000 CZK}
  \tr
    \td{February}
```

```
        \td{Skate}
        \td{2000 CZK}
    \tr
        \td{June}
        \td{Bicycle}
        \td{7500 CZK}
}
\endtable
```

```
\table{center}{Table with some horizontal lines}
\borderstyle{none}
\tablebody{ccr}{
    \trwithline
        \th{Month}
        \th{Goods}
        \th{Price}
    \trwithline
        \td{January}
        \td{Laptop}
        \td{15000 CZK}
    \tr
        \td{February}
        \td{Skate}
        \td{2000 CZK}
    \tr
        \td{June}
        \td{Bicycle}
        \td{7500 CZK}
    \trwithline
}
\endtable
```

```
\longtable{ccc}{Some long table}{
    \th{Title of 1th column}
    \th{Title of 2nd column}
    \th{Title of 3rd column}
    \tr
        \td{data}
        \td{data}
        \td{data}
    \tr
        \td{data}
        \td{data}
        \td{data}
}
```

```
\tr
  ..some more data...

\table[right]{Tabulka with defined width of a table, width of lines, height of rows a
\borderstyle[2pt]{cross}
%\linewidth{2pt}
%\lineheight{2pt}
\rowheight{15pt}
\cellspc{1cm}

\tablebody[12cm]{ccr}{
  \tr
    \th{Month}
    \th{Goods}
    \th{Price}
  \tr
    \td{January}
    \td{Laptop}
    \td{15000 CZK}
}
\endtable

\table[left]{Striped table}
\striped[0.6]
\borderstyle{none}
\tablebody{ccr}{
  \tr
    \th{Month}
    \th{Goods}
    \th{Price}
  \trwithline
    \td{January}
    \td{Laptop}
    \td{15000 CZK}
  \tr
    \td{February}
    \td{Skate}
    \td{2000 CZK}
  \tr
    \td{June}
    \td{Bicycle}
    \td{7500 CZK}
  \tr
    \td{June}
```



```
        \td{Bicycle}  
        \td{7500 CZK}  
    \tr  
        \td{June}  
        \td{Bicycle}  
        \td{7500 CZK}  
    }  
\endtable
```

## C Přiložené CD

Přiložené CD obsahuje:

- Balík napsaný ve formátu .tex,
- anglickou dokumentaci ve formátu .tex a formátu PDF,
- práci v elektronické podobě.