

Dokumentační systém Mf

Martin Mandák

Obsah

1. Definiční soubor maker MDef	2
1. 1. Co je zahrnuto	2
1. 2. Dokumentační platforma	3
1. 3. Hierarchie dokumentace	3
2. Uživatelské příkazy dokumentace	4
2. 1. Dělení na knihy	4
2. 1. 1. Popis hlavního souboru	4
2. 1. 2. Popis knihy	4
2. 2. Nadpisy	5
2. 3. Odrážky a číslování	5
2. 4. Rovnice	5
2. 5. Tabulky	5
2. 6. Obrázky	6
2. 7. Reference	6
2. 8. Prostředí verbatim	7
2. 9. Dokumentování zdrojových kódů programů	7
2. 9. 1. Makra pro vložení C++ souboru	7
2. 9. 2. Syntaxe dokumentace C++ souboru	8
2. 9. 3. Makra pro vložení Tcl souboru	8
2. 10. Poznámka	8
2. 11. Obsah	9
2. 12. Odkazy na literaturu	9
2. 13. Ostatní příkazy	9
3. Literatura	10
4. Popis algoritmů	11
4. 1. Test načtení souboru	11
4. 2. Inicializace proměnných (knihy)	11
4. 3. Inicializace proměnných (kapitoly)	11
4. 4. Definice makra pro knihu	12
4. 5. Kapitoly	12
4. 6. Zpracování referencí na knihu a kapitoly	14
4. 7. Výčty a odrážky	15
4. 8. Deklarace souboru referencí a stránková reference	16
4. 9. Obsah	17
4. 10. Seznam literatury	18
4. 11. Rovnice	19
4. 12. Různé definice	19
4. 13. Poznámky	21
4. 14. Tabulky	21
4. 15. Obrázky	22
4. 16. Verbatim	22
4. 16. 1. Nečíslovaný verbatim	22
4. 16. 2. Verbatim s číslováním řádek	23
4. 16. 3. Vkládání C++ a Tcl zdrojových souborů	23
4. 17. Parametry \TeX tu	24
4. 18. Záhloví a pata stránky	25
4. 19. Načtení referencí	25
4. 20. Titulní stránka knihy	25
4. 21. Titulní stránka dokumentace	26
5. Systém autodokumentace	26
5. 1. Princip autodokumentace	26

1. Definiční soubor maker MDef

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %%
3 %%   Kodovani:      $Enc: win1250$
4 %%   mdef.tex:     definiční soubor maker MDef
5 %%   verze:        2.0
6 %%   autor:        Martin Mandák (martin@mandak.net, http://www.mandak.net)
7 %%   TeX system:   plainTeX a odvozené systémy
8 %%
9 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
10 %%
11 %%   Tento soubor maker vznikl při tvorbě dokumentace k projektu Stabilita
12 %%   letounu. Makra z něj můžete libovolně šířit, používat a modifikovat
13 %%   jen s podmínkou, že bude ve Vašem dokumentu uveden autor tohoto
14 %%   souboru maker stejně jako každý, kdo a kdy provedl nějakou změnu.
15 %%
16 %%   Soubor je schopen sám sebe dokumentovat spuštěním dávky
17 %%           csplain mdef.tex
18 %%           nebo pdfcspain mdef.tex
19 %%   Další informace zjistíte nejlépe z takto vzniklé dokumentace...
20 %%
21 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

Tato sbírka maker tvoří ucelený systém pro vytváření dokumentace. Je napsána takovým způsobem, že umožňuje „dokumentovat sama sebe“. Vznikla z důvodu zbytečné komplikovanosti systému \LaTeX , který sice má velké možnosti přizpůsobení, ale působí značně těžkopádným dojmem, mnoho úprav podle mých představ by znamenalo třeba i zásahy do samotného zdroje \LaTeX u nebo instalaci balíků, které často dělají spoustu věcí, které vůbec nepotřebují a zbytečně tak zabírají místo v paměti, zpomalují zpracování dokumentu a ztěžují hledání případných chyb.

Jako řešení tohoto problému jsem zvolil napsání vlastní sady maker v plain \TeX u, která jsou snadno přizpůsobitelná mým požadavkům jejich přímou úpravou (na rozdíl od \LaTeX u zde vždy vím, co dělám). Vygenerováním této dokumentace jste dostali návod, jak používat zde definovaná makra, ale i vysvětlující komentář, z nějž by mělo být zřejmé, co a jak upravit, abyste získali vlastní požadovaný vzhled dokumentu. Pokud budete mít vlastní specifické požadavky, pak vám tento můj výtvar snad přijde vhod jako studijní materiál pro vytvoření vlastních maker.

Pokud budete mít jakékoliv dotazy nebo připomínky k tomuto souboru maker nebo i obecně k \TeX u, můžete mě kontaktovat na emailové adrese martin@mandak.net. Nové verze a informace k tomuto i jiným programům naleznete na adrese <http://www.mandak.net>. Současná dokumentace odpovídá stavu před implementací pdf odkazů.

1. 1. Co je zahrnuto

Zde uvádím stručný seznam možností tohoto balíku maker:

- Dělení dokumentu na knihy, 5 úrovní kapitol
- 3 úrovně odrážek nebo číslovaných seznamů
- číslované rovnice, obrázky a tabulky
- verbatim prostředí v několika verzích, vkládání dokumentovaného C++ nebo Tcl souboru
- generování několika typů obsahů
- vkládání poznámek, různé značky k různým účelům (viz. kapitolu 2. 13. Ostatní příkazy)
- křížové reference na knihu, kapitolu, rovnici, stránku, obrázek, tabuku
- všechny odkazy jsou při kompilaci pdf \TeX em funkční (od MDef verze 2, dokumenty napsané pro starší verzi stačí překompilovat s novou verzí maker)

Do budoucna:

- načtení obsahu vždy na začátku dokumentu kvůli pdf outlineům. Momentálně se načítá jen při obsahu, takže pokud by obsah byl na konci dokumentace, outlinky by asi byly 1rozměrné...
- Dodělat reference mezi různými objekty definované uživatelem. Pro odkazy na sekci již jsou, ale jen pro počítání s bookno, samostatně byly rozladěny a momentálně by asi způsobovaly chybu.

1. 2. Dokumentační platforma

Protože tento dokumentační systém je součástí multiplatformního projektu, věnoval jsem zpracování dokumentace velkou pozornost. Vzhledem k univerzálnosti byla vybrána jako primární platforma pro dokumentaci systém $\text{T}_{\text{E}}\text{X}$, který poskytuje špičkové výstupy v oblastech publikace především matematiky. Jeho velkou předností je, že výstup je dvi nebo postscript (ideální pro tisk), ale především dnes už také pdf jako dnes asi nejuniverzálnější formát pro dokumentaci přenášenou mezi platformami při dokonalém zachování formátování všech grafických prvků i kódování národních znakových sad. Výhoda $\text{T}_{\text{E}}\text{X}$ u je také možnost algoritmizovat zpracování zdrojových textů, čehož jsem s výhodou využili při naprogramování zpracování dokumentace vepsané přímo do zdrojového C++ kódu.

Další potenciální alternativní výstupy generované ze zdrojových kódů $\text{T}_{\text{E}}\text{X}$ u předpokládám xml resp. html, unixové manuálové stránky, případně help soubor pro MS Windows. Vzhledem k co nejsnadnějšímu přizpůsobení vzhledu a konverzím dokumentace jsem zamítl připravené dokumentační formáty jako $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, protože při komplikovanější přizpůsobování použitých stylů nám vyšlo jako jednodušší vytvořit si vlastní od základu přizpůsobený našim požadavkům a v případě potřeby mnohem snadněji upravitelný.

Zvolili jsme tedy vytvoření vlastního formátu na bázi formátu plain $\text{T}_{\text{E}}\text{X}$, jehož použití je takové, aby byly snadné i požadované konverze do případných jiných formátů než dvi, ps a pdf. Vypracování nástrojů pro alternativní výstupy je otázkou budoucího vývoje.

1. 3. Hierarchie dokumentace

Dokumentace je založena na formátu plain $\text{T}_{\text{E}}\text{X}$, takže fungují klasická plainová makra, ale ta by se neměla přímo používat z důvodu zachování plné kontroly nad vzhledem dokumentu při generování všech formátů dokumentace.

Základní dělení dokumentace je na tzv. knihy, z nichž každá by měla tvořit určitou část dokumentace takovým způsobem, aby se dala použít jak samostatně, tak i jako součást celé kompletní dokumentace. Každá kniha je dále dělena na kapitoly, přičemž má možnost hierarchického dělení až na pět úrovní kapitol.

Systém je navržen tak, aby bylo možno vytvořit jak kompletní dokumentaci (všechny knihy v jednom celku) tak dílčí dokumentace, kdy potřebujeme pouze určitou knihu, bez jakýchkoliv úprav zdrojových kódů dokumentace, jen spuštěním $\text{T}_{\text{E}}\text{X}$ procesoru pro jiný soubor.

Všechna makra a nastavení jsou uložena v souboru `mdef.tex`, který se příkazem `\input` vloží za začátku každé knihy i celé dokumentace (podrobněji viz. kapitolu 2. 1. [Dělení na knihy](#)). Struktura kompletního dokumentu pak vypadá takto:

- celá dokumentace
 - první kniha
 - kapitoly a texty první knihy...
 - druhá kniha
 - kapitoly a texty druhé knihy...
 - další kniha...
 - kapitoly a texty další knihy...

Přičemž každá kniha může stát také samostatně a nezávisle na ostatních.

2. Uživatelské příkazy dokumentace

V následující části jsou stručně popsány příkazy dokumentace definované v souboru `mdef.tex`. Při psaní tohoto popisu jsem předpokládal, že čtenář je obeznámen se základními principy a vlastnostmi systému `plainTeX` alespoň na úrovni knihy [3].

2. 1. Dělení na knihy

Každá kniha tvoří buďto část celé dokumentace, anebo může existovat samostatně. Tím je vlastně dáno, že existují z hlediska našeho dokumentačního systému čtyři typy souborů:

1. kořenový soubor celé dokumentace
(viz. kapitolu 2. 1. 1. [Popis hlavního souboru](#))
2. soubory představující kořenové soubory jednotlivých knih
(viz. kapitolu 2. 1. 2. [Popis knihy](#))
3. další soubory, na něž se soubory předchozích dvou skupin odkazují. Jde o další `TeX` soubory, ale i grafické či jiné soubory použité v dokumentaci.
4. soubor `mdef.tex`, který je vlastně speciálním případem 3. skupiny souborů. Má smysl, aby byl obsažen pouze na začátku souborů 1. a 2. skupiny, jinde jeho `\input` nemá smysl, ale jeho `\input` navíc nevádí — soubor se „sám kontroluje“, aby byl zahrnut pouze při prvním `\input mdef` a v dalších je jeho obsah vynechán.

2. 1. 1. Popis hlavního souboru

Hlavní soubor je vždy jen jeden pro celý dokumentační systém, resp. pro určitou kombinaci spojení několika různých knih a příkazem `\input mdef` je z něj volán soubor s makry a definicemi dokumentačního systému (podrobněji viz. kapitolu 4. [Popis algoritmů](#)).

Poté se spouští makro `\maindoc{nadpis 1}{nadpis 2}{anotace}`, které systému říká, že se kompiluje více knih do jednoho celku a dále vysází titulní stranu dokumentace (`nadpis 1` a `nadpis 2` jsou uprostřed na dvou řádcích) a dále stranu s anotací, kde mohou být uvedeni autoři, copyright apod. Syntaxe `anotace` je taková, že v ní nesmí být vícekrát řádkováno (`\par`), tisku na více řádek se docílí makrem `\cr` (jako v prostředí `\halign`), které znamená vždy jedno zařádkování.

Za makrem `\maindoc` se příkazem `\input` volají kořenové soubory knih a makra pro globální obsah, viz. kapitolu 2. 11. [Obsah](#). Možné je zde vkládat i lokální obsahy knih vždy za příslušným souborem knihy (v tom případě nebude takový obsah zahrnut při samostatném zpracování knihy). Dokumentace je ukončena plainově příkazem `\bye` nebo `\end`.

2. 1. 2. Popis knihy

Každá kniha může být zpracována programem `TeX` či jiným procesorem samostatně na jeho bázi nebo jako součást většího dokumentačního celku.

Na začátku každého souboru knihy je `\input mdef`, který zavede příslušná makra pouze v případě, že se kompiluje pouze jedna kniha. Následně se volá makro `\kniha{nadpis 1}{nadpis 2}{autor}`, kde `nadpis 1` a `nadpis 2` mají podobný efekt jako u příkazu `\maindoc`, `nadpis 1` bude použit v obsahu nebo v seznamu knih. Autor (či jiná vhodná poznámka jako datum apod.) bude uveden v pravém dolním rohu titulní strany knihy (neregeneruje se samostatná stránka s anotací jako u `\maindoc`).

Za makrem `\kniha` následuje samotná dokumentace. Ta má všechny číslované objekty číslovány vždy od jedné (jak číslování kapitol, tak rovnic, tabulek...), je-li kniha součástí většího celku, je před každé číslování přidána římská číslice vyjadřující číslo aktuální knihy. Pouze číslování stránek je nezávislé na knihách a je vždy od začátku celého dokumentu (ať knihy nebo kompletní dokumentace) až do konce kontinuální.

Soubor představující kmenový soubor knihy je ukončen makrem `\konecknihy`, které je definováno tak, aby se ukončilo buď pouze načítání souboru (a pokračuje se nadřazeným souborem hlavní dokumentace) nebo kompletně ukončil běh \TeX u (v případě zpracování pouze jedné knihy). Makro `\konecknihy` má vliv pouze na ukončování načítání souboru \TeX em, z hlediska dělení dokumentu na knihy nemá jiný význam, aktuální kniha pokračuje až do dalšího makra `\kniha` nebo ukončení celého dokumentu.

2. 2. Nadpisy

K dispozici je pět úrovní nadpisů, všechny jsou hierarchicky číslovány a je tedy potřeba postupovat od vyšší úrovně k nižší (jinak se vynechaná úroveň nadpisu projeví jako číslo 0 v číslování). Postupuje se v tomto pořadí:

- `\section{Nadpis první úrovně}`,
- `\subsection{Nadpis druhé úrovně}`,
- `\subsubsection{Nadpis třetí úrovně}`,
- `\partsection{Nadpis čtvrté úrovně}`,
- `\ppartsection{Nadpis páté úrovně}`,
- `\nosection{Nadpis úvodníku}` — ten je na hierarchii nezávislý.

Všechny úrovně nadpisů jsou implicitně uváděny v obsahu a na kterýkoliv z nadpisů se lze odkazovat pomocí `\seclabel` a `\secref`, viz. kapitolu 2. 7. Reference.

Makro `\nosection` není číslováno, je zahrnuto z důvodu potřeby nečíslovaných kapitol na začátku dokumentace (mimo knihu). V obsahu bude uvedeno na stejné úrovni jako `\section`.

2. 3. Odrážky a číslování

Makra pro odrážky a číslování vypadají z pohledu uživatele maker dokumentačního systému tak, že „vnitřek“ seznamu vypadá stále stejně a pokud se změní uvozující značka, lze jednoduše změnit číslovaný seznam na seznam s odrážkami nebo obráceně.

Dostupné jsou tři úrovně odrážek nebo číslování. Prostředí pro výčty resp. pro odrážky začíná příkazem `\beginum` a končí příkazem `\endnum` resp. `\begitem` a `\enditem`.

V obou prostředích se pro jednotlivé položky používají příkazy `\item`, `\iitem` a `\iiitem` pro příslušný prvek první, druhé a třetí úrovně.

V prostředích pro odrážky a výčty by se neměl vyskytovat `\par`, tedy dvojité zařádkování, protože tím dochází k chybnému přenastavení parametrů sazby.

2. 4. Rovnice

Rovnice a matematika jsou jednou z nejsilnějších stránek systému \TeX . Zahrnul jsem tedy několik způsobů pro jejich vkládání:

1. Přímo do textu pomocí oddělovačů `$`,
2. klasicky „plainově“ pomocí `$$`, příslušná rovnice pak nebude číslována,
3. po vzoru \LaTeX u uzavřená mezi `\[` a `\]`, rovnice pak bude očíslována a lze se na ni odkazovat.

Syntaxe uvnitř rovnice musí odpovídat plain \TeX ovým zvyklostem.

2. 5. Tabulky

Tabulky v mém systému fungují tak, že se příkazem `\zactab` zahájí prostředí tabulky. Hned za tímto příkazem je vše až do konce řádku bráno jako popis k tabulce. Na dalším řádku je popis tabulky podle syntaxe prostředí `\halign`. Za popisem tabulky na dalším řádku jsou již vlastní data tabulky.

Políčka tabulky mohou být oddělena buď klasicky \TeX ově znakem `&` nebo tabulátorem, což umožňuje použít např. tabulky exportované StarCalcem, Excelem nebo jiným tabulkovým procesorem, databázovým programem, programem pro měření a sběr dat apod. Řádky tabulky se neukončují sekvencí `\cr`, ale obyčejným zařádkováním. Při použití `\cr` (není doporučeno) nesmí být bezprostředně před ani za touto značkou zařádkováno.

Tabulkové prostředí je ukončeno příkazem `\kontab` umístěným na samostatném řádku.

Další verzi tabulky je její obdoba bez popisku, může sloužit např. k zápisu soustav rovnic apod. Její číslo není započítáváno do počtu tabulek. Vnitřní syntaxe této tabulky je stejná jako v předchozím případě, pouze místo `\zactab` a `\kontab` se použije `\zacmat` a `\konmat` a popis tabulky na prvním řádku se vynechává (tedy je rovnou zařádkováno).

U obou typů tabulek se horizontální linka vkládá příkazem `\linka` na samostatném řádku. Chceme-li vložit dvojitou linku (např. pro oddělení záhlaví), použijeme místo `\linka` příkaz `\dlinka`. V současné verzi je tabulka uzavřena v jediném boxu, což v praxi znamená, že se musí vejít na jednu stránku, jinak dojde k jejímu přetečení přes okraje stránky.

2. 6. Obrázky

Dokumentace umožňuje i vkládání obrázků. Obrázky musí být připraveny v různých formátech podle požadovaného výstupu dokumentace. Pro postscriptový výstup je potřeba mít obrázky ve formátu eps (encapsulated postscript) a to jak rastrové, tak vektorové. Pro pdf výstup se nejlépe použije formát pdf (vektorová grafika) nebo pro rastrovou grafiku jpg či tiff. O tom, v jakém formátu se bude hledat obrázek při zpracování pdf \TeX em rozhoduje makro `\obrazekpripřípona`, kde `přípona` je přípona vkládaných grafických obrázků a implicitně je pdf. Po zavolání tohoto makra se přípona mění globálně a bude se používat pro obrázky až do konce zpracování dokumentu nebo do dalšího zavolání makra `\obrazekprip` s jinou příponou.

Chceme-li mít obrázek v jiné než „přirozené“ velikosti, můžeme použít registry `\obrazekx` resp. `\obrazeky`, které jsou typu `dimen` a lze jim přiřadit šířku resp. výšku (případně oba parametry) následujícího obrázku. Nulová hodnota znamená původní velikost a na tuto hodnotu jsou oba registry nastaveny ihned po vložení obrázku.

Použití makra pro vložení obrázku je následující: `\obrazek{soubor}{Popisek obrázku}`, kde `soubor` je jméno souboru bez přípony, tu si systém připojí pro příslušný výstup jakou bude potřebovat (a soubor v příslušném formátu ovšem musí existovat).

2. 7. Reference

V jednotlivých knihách jsou zavedeny křížové reference na různé objekty. Odkazovat se lze na rovnice, tabulky, stránky, obrázky, kapitoly, knihy, fungují dopředné i zpětné reference. Všechny reference spolehlivě fungují po trojím průchodu \TeX procesorem. Druhý průchod je nutný z důvodu vybavení i dopředných referencí (načtení souboru referencí z předchozího běhu), třetím průchodem dojde k případnému srovnání odkazů které se změnily např. v důsledku prodloužení obsahu na více stránek po druhém průchodu \TeX procesorem.

Příslušnou značku $\backslash\text{label}$ je nejlepší umísťovat těsně za příslušný objekt, na nějž chceme odkazovat. Umístění před odkazovaný objekt způsobí, že značka bude odkazovat na předchozí objekt stejného typu. Konkrétní příkazy pro návěští / odkaz jsou:

knihy	$\backslash\text{knihalabel}\{\text{reference}\}$	$\backslash\text{kniharef}\{\text{reference}\}$
kapitoly	$\backslash\text{seclabel}\{\text{reference}\}$	$\backslash\text{seceref}\{\text{reference}\}$
rovnice	$\backslash\text{eqlabel}\{\text{reference}\}$	$\backslash\text{eqref}\{\text{reference}\}$
stránky	$\backslash\text{strlabel}\{\text{reference}\}$	$\backslash\text{strref}\{\text{reference}\}$
tabulky	$\backslash\text{tablabel}\{\text{reference}\}$	$\backslash\text{tabref}\{\text{reference}\}$
obrázky	$\backslash\text{obrlabel}\{\text{reference}\}$	$\backslash\text{obrref}\{\text{reference}\}$

Systém \LaTeX má definovanou proměnnou $\backslash\text{globalrefs}$, která je implicitně nula a v tom případě jsou všechny reference vždy lokální v rámci jedné knihy (tedy lze mít i při jednom zpracování dokumentace více stejných značek, každou v jiné knize, i když to zásadně nedoporučuji). Nastavíme-li $\backslash\text{globalrefs}$ na jedničku, budou odkazy kapitol všech úrovní, odkazy na stránku a odkazy na literaturu brány jako globální a lze se odkazovat na kapitoly, stránky a položky literatury i mezi knihami (ovšem v případě stejných značek bude \TeX hlásit chybu).

2. 8. Prostředí verbatim

Prostředí verbatim funguje tak, že vše, co je napsáno mezi zadanými značkami, je vysázeno bez formátování — tedy tak, jak je zapsáno ve zdrojovém textu. Toho se využívá např. při sazbě zdrojových kódů programů apod. Za tímto účelem je v systému \LaTeX připraveno několik prostředí:

1. „obyčejný“ verbatim, začíná příkazem $\backslash\text{verb}$ a končí příkazem $\backslash\text{everb}$, oba příkazy je nejlepší psát samostatně stojící na řádku,
2. verbatim s číslováním řádek pro snadnější orientaci ve zdrojových textech. K zahájení a ukončení prostředí se použijí příkazy $\backslash\text{cverb}$ a $\backslash\text{ecverb}$. Že se má začít číslovat od začátku (prvního řádku) oznámíme \TeX u příkazem $\backslash\text{begcf}$, jinak se čísluje sekvenčně i přes několik volání $\backslash\text{cverb}$.
3. speciální prostředí pro přímé vkládání dokumentovaných zdrojových textů programů, podrobněji viz. kapitolu 2. 9. Dokumentování zdrojových kódů programů

2. 9. Dokumentování zdrojových kódů programů

Systém \LaTeX umožňuje psát dokumentaci ke zdrojovým textům programů (C++ a Tcl) přímo do těchto zdrojových kódů. To vede především k jednoduššímu udržování dokumentace při změnách v programu, ale také k lepší přehlednosti programu pro jeho autora i případného upravovatele.

2. 9. 1. Makra pro vložení C++ souboru

Jsou implementována dvě makra, která se liší pouze způsobem číslování řádek zdrojového kódu programu. Jde vlastně o „speciální“ `\input` rutiny, které připraví takové chování `TeXu`, že soubor, který je běžně kompilovatelný `C++` kompilátorem se v nezměněné podobě dá zařadit do dokumentace a kompilovat takto `TeXem`.

Zmíněná makra mají stejnou syntaxi jako makro `\input`. Jejich tvar je `\csrdocinput soubor` a `\csrctldocinput soubor`, kde soubor je jméno `C++` souboru včetně přípony. První z maker dokumentuje zdrojový kód a řádky céčkovského zdroje jsou číslovány od jedné postupně tak, jako by v souboru chyběla dokumentace. Tento způsob ale může někomu vadit (při hledání v souboru jsou čísla řádek v dokumentaci různá od skutečných), proto může použít druhou verzi makra, která čísluje řádky podle jejich skutečné pozice ve vstupním souboru. Pokud budou v `C++` souboru nějaké `TeXové` definice, které nebudou `\global`, budou tyto platit jen v rámci vloženého souboru, ve zbytku dokumentu se neprojeví.

2. 9. 2. Syntaxe dokumentace C++ souboru

Zdokumentování lze aplikovat na jakýkoliv `C++` zdrojový kód programu. Základním pravidlem je, že všechna dokumentace je umístěna v `C++` komentáři („dvojlomítka“ `//`), tudíž ji céčkovský kompilátor přeskóčí. `TeX` procesor je ale nastaven tak, že si těchto dvojlomítek „nevšímá“, takže do nich lze psát volně dokumentaci, která před začátkem výpisu zdrojového kódu programu zapne prostředí `\cverb` a před dalším začátkem dokumentace jej ukončí. V praxi může dokumentovaný `C++` soubor vypadat např. takto:

```
// % Tento komentář v komentáři se neobjeví nikde
// \section{Nějaká kapitola popisující program}
// Text představující popis programu, lze tady používat běžné TeX ové
// příkazy nebo jen psát text\dots
// \par
// \dots{} Odstavec se bohužel musí psát jak je výše předvedeno, dvojitě
// zařádkování s dvojitými lomítky nefunguje.
// \cverb
#include <stdio.h>

int main () { // tohle se vypíše jak je zapsáno (jsme ve verbatimu)
    printf ( "Ahoj světe!" ) ;
    return 0 ;
}
//\ecverb
// A zde může pokračovat dokumentace jako v předchozím případě
// až do konce souboru, jehož načítání TeX em je ukončeno standartně:
// \endinput
```

2. 9. 3. Makra pro vložení Tcl souboru

Analogicky jako vkládání céčkovského dokumentovaného souboru funguje i dokumentování `Tcl` souborů. Pouze místo `C++` komentáře se použije `Tcl` způsob „#“ a soubory se vkládají pomocí příkazů `\tclsrcdocinput` a `\tclsrctldocinput`.

2. 10. Poznámka

Makro `\poznamka` funguje tak, že jeho parametrem je poznámka, která se uvede ihned na místě vložení makra. Vypsána je kurzívou, odsazena 1 cm od levého okraje a od okolního textu oddělena linkami šíře 50 mm. Parametr může obsahovat i `\par`, tedy odstavcový zlom, který se projeví i v poznámce jako klasický odstavec, ovšem se zmíněnými úpravami vzhledu oproti okolnímu textu.

2. 11. Obsah

Dokumentační systém umožňuje použití několika typů obsahu podle tisknutého rozsahu a hloubky členění.

makro	Popis makra
<code>\celyobsah</code>	Kompletní obsah celé dokumentace
<code>\seznamknih</code>	Obsah jako seznam knih v kompletní dokumentaci
<code>\obsahknihy</code>	Obsah konkrétní knihy

Tabulka 1: Možnosti tisku obsahu

První dva uvedené způsoby se mohou používat, pouze pokud generujeme kompletní dokumentaci. První z nich udělá kompletní obsah ze všech knih a s nadpisy všech úrovní. `\seznamknih` funguje jako `\celyobsah` s tím rozdílem, že kapitoly jsou vynechány a jde tedy jen o výpis knih, z nichž dokumentace sestává.

Poslední je `\obsahknihy`, který vytvoří lokální obsah pouze aktuálně kompilované knihy. Použije se vždy aktuální kniha, v níž je obsah použit, přičemž jsou zahrnuty všechny kapitoly první až páté úrovně příslušné knihy. Pokud je zpracovávána pouze jedna kniha (ne celá dokumentace), jde vlastně o celý obsah té knihy.

2. 12. Odkazy na literaturu

Odkazovat se na literaturu lze podobně jako u ostatních objektů makrem `\litref{značka}`. V místě vložení bude číslo literatury označené značkou `značka` v hranatých závorkách. Číslování literatury se děje lokálně v rámci každé knihy.

Pro samotný seznam literatury je nejlépe napsat všechny jeho položky na jednom místě. Každá položka literatury se vkládá makrem `\literatura{název}{autor}{vydání}{značka}`. Přímo v místě vložení tohoto makra bude položka seznamu literatury a lze se na ni odkazovat výše zmíněným makrem `\litref`.

2. 13. Ostatní příkazy

Dokumentační systém obsahuje ještě několik dalších příkazů pro označení určitých typů textu. Tyto příkazy se při tisku dokumentu nemusí vůbec projevit, nebo jen způsobí např. tisk jiným fontem než je okolní text. Budou ale účelně využity třeba při tvorbě pdf nebo xml verze dokumentace (např. hypertextové odkazy) nebo ke generování seznamů příslušných objektů.

Příkaz	Popis příkazu
<code>\soubor{}</code>	parametr je jméno souboru
<code>\texmac{}</code>	parametr je \TeX ovské makro, není jako verbatim!
<code>\src{}</code>	parametr je příkaz v jazyce C či C++
<code>\tclsrc{}</code>	parametr je příkaz v jazyce Tcl
<code>\tclargs{}</code>	parametr jsou argumenty příkazu v jazyce Tcl
<code>\data{}</code>	parametr jsou data vstupně-výstupních souborů
<code>\httpodkaz{}</code>	1. parametr udává http adresu, 2. je text, který bude sloužit jako odkaz
<code>\ftpodkaz{}</code>	1. parametr udává ftp adresu, 2. je text, který bude sloužit jako odkaz
<code>\mailodkaz{}</code>	1. parametr je e-mailová adresa, 2. je text, který bude sloužit jako odkaz
<code>\newline</code>	„tvrdé“ odřádkování
<code>\newpage</code>	„tvrdý“ zlom stránky
<code>\\</code>	makro pro tisk zpětného lomítka
<code>\l</code>	tiskne levou složenou závorku
<code>\r</code>	tiskne pravou složenou závorku
<code>\softinput #1</code>	vloží do dokumentu soubor jen pokud existuje
<code>\autodoc</code>	vloží do dokumentace popis balíku \TeX (jako knihu)
<code>\autodocman</code>	jako <code>\autodoc</code> , ale vloží pouze uživatelskou část (použití příkazů)
<code>\autodocman</code>	jako <code>\autodoc</code> , ale vloží pouze část s popisem algoritmů maker \TeX
<code>\eqcr</code>	v prostředí <code>\eqalignno</code> místo <code>\cr</code> , čísluje řádek jako rovnici (možné odkazy)

Tabulka 2: Seznam „drobných“ příkazů dokumentačního systému

Dále musím zmínit změnu `\catcode` znaku „“, které je nastaveno na 12 (tzn. obyčejný znak) z důvodu častého výskytu ve zdrojových kódech programů psaných v jazyce C++. Pro matematická prostředí se význam tohoto znaku nemění.

3. Literatura

- [1] Petr Olšák: *TeXBook naruby*
Konvoj, Brno 2000, 80-85615-64-9
- [2] Petr Olšák: *Typografický systém TeX*
Konvoj, Brno 2000, ISBN 80-85615-91-6
- [3] Michael Doob, překlad a úprava Josef Daneš, Jiří Veselý: *Jemný úvod do TeXu*
CsTUG, Praha 1993, 3. upravené vydání českého překladu
- [4] Jiří Kosek: *XML pro každého*
Grada, Praha 2000, ISBN 80-7169-860-1
- [5] různé články: *Zpravodaj československého sdružení uživatelů TeXu*
CsTUG, Brno, ISSN 1211-6661
- [6] pdf \TeX examples aj.: *TeXLive 5d CD-ROM*
sdružení uživatelů \TeX u, VII. 2000

4. Popis algoritmů

V následující části popisují detailně makra, kterými je systém MDF naprogramován. V současné verzi jsou pomocí něj psané dokumenty zpracovatelné $\text{csplain}\text{T}_{\text{E}}\text{X}$ em pro dvi nebo ps výstup. Problémy nečiní ani kompilace $\text{pdfcsplain}\text{T}_{\text{E}}\text{X}$ em pro výstup ve formátu pdf , ale zatím nejsou využity všechny možnosti pro zpracování tímto programem.

4. 1. Test načtení souboru

Nejprve si program otestuje, zda byl již soubor maker načten. V případě dokumentace, kde je více knih (viz. kapitolu 2. 1. *Dělení na knihy*), se nebudou makra načítat několikrát, ale především inicializace některých proměnných (např. číslování stránek) několikrát během zpracování dokumentu by vedla k nesmyslným výstupům.

Jako test pro zjištění načtení souboru jsem si vybral řídicí sekvenci DefiniceNacteny , která je definována po prvním průchodu souborem maker a při její existenci v dalších průchodech je načítání souboru ihned ukončeno. Připojen je i výpis o těchto činnostech na terminál.

```

22 \expandafter\ifx\csname DefiniceNacteny\endcsname \relax
23 \expandafter\def \csname DefiniceNacteny\endcsname {1}
24 \message{Načítám soubor mdef.tex...}
25 \else
26 \message{- přeskočen}
27 \expandafter\endinput \fi

```

4. 2. Inicializace proměnných (knihy)

Číslo aktuálně zpracovávané knihy je uchováváno v proměnné $\backslash\text{bookno}$. Protože se ale používá římského číslování knih, je definováno i makro $\backslash\text{thebook}$, které je implicitně prázdné — při zpracování jedné knihy je toto číslo vynecháváno. V tom případě také makro $\backslash\text{konecknihy}$ bude ukončovat celou dokumentaci a je tedy implicitně definováno jako $\backslash\text{end}$.

```

28 \def\thebook{} % číslo knihy římsky, používá se bez mezer kolem
29 \def\bookname{} % jméno aktuálně zpracovávané knihy
30 \def\konecknihy{\end} % implicitní konec knihy
31 \newcount \bookno % aktuální číslo knihy

```

4. 3. Inicializace proměnných (kapitoly)

Zde stojí za zmínku především makro $\backslash\text{fpkapno}$, v němž je uloženo číslo aktuální kapitoly včetně čísel všech úrovní podkapitol. Další makra slouží k uchování názvů kapitol pro potřeby především záhlaví stránek. Konkrétní význam maker je zřejmý z komentářů:

```

32 \def\fpkapno{} % místo čísla aktuální kapitoly bude řetězec
33 \newcount \kapno % aktuální kapitola 1. úrovně
34 \newcount \subkapno % aktuální kapitola 2. úrovně
35 \newcount \subsubkapno % aktuální kapitola 3. úrovně
36 \newcount \parkapno % aktuální kapitola 4. úrovně
37 \newcount \pparkapno % aktuální kapitola 5. úrovně
38 % následující definice se používají pro hlavičku stránky:
39 \def\fpkapname{} % název první kapitoly aktuální stránky
40 \def\lpagelkap{} % název poslední kapitoly na poslední stránce
41 \def\lpagefkap{} % název první kapitoly na poslední stránce
42 \newcount \lastpagekap % poslední stránka začátku kapitoly
43 % a tyto definice používá pdf\TeX:
44 \expandafter\ifx\csname pdfoutput\endcsname \relax
45 \else
46 \pdflinkmargin=0pt
47 \pdfmovechars=2
48 \pdfpkresolution=300
49 \pdfuniqueresname=0
50 \pdfcompresslevel=4
51 \fi
52 % následující definice se používají pro hlavičku stránky:
53 \def\fpkapname{} % název první kapitoly aktuální stránky
54 \def\lpagelkap{} % název poslední kapitoly na poslední stránce
55 \def\lpagefkap{} % název první kapitoly na poslední stránce
56 \newcount \lastpagekap % poslední stránka začátku kapitoly

```

57
58

4. 4. Definice makra pro knihu

Nejdříve jsou vynulována všechna počítadla pro kapitoly, k tomu můžeme doplnit nulování očíslování rovnic, tabulek... Dále jsou definovány údaje pro titulní stránku knihy (`\nazev`, `\podnazev` a `\autor`).

Další část definice je větvená podle toho, zda je zpracovávána jen jedna kniha nebo kompletní dokument:

- při samostatné knize se definuje makro `\thebook` pro číslování knih jako prázdné, poté je vynulováno počítadlo stránek a vygenerována titulní strana.
- Je-li dokument složen z více knih, je po přechodu na novou stránku zvýšeno číslo knihy o 1, v makru `\bookname` definujeme jméno knihy a makrem `\thebook` definujeme římské očíslování knihy. Nakonec je proveden zápis do souboru pro generování obsahu a makrem `\titulknihy` je vysázena titulní stránka knihy.

```

59 \def\kniha #1#2#3{
60   \litnum = 0   \kapno = 0
61   \cisrov = 0   \cistab = 0   \cisobr = 0
62   \subkapno = 0 \subsubkapno = 0 \parkapno = 0 \pparkapno = 0
63   \def\nazev{\thebook{ } #1}
64   \def\podnazev{#2}
65   \def\autor{#3}
66   \expandafter\ifx\cename maindoccs\endcsname \relax
67     \def\thebook{ } \pageno = 0 \titulknihy
68   \else
69     \newpage
70     \global\advance\bookno by1 \def\bookname{#1}
71     \def\thebook{\uppercase\expandafter{\romannumeral\bookno}. }
72     \titulknihy
73     \maketoclinebook{\thebook #1}
74     \expandafter\ifx\cename pdfoutput\endcsname \relax \else {
75       \pdfdest name{book\thebook} fith
76       \pdfoutline goto name{book\thebook} count 0 {\bookname}} \fi
77   \fi}

```

4. 5. Kapitoly

Podobně jako kniha jsou definována i makra pro definici kapitol různých úrovní. Nejdříve se tedy nulují počítadla všech hierarchicky nižších objektů a počítadlo aktuálního objektu se zvýší o jedničku. Dále je definováno makro `\fpkapno`, které se používá k označování čísla aktuální kapitoly, ve `\vbox` je vypsán název kapitoly s jejím číslem (`\vbox` je použit z důvodu zalomení případného dlouhého nadpisu). Záporná penalta před názvem kapitoly slouží k vynucení zalomení stránky v případech, kdy by byl její nadpis „příliš nízký“ na stránce. Nakonec je použit příkaz `\maketocline x` příslušné úrovně x pro zápis nadpisu do souboru obsahu.

U hlavní kapitoly `\section` je navíc vyhodnocení podmínky pro přiřazení názvu nadpisu do hlavičky stránky.

```

78 %\newcount\secc
79 %%%
80 %\expandafter\def\cename secc1. \endcsname{3}
81 \newcount\secoutlinea
82 \newcount\secoutlineb
83 \newcount\secoutlinec
84 \newcount\secoutlined
85 \gdef\seccounter#1#2{\expandafter\gdef\cename #1\endcsname{#2}}
86 %%%
87 \def\section #1{\subkapno = 0 \subsubkapno = 0 \parkapno = 0 \pparkapno = 0
88   \def\fpkapno{\thebook\the\kapno . }
89   % \newpage % neni-li newpage, je \vskip 24pt:
90   \def\fpkapname{#1} \vskip 24 pt
91   \global\advance\kapno by1
92   \penalty -2000
93 %%%
94 \expandafter\ifx\cename pdfoutput\endcsname \relax \else
95   \expandafter\ifx\cename secc\fpkapno\endcsname \relax % neex.-li pocitadlo, je 0
96   \expandafter\def\cename secc\fpkapno\endcsname{0}
97   \fi}

```

```

98     \secoutlinea=0
99     \pdfdest name {section\fpkapno} fith
100    \pdfoutline goto name {section\fpkapno} count
101    \expandafter\csname secc\fpkapno\endcsname {\the\kapno. \fpkapname}
102    \fi
103    %%%
104    \leftline{\bigggg\ vbox{\fpkapno #1}}
105    \ifnum \lastpagekap = \pageno
106    \else \global\lastpagekap=\pageno
107    \edef\lpagefkp{\the\kapno . #1} \fi
108    \edef\lpagek{\fpkapno #1}
109    \vskip Opt %-\parskip
110    \maketoclinea{\fpkapno #1 }}

```

Další makra pro nižší úrovně kapitol nejsou již popisována, fungují analogicky jako `\section`. Jsou to:

- `\subsection`
- `\subsubsection`
- `\partsection`
- `\ppartsection`

```

111 \def\subsection #1{\global\advance\subkapno by1
112 \subsubkapno = 0 \parkapno = 0 \pparkapno = 0
113 \def\fpkapno{\thebook\the\kapno . \the\subkapno . }
114 \def\fpkapname{#1} \vskip 20 pt \penalty -1500
115 \expandafter\ifx\csname pdfoutput\endcsname \relax \else
116 \expandafter\ifx\csname secc\fpkapno\endcsname \relax % neex.-li pocitadlo, je 0
117 \expandafter\def\csname secc\fpkapno\endcsname{0}
118 \fi
119 \secoutlineb=0
120 \advance\secoutlinea by1
121 \maketoclinecc{secc\the\kapno . }\the\secoutlinea}
122 \pdfdest name{section\fpkapno} fith
123 \pdfoutline goto name{section\fpkapno} count
124 \expandafter\csname secc\fpkapno\endcsname {\the\kapno . \the\subkapno . \fpkapname}
125 \fi
126 \leftline{\bigg\ vbox{\fpkapno #1}}
127 \vskip Opt %-\parskip
128 \maketoclineb{\fpkapno #1} }
129
130 \def\subsubsection #1{\global\advance\subsubkapno by1
131 \parkapno = 0 \pparkapno = 0
132 \def\fpkapno{\thebook\the\kapno . \the\subkapno .
133 \the\subsubkapno . }
134 \def\fpkapname{#1} \vskip 18 pt \penalty -1000
135 \expandafter\ifx\csname pdfoutput\endcsname \relax \else
136 \expandafter\ifx\csname secc\fpkapno\endcsname \relax % neex.-li pocitadlo, je 0
137 \expandafter\def\csname secc\fpkapno\endcsname{0}
138 \fi
139 \secoutlinec=0
140 \advance\secoutlineb by1
141 \maketoclineccc{secc\the\kapno . \the\subkapno . }\the\secoutlineb}
142 \pdfdest name{section\fpkapno} fith
143 \pdfoutline goto name{section\fpkapno} count
144 \expandafter\csname secc\fpkapno\endcsname
145 {\the\kapno . \the\subkapno . \the\subsubkapno . \fpkapname}
146 \fi
147 \leftline{\bigg\ vbox{\fpkapno #1}}
148 \vskip Opt %-\parskip
149 \maketoclinec{\fpkapno #1} }
150
151 \def\partsection #1{\global\advance\parkapno by1
152 \pparkapno = 0
153 \def\fpkapno{\thebook\the\kapno . \the\subkapno .
154 \the\subsubkapno . \the\parkapno . }
155 \def\fpkapname{#1} \vskip 14 pt \penalty -1000
156 \expandafter\ifx\csname pdfoutput\endcsname \relax \else
157 \expandafter\ifx\csname secc\fpkapno\endcsname \relax % neex.-li pocitadlo, je 0
158 \expandafter\def\csname secc\fpkapno\endcsname{0}
159 \fi
160 \secoutlined=0
161 \advance\secoutlinec by1
162 \maketoclineccc{secc\the\kapno . \the\subkapno . \the\subsubkapno . }\the\secoutlinec}
163 \pdfdest name{section\fpkapno} fith
164 \pdfoutline goto name{section\fpkapno} count
165 \expandafter\csname secc\fpkapno\endcsname
166 {\the\kapno . \the\subkapno . \the\subsubkapno . \the\parkapno . \fpkapname}
167 \fi
168 \leftline{\bigbf\ vbox{\fpkapno #1}}
169 \vskip Opt %-\parskip

```

```

170 \maketoclined{\fkapno #1} }
171
172 \def\ppartsection #1{\global\advance\pparkapno by1
173 \def\fkapno{\thebook\the\kapno . \the\subkapno .
174 \the\subsubkapno . \the\parkapno . \the\pparkapno . }
175 \def\fkapname{#1} \vskip 12 pt \penalty -1000
176 \expandafter\ifx\curname pdfoutput\endcsname \relax \else
177 \advance\secoutlined by1
178 \maketoclinecc{secc\the\kapno . \the\subkapno . \the\subsubkapno .
179 \the\parkapno . }\the\secoutlined}
180 \pdfdest name{section\fkapno} fith
181 \pdfoutline goto name{section\fkapno} count 0
182 {\the\kapno . \the\subkapno . \the\subsubkapno .
183 \the\parkapno . \the\pparkapno . \fkapname}
184 \fi
185 \leftline{\big\vbbox{\fkapno #1}}
186 \vskip 0pt %-\parskip
187 \maketoclinee{\fkapno #1} }

```

Nakonec ještě definujeme makro `\nosection`, jehož výstup vypadá jako u makra `\section` s tím rozdílem, že takováto kapitola není očíslována. Toto makro je vhodné např. pro úvod kapitoly či dokumentace. Z důvodu nápisu v hlavičce stránky přepisuje význam maker `\thebook` a `\bookname`, protože se jeho použití předpokládá „mimo knihu“.

```

188 \def\nosection #1{\def\fkapno{
189 % \newpage % není-li newpage, je \vskip 24pt:
190 \vskip 24 pt \penalty -2000
191 \expandafter\ifx\curname pdfoutput\endcsname \relax \else {\pdfdest name{section\fkapno}
fith
192 \pdfoutline goto name{section\fkapno} count 0 {\fkapname}} \fi
193 \leftline{\bigggg\vbbox{\fkapno #1}}
194 \ifnum \lastpagekap = \pageno %\message{NIC.}%
195 \else \global\lastpagekap=\pageno %\message{NECO.}%
196 \edef\lpagefkp{#1}\fi
197 \edef\lpagek{#1}
198 \vskip 0pt %-\parskip
199 \def\thebook{} \def\bookname{#1}
200 \maketoclinea{\fkapno \noexpand#1 }

```

4. 6. Zpracování referencí na knihu a kapitoly

Následně jsou definována makra pro odkaz a značku na kapitolu a dále pomocné makro `\secREF`, které slouží k vložení odkazů ze souboru `\jobname.ref` z předchozího běhu \TeX procesoru (tj. zpětné reference).

Makro `\seclabel` zapisuje aktuální číslo a název kapitoly do souboru referencí pomocí makra `\zapisref`, kde jsou pro výstupní rutinu „anulována“ makra používaná v názvech kapitol, aby bylo možno je opět načíst pro reference. Zároveň je brána v úvahu hodnota počítadla `\globalrefs`, která udává, zda se budou reference na kapitoly provádět v rámci celého dokumentu (tj. i mezi knihami) nebo v rámci každé knihy zvlášť.

```

201 \newtoks\pisref
202 \newcount\globalrefs
203 \def\zapisref#1{\let\_= \relax \let~ = \relax \let\dots = \relax
204 \let\TeX = \relax \let\MDef = \relax
205 \pisref={#1}\edef\act{\write\REF{\the\pisref}}\immediate\act}}
206 \def\seclabel #1{%
207 \if 0\the\globalrefs
208 \zapisref{\string\secREF{\the\bookno#1}{\fkapno \fkapname}}
209 % \zapisref{\string % \secREF{\the\bookno#1}{\fkapno \fkapname}
210 \expandafter\ifx\curname pdfoutput\endcsname \relax \else
211 \pdfdest name{\the\bookno#1} fitr width 210mm height 1em depth 0mm
212 \fi
213 \else \zapisref{\string % \secREF{#1}{\fkapno \fkapname}
214 \expandafter\ifx\curname pdfoutput\endcsname \relax \secREF{#1}{\fkapno \fkapname}
\else
215 \pdfstartlink attr{Border [0 0 0]}goto name{#1}
216 \Blue\secREF{#1}{\fkapno \fkapname}\Black \pdfendlink
217 \fi
218 }
219 \fi}

```

`\secref` dodá na výstup řetězec definovaný příkazem `\seclabel` pokud existuje, jinak vypíše „Warning...“. Opět zde bereme v úvahu hodnotu `\globalrefs`.

```

220 \def\secref #1{%
221 \if 0\the\globalrefs
222 \expandafter\ifx\curname sec:\the\bookno#1\endcsname \relax

```

```

223     \message{Warning: Undefined sec number [#1] (local)}??%
224   \else
225 %     \csname sec:\the\bookno#1\endcsname
226     \expandafter\ifx\csname pdfoutput\endcsname \relax
227     \csname sec:\the\bookno#1\endcsname
228     \else
229       \hbox{\pdfstartlink attr{/Border [0 0 0]}
230         goto name{\the\bookno#1} \Blue
231         \csname sec:\the\bookno#1\endcsname
232         \Black\pdfendlink}\fi\fi
233   \else%
234     \expandafter\ifx\csname sec:#1\endcsname \relax
235     \message{Warning: Undefined sec number [#1]}??%
236     \else\csname sec:#1\endcsname
237     \expandafter\ifx\csname pdfoutput\endcsname \relax \else
238     \pdfdest name{#1} fith
239   \fi
240 %\fi
241 \fi
242 \fi}

```

Makro `\secREF` je pomocným makrem, které se používá při načítání a zápisu souboru referencí:

- při zápisu je toto makro zapsáno, přičemž jeho parametry jsou značka odkazu a text ukazující na dané místo (tj. zde číslo a název kapitoly)
- při načítání souboru referencí tato makra definují řídicí sekvence, které definují příslušné odkazy nebo v případě, že řídicí sekvence již existuje, hlásí „Error...“.

```

243 \def\secREF #1#2{
244   \expandafter\ifx\csname sec:#1\endcsname \relax
245   \expandafter\def \csname sec:#1\endcsname {#2}
246   \else \errmessage{Error: Double sec mark [#1]}\fi}

```

Analogickým způsobem jsou definována i makra pro tvorbu referencí na všechny ostatní objekty, takže dále již nejsou podrobně popisována. Zde dodefinujeme makra pro odkazy na knihu:

```

247 \def\knihalabel #1{
248   \immediate\write\REF{\string\knihaREF{#1}{\thebook \bookname}}
249 \def\kniharef #1{\expandafter\ifx\csname kniha:#1\endcsname \relax
250   \message{Warning: Undefined book number [#1]}??%
251   \else\csname kniha:#1\endcsname\fi}
252 \def\knihaREF #1#2{\expandafter\ifx\csname kniha:#1\endcsname \relax
253   \expandafter\def \csname kniha:#1\endcsname {#2}
254   \else \errmessage{Error: Double book mark [#1]}\fi}

```

4. 7. Výčty a odrážky

Pro toto prostředí nejdříve definujeme počítadla pro všechny tři úrovně. Prostředí podle hodnoty počítadel rozeznává aktuální stav, v jakém se \TeX procesor vzhledem k těmto prostředím vyskytuje.

Je-li hodnota počítadel nulová, nejsme v žádném prostředí výčtů či odrážek. Je-li tato hodnota záporná, jsme v prostředí odrážek (bývá nastavena na -1). Při kladných hodnotách počítadel se vyskytujeme v prostředí výčtů a příslušné hodnoty počítadel odpovídají očíslování aktuální položky výčtu.

```

255 \newcount\numitemnum \newcount\numiitemnum \newcount\numiiitemnum
256 \numitemnum = 0 \numiitemnum = 0 \numiiitemnum = 0
257 \let\puvitem=\item % v \puvitem ponecháme původní význam \item

```

Teď definujeme makra pro začátek obou prostředí. Tato makra dělají pouze to, že nastaví počítadla všech tří úrovní do stavu, který byl pro obě prostředí popsán v předchozím odstavci.

```

258 \def\begitem{\numitemnum = -1 \numiitemnum = -1 \numiiitemnum = -1
259   \par \bgroup \parindent Opt \parskip Opt}
260 \def\begnum{\numitemnum = 1 \numiitemnum = 1 \numiiitemnum = 1
261   \par \bgroup \parindent Opt \parskip Opt}

```

Na konci prostředí jsou čítače uvedeny do nulového stavu a je ukončena skupina (kterou zahajuje poslední z položek prostředí). Ukončení pro obě prostředí jsou definována stejně (proto je zde `\def\endnum{\enditem}`).

```

262 \def\enditem{\vskip Opt \egroup
263   \numitemnum = 0 \numiitemnum = 0 \numiiitemnum = 0
264   \vskip Opt}
265 \def\endnum{\enditem}

```

Následující makra `\item`, `\iitem` a `\iiitem` definují postupně všechny tři úrovně položek seznamů s číslováním nebo odrážkami.

Makro nejdříve zkontroluje, zda byl zahájen některý ze seznamů. V případě použití mimo seznam vypíše \TeX chybu. Poté je makro rozděleno na dvě části podle toho, jestli se vyskytujeme v číslovaném seznamu nebo seznamu s odrážkami. S každou položkou je zrušena skupina, která je opět zahájena, jsou v ní definovány rozměry aktuální položky a je uzavřena další položkou nebo na konci ukončovacím makrem prostředí.

V seznamu s odrážkami je pouze nastaveno správné odsazení textu položky a vytištěna její značka.

V číslovaném seznamu jsou navíc nulovány všechny nižší úrovně číslování. Po vysázení čísla položky a jejího textu je číslo položky zvýšeno o jedničku.

```

266 \def\item{
267   \ifnum\numitemnum = 0 \errmessage{Chybné umístění položky item.}
268   \else
269     \par \hangafter=0 \hangindent=10mm
270     \ifnum\numitemnum = -1
271       \textindent{\hbox to10mm{\hfil$\bullet$\hfil}}
272     \else
273       \textindent{\hbox to10mm{\hfil\the\numitemnum . }}
274     \advance\numitemnum by1 \numiitemnum = 97 \numiitemnum = 1
275   \fi
276 \fi
277 }

```

Ostatní úrovně položek „odrážkovaných“ nebo číslovaných seznamů jsou vytvořeny analogicky k první úrovni, liší se přičítáním čítače pro tu kterou úroveň v seznamu a rozměry a odsazení položek seznamu dané úrovně.

```

278 \def\iitem{
279   \ifnum\numiitemnum = 0 \errmessage{Chybné umístění položky iitem.}
280   \else
281     \par \hangafter=0 \hangindent=20mm
282     \ifnum\numitemnum = -1
283       \textindent{\hbox to20mm{\hfil$\circ$\quad}}
284     \else
285       \textindent{\hbox to20mm{\hfil\char\numiitemnum . }}
286     \advance\numiitemnum by1 \numiitemnum = 1
287   \fi
288 \fi
289 }
290 \def\iiitem{
291   \ifnum\numiitemnum = 0 \errmessage{Chybné umístění položky iiitem.}
292   \else
293     \par \hangafter=0 \hangindent=30mm
294     \ifnum\numitemnum = -1
295       \textindent{\hbox to30mm{\hfil$\cdot$\quad}}
296     \else
297       \textindent{\hbox to30mm{\hfil\the\numiitemnum . }}
298     \advance\numiitemnum by1
299   \fi
300 \fi
301 }

```

4. 8. Deklarace souboru referencí a stránková reference

Zde je deklarován ukazatel na soubor $\backslash\text{REF}$, do nějž budou zapisovány všechny reference pro načítání v dalším běhu \TeX procesoru. Dále jsou na tomto místě deklarována makra pro zápis a načítání stránkové reference, tak jako u odkazů na kapitoly se bere v úvahu hodnota proměnné $\backslash\text{globalrefs}$ pro tvorbu odkazů i mezi knihami.

```

302 \newwrite\REF
303 \def\strlabel #1{%
304   \if 0\the\globalrefs
305     \immediate\write\REF{\string\strREF{\the\bookno#1}{\the\pageno}}
306   \else\immediate\write\REF{\string\strREF{#1}{\the\pageno}} \fi
307 \def\strref #1{%
308   \if 0\the\globalrefs
309     \expandafter\ifx\cname str:\the\bookno#1\endcsname \relax
310     \message{Warning: Undefined page number [#1]}??%
311   \else\cname str:\the\bookno#1\endcsname\fi
312 \else
313   \expandafter\ifx\cname str:#1\endcsname \relax
314   \message{Warning: Undefined page number [#1]}??%
315   \else\cname str:#1\endcsname\fi
316 \fi}
317 \def\strREF #1#2{\expandafter\ifx\cname str:#1\endcsname \relax

```

```

318     \expandafter\def \csname str:#1\endcsname {#2}
319     \else \errmessage{Error: Double page mark [#1]}\fi}

```

4. 9. Obsah

Obsah je zde řešen „způsobem dvojího načítání“, jak jsem si jej pro sebe nazval. Princip je takový, že všechny položky obsahu se zapisují do souboru `\jobname.top` při kterémkoliv průchodu \TeX u. Ale pokud tento soubor již existuje z minulého zpracování, je nejdříve přepokopírován do `\jobname.toc`, který je od té chvíle pouze pro čtení a je načítán v případě volání některého z maker pro obsah.

Nejdříve si musíme založit ukazatele na soubory pro čtení a zápis, dále proměnnou pro číslo knihy, pro kterou je generován obsah.

```

320 \newread\TOCTMP
321 \newwrite\TOC
322 \newcount\kaptoc

```

Poté provedeme zápis souboru `\jobname.top` do souboru `\jobname.toc`. To je nejjednodušší tak, že nadefinujeme všechna makra, která se mohou vyskytnout v souboru `\jobname.top` tak, aby sama sebe zapsala do souboru `\jobname.toc`. Parametry zde použitých maker jsou:

1. číslo knihy
2. název kapitoly včetně očíslování
3. strana (resp. nějaké označení místa), na které bude položka ukazovat

```

323 \newtoks\pisretezec
324 \def\zapis#1{\let\_=\relax \let~=\relax \let\dots=\relax
325             \let\TeX=\relax \let\Mdef=\relax
326 \pisretezec=#1}\edef\act{\write\TOC{\the\pisretezec}}\immediate\act}}
327
328 \def\toclinebook#1#2#3{\zapis{\string\toclinebook{#1}{#2}{#3}}}
329 \def\toclinea#1#2#3{\zapis{\string\toclinea{#1}{#2}{#3}}}
330 \def\toclineb#1#2#3{\zapis{\string\toclineb{#1}{#2}{#3}}}
331 \def\toclinec#1#2#3{\zapis{\string\toclinec{#1}{#2}{#3}}}
332 \def\toclined#1#2#3{\zapis{\string\toclined{#1}{#2}{#3}}}
333 \def\toclinee#1#2#3{\zapis{\string\toclinee{#1}{#2}{#3}}}
334 \immediate\openout\TOC=\jobname.toc
335 \immediate\openin\TOCTMP=\jobname.top
336 % test (ne)existence souboru \TOCTMP :
337 \ifeof\TOCTMP message{Soubor \jobname.top neexistuje...}
338 \else\immediate\closein\TOCTMP\input \jobname.top\fi
339 \immediate\closeout\TOC
340 \immediate\openout\TOC=\jobname.top

```

Poté definujeme jednotlivé řádky pro různé úrovně kapitol, jak budou vypadat v obsahu (tj. implicitní při načtení souboru `\jobname.toc`). Vždy je vložena podmínka, zda se generuje kompletní obsah celé dokumentace složené z více knihy nebo obsah pouze jedné knihy ať už umístěné samostatně nebo v rámci většího celku.

```

341 \def\toclinebook#1#2#3{%
342     \ifnum 0 = \the\kaptoc
343         \vskip6pt \penalty -1500
344         \line{\bigbf #2\quad\dotfill\hbox to12mm{\hfil #3}} \fi}%
345 \def\toclinea#1#2#3{%
346     \ifnum 0 = \the\kaptoc
347         \vskip6pt \penalty -1000
348         \line{\big #2\quad\dotfill\hbox to12mm{\hfil #3}}%
349     \else\ifnum #1 = \the\kaptoc
350         \vskip6pt \penalty -1000
351         \line{\big #2\quad\dotfill\hbox to12mm{\hfil #3}}%
352     \fi\fi}
353 \def\toclineb#1#2#3{%
354     \ifnum 0 = \the\kaptoc \line{\hskip24pt{\rm #2 \hfill #3}}%
355     \else\ifnum #1 = \the\kaptoc \line{\hskip24pt{\rm #2 \hfill} #3}%
356     \fi\fi}
357 \def\toclinec#1#2#3{%
358     \ifnum 0 = \the\kaptoc \line{\hskip36pt{\rm #2 \hfill #3}}%
359     \else\ifnum #1 = \the\kaptoc \line{\hskip36pt{\rm #2 \hfill} #3}%
360     \fi\fi}
361 \def\toclined#1#2#3{%
362     \ifnum 0 = \the\kaptoc \line{\hskip48pt{\rm #2 \hfill #3}}%
363     \else\ifnum #1 = \the\kaptoc \line{\hskip48pt{\rm #2 \hfill} #3}%
364     \fi\fi}
365 \def\toclinee#1#2#3{%
366     \ifnum 0 = \the\kaptoc \line{\hskip60pt{\rm #2 \hfill #3}}%
367     \else\ifnum #1 = \the\kaptoc \line{\hskip60pt{\rm #2 \hfill} #3}%
368     \fi\fi}

```

Následující makro `\xobsah` vypíše obsah kapitoly číslo `\kaptoc`, pokud je `\kaptoc` větší než 0. Je-li `\kaptoc` rovno nule, je vygenerován kompletní obsah všech knih. K oběma je použito formátování uvedené výše.

```

369 \def\xobsah{
370   \ifnum \the\kaptoc = 0
371     \newpage\bgroup\headline{\hfil}\leftline{\bigg Obsah}}%
372   % \maketoclinebook{Obsah}
373   \else \newpage\bgroup\leftline{\bigg Obsah \thebook\bookname}} \fi%
374   % \maketoclinea{Obsah \thebook\bookname}\fi
375   \immediate\openin\TOCTMP=\jobname.toc
376   \ifeof\TOCTMP\message{Soubor \jobname.toc neexistuje...}%
377   \else\immediate\closein\TOCTMP\input \jobname.toc \vskip Opt
378   \fil\eject\egroup\fi
379 }

```

Makra `\obsahknihy` a `\celyobsah` generují částečný nebo kompletní obsah výše zmíněným způsobem. Makro `\seznamknihi` vygeneruje stručnější kompletní obsah, v němž budou uvedeny pouze knihy. Makra vypisující do obsahu další úrovně kapitol jsou definována jako prázdná.

```

380 \def\obsahknihy{\kaptoc = \the\bookno \xobsah}
381 \def\celyobsah{\kaptoc = 0 \xobsah}
382 \def\seznamknihi{\kaptoc = 0
383   \def\toclinebook##1##2##3{%
384     \vskip6pt\line{\bigbf ##2\quad\hrulefill\hbox to12mm{\hfil ##3}}} }
385   \def\toclinea##1##2##3{ % od této úrovně se položky v seznamu knih
386     \def\toclineb##1##2##3{ % neuvádějí...
387       \def\toclinec##1##2##3{
388         \def\toclined##1##2##3{
389           \def\toclinee##1##2##3{
390             \xobsah}}

```

Ohledně obsahu zbývá dodefinovat ještě makra pro zápis příslušné obsahové reference do souboru `\jobname.top`. Makrům pro příslušné úrovně zápisu se zadává pouze jméno kapitoly či knihy. Číslo knihy a stránka, na niž budeme odkazovat se doplňuje automaticky. Následující makra jsou pak volána makry pro začátek knihy nebo kapitol:

```

391 \def\maketoclinebook#1{\zapis{\string\toclinebook{\the\bookno}{#1}{\the\pageno}}}
392 %\def\maketoclinea#1{\zapis{\string\toclinea{\the\bookno}{#1}{
393 % \expandafter\ifx\csname pdfoutput\endcsname \relax \the\pageno \else
394 % \pdfstartlink attr{/Border [0 0 0]}goto name{section\fpkapno}
395 % \Blue\the\pageno\Black \pdfendlink \fi}}}
396 \def\maketoclinea#1{\zapis{\string\toclinea{\the\bookno}{
397 \expandafter\ifx\csname pdfoutput\endcsname \relax #1 \else
398 \pdfstartlink attr{/Border [0 0 0]}goto name{section\fpkapno}
399 \Blue #1 \Black \pdfendlink \fi} {
400 \expandafter\ifx\csname pdfoutput\endcsname \relax \the\pageno \else
401 \pdfstartlink attr{/Border [0 0 0]}goto name{section\fpkapno}
402 \Blue\the\pageno\Black \pdfendlink \fi}}}
403 \def\maketoclineb#1{\zapis{\string\toclineb{\the\bookno}{#1}{
404 \expandafter\ifx\csname pdfoutput\endcsname \relax \the\pageno \else
405 \pdfstartlink attr{/Border [0 0 0]}goto name{section\fpkapno}
406 \Blue\the\pageno\Black \pdfendlink \fi}}}
407 \def\maketoclinec#1{\zapis{\string\toclinec{\the\bookno}{#1}{
408 \expandafter\ifx\csname pdfoutput\endcsname \relax \the\pageno \else
409 \pdfstartlink attr{/Border [0 0 0]}goto name{section\fpkapno}
410 \Blue\the\pageno\Black \pdfendlink \fi}}}
411 \def\maketoclined#1{\zapis{\string\toclined{\the\bookno}{#1}{
412 \expandafter\ifx\csname pdfoutput\endcsname \relax \the\pageno \else
413 \pdfstartlink attr{/Border [0 0 0]}goto name{section\fpkapno}
414 \Blue\the\pageno\Black \pdfendlink \fi}}}
415 \def\maketoclinee#1{\zapis{\string\toclinee{\the\bookno}{#1}{
416 \expandafter\ifx\csname pdfoutput\endcsname \relax \the\pageno \else
417 \pdfstartlink attr{/Border [0 0 0]}goto name{section\fpkapno}
418 \Blue\the\pageno\Black \pdfendlink \fi}}}
419 \def\maketoclinecc#1#2{\zapis{\string\seccounter{#1}{#2}}}
420 %\toclinee{\the\bookno}{#1}{
421 % \expandafter\ifx\csname pdfoutput\endcsname \relax \the\pageno \else
422 % \pdfstartlink attr{/Border [0 0 0]}goto name{section\fpkapno}
423 % \Blue\the\pageno\Black \pdfendlink \fi}}}

```

4. 10. Seznam literatury

Pomocí makra `\literatura` je vysázena položka literatury přímo v místě použití tohoto makra. K tomu účelu výhodně využijeme plainové makro `\item` (v prostředí výčtů bylo předefinováno, původní význam zůstal v `\puvitem`). Zároveň volá zde definované makro `\litlabel`, které vytvoří odkaz na danou položku literatury. Reference pak funguje analogicky jako ostatní reference, výstupem je číslo literatury v hranatých závorkách. Číslování literatury je uloženo v registru `\litnum`.

```

424 \newcount\litnum
425 \def\literatura#1#2#3#4{
426   \global\advance\litnum by1
427   {\parindent 9mm \parskip 14pt
428     \par
429     \hangindent 24mm \hangafter 1
430     \puvitem {\hbox to 9mm{\hfil{\bf [the\litnum]}}}
431     {\rm#2}: {\it#1} }\newline
432   {\rm#3}\par}
433   \litlabel{#4}}
434
435 \def\litlabel #1{%
436   \if 0\the\globalrefs
437     \immediate\write\REF{\string\litREF{the\bookno#1}{the\litnum}}
438   \else
439     \immediate\write\REF{\string\litREF{#1}{the\litnum}}
440   \fi}
441 \def\litref #1{%
442   \if 0\the\globalrefs
443     \expandafter\ifx\csname lit:\the\bookno#1\endcsname \relax
444     \message{Warning: Undefined lit number [#1]}??%
445     \else\csname lit:\the\bookno#1\endcsname\fi
446   \else
447     \expandafter\ifx\csname lit:#1\endcsname \relax
448     \message{Warning: Undefined lit number [#1]}??%
449     \else\csname lit:#1\endcsname\fi
450   \fi}
451 \def\litREF #1#2{\expandafter\ifx\csname lit:#1\endcsname \relax
452   \expandafter\def \csname lit:#1\endcsname {#2}
453   \else \errmessage{Error: Double lit mark [#1]}\fi}

```

4. 11. Rovnice

Protože samotný plain \TeX má vcelku dostačující možnosti pro psaní rovnic, jsou dodefinována pouze makra pro možnosti číslování rovnic a možnosti odkazovat se na ně.

Makra `[a]` byla inspirována \LaTeX em a slouží k sazbě číslované rovnice, na niž se lze odkazovat makrem `\eqref`. Sekvence `\eqcr` se používá v prostředí `\eqalignno` místo `\cr` z důvodu číslování rovnic — `\eqcr` přidá na pravé straně číslo rovnice a zvětší jej o jedničku.

```

454 \newcount \cisrov
455 \cisrov = 0
456 \def\[{ \global\advance\cisrov by1 $$ }
457 \def\] { \eqno (\thebook\the\cisrov)\quad $$ }
458 \def\eqcr { \global\advance\cisrov by1 (\thebook\the\cisrov)\quad \cr }
459 \def\eqlabel #1{
460   \immediate\write\REF{\string\eqREF{the\bookno#1}{thebook\the\cisrov}}
461 \def\eqref #1{\expandafter\ifx\csname eq:\the\bookno#1\endcsname \relax
462   \message{Warning: Undefined eq. number [#1]}??%
463   \else(\csname eq:\the\bookno#1\endcsname)\fi}
464 \def\eqREF #1#2{\expandafter\ifx\csname eq:#1\endcsname \relax
465   \expandafter\def \csname eq:#1\endcsname {#2}
466   \else \errmessage{Error: Double eq. mark [#1]}\fi}

```

4. 12. Různé definice

Zde jsou definovány drobnější definice pro různý účel, většina pouze formátuje text určitým způsobem (např. jiný font), v budoucnu může být přidáno interaktivní odkazování např. na soubor nebo internetový odkaz, které z takto vytvořeného dokumentu udělá třeba „klikací“ pdf dokument. Jiná možnost je použití takových maker třeba pro xml dokumentaci, protože zde není pouze definován pouze vzhled výstupu, ale data jsou členěna i podle významu.

```

467 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% RUZNE DEFINICE %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
468 \def\soubor#1{\tt #1} % Název souboru
469 \def\texmac#1{\tt #1} % TeXovské makro
470 \def\csrc#1{\tt #1} % céčkovský zdroj/příkaz
471 \def\tclsrc#1{\tt #1} % Tcl příkaz
472 \def\tclargs#1{\it #1} % Tcl parametry příkazu
473 \def\data#1{\tt #1} % data i/o souboru
474
475 \def\httpodkaz #1#2{\tt % internetový odkaz {adresa}{text}
476 \expandafter\ifx\csrcname pdfoutput\endcsrcname \relax #2\else
477 \pdfstartlink attr{/Border [0 0 0]} user{
478 /Subtype /Link
479 /A <<
480 /Type /Action
481 /S /URI
482 /URI (http://#1)
483 >>}\Blue#2\Black\pdfendlink\fi}}
484 \def\ftpodkaz #1#2{\tt % internetový odkaz {adresa}{text}
485 \expandafter\ifx\csrcname pdfoutput\endcsrcname \relax #2\else
486 \pdfstartlink attr{/Border [0 0 0]} user{
487 /Subtype /Link
488 /A <<
489 /Type /Action
490 /S /URI
491 /URI (ftp://#1)
492 >>}\Blue #2\Black\pdfendlink\fi}}
493 \def\mailodkaz #1#2{\tt % mailový odkaz {adresa}{text}
494 \expandafter\ifx\csrcname pdfoutput\endcsrcname \relax #2\else
495 \pdfstartlink attr{/Border [0 0 0]} user{
496 /Subtype /Link
497 /A <<
498 /Type /Action
499 /S /URI
500 /URI (mailto:#1)
501 >>}\Blue #2\Black\pdfendlink\fi}}
502 \def\newline{ \hfil\break } % nový řádek po vzoru LaTeXu
503 \def\newpage{ \vfil\eject } % nová stránka po vzoru LaTeXu
504 \def\{\backslash$} % zpětné lomítko
505 \def\p{\}$} % pravá složená závorka
506 \def\l{\$}\} % levá složená závorka
507
508 % následující část je zde pouze pro zpětnou kompatibilitu
509 % s již existující dokumentací
510 \def\soub{\soubor} % jméno souboru
511 \def\shell #1{\tt #1} % výpis z příkazové řádky
512 \def\cpp{\csrc} % zdrojový kód v jazyce C nebo C++ (není to verbatim!)
513 \def\guim #1{\it #1} % popisek objektu grafického prostředí
514 \def\sip{\$}\rightarrow$} % šipka vpravo (pro popisky hierarchie jako menu apod.)
515
516 \chypch % české dělení slov
517
518 \catcode'\_ =12 % podtržítka je obyčejný znak,
519 \everymath{\catcode'\_ =8 } % ale v matematických režimech
520 \everydisplay{\catcode'\_ =8 } % zůstává jeho význam dolní index
521
522 % Dále definujeme používaná loga \MDef{} a \LaTeX :
523 \def\MDef{M\kern-0.40em\hbox{D}}%
524 \kern-0.55em\lower-0.30ex\hbox{e}}%
525 \kern0.0em\lower0.125ex\hbox{f}}
526 \def\LaTeX{L\kern-.36em\raise0.5ex\hbox{\sevenrm A}\kern-.12em\TeX}

```

Definice makra `\softinput`, které načte soubor pouze v případě, že existuje a nezpůsobuje nepříjemné dotazy TeXu k uživateli:

```

527 \newread\test
528 \def\softinput#1{\immediate\openin\test=#1
529 \ifeof\test\message{Soubor #1 neexistuje...}%
530 \else\immediate\closein\test\input #1\fi}

```

Následující makro je reklamou autora tohoto balíku maker.

```

531 \def\reklama{{~\vfil\newline\baselineskip 7 pt
532 \rightline{\vbox{\halign{\strut\hfil##\cr \noalign{\hrule}

```

```

533 \malit Tento dokument byl zpracován pomocí maker
534 \httpodkaz{www.mandak.net}{\maly \MDef}} v systému
535 \httpodkaz{www.cstug.cz}{\maly plain\TeX}}.\cr
536 % \malit Copyright {\maly\copyright{}} 1999--2003 Martin Mandák
537 % (\mailodkaz{magor@email.cz}{\malit martin@mandak.net}})%
538 % \cr
539 }}}
540 \eject}

```

4. 13. Poznámky

Makro `\poznamka` slouží k vložení poznámky, která bude nahoře a dole oddělena čarou širokou 50 mm (`\hbox to50mm\hrulefill`), mezi těmito čarami bude nadpis „Poznámka:“ a samotný text poznámky odsazený zleva o 10 mm (`\hskip 10mm`) a široký 15 cm, což je o právě těch 10 mm méně než je defaultní šířka odstavce. Po vložení parametru makra musí být ještě `\par`, jinak je hodnota `\hsize` převzata z globální proměnné a ne z vnitřku makra (tady se projevuje nebezpečnost konstrukce `\long\def`).

```

541 \long\def\poznamka #1{\newline\hbox to50mm{\hrulefill}
542 \hfil\break\leftline{\it Poznámka:}}\vskip 6pt%
543 \hbox{\hskip 10mm\vbox{\hsize = 15cm
544 \it #1 \par}}}
545 \hbox to50mm{\hrulefill}\vskip 0pt}

```

4. 14. Tabulky

Tabulky využívají prostředí `\halign` s několika úpravami oproti standardnímu plain \TeX u: především je kategorie znaku tabulátor definována jako 4, tedy stejná jako má znak `&` a lze používat tabulátor jako oddělovač políček tabulky. Další podstatnou změnou je kategorie znaku `0x0D`, která je 13, tedy aktivní znak a jeho význam je nedefinován jako `\cr`. Proto se jako oddělovač řádek v tabulkách nesmí používat `\cr`, pokud je mezi řádky tabulky zařádkováno i ve zdrojovém souboru. K tomu jsou dodefinována makra `\linka` a `\dlinka`, která vytvoří jednoduchou resp. dvojitou vodorovnou linku mezi řádky tabulky. K tomuto účelu obě makra využívají obvyklé `\noalign{\hrule}`. Nejdříve definujeme pomocné makro `\tabulex`, které navíc vysází a započítá číslo a popis tabulky:

```

546 \newcount\cistab
547 \cistab = 0
548 {\catcode'\^^M=13 %
549 \catcode'\^^I=4 %
550 \gdef\tabulex #1^^M#2^^M#3\kontab{\global\advance\cistab by1
551 \hfil\vbox{
552 %\vskip 0 pt
553 \bgroup %
554 \catcode9=4 \catcode13=13 %
555 \def\linka^^M{\noalign{\hrule}}
556 \def\dlinka^^M{\noalign{\hrule\break\vskip 1pt\hrule}}
557 \def^^M{ \cr } %
558 \offinterlineskip{ %
559 \halign{\strut #2 \cr %
560 #3 \cr} } } \egroup
561 \centerline{\it\hskip 0mm Tabulka \thebook\the\cistab: #1}}
562 \egroup \vskip 0pt}

```

Dalším pomocným makrem je `\tabuley`, které vytvoří tabulku bez popisku a očíslování.

```

563 \gdef\tabuley^^M#1^^M#2\konmat{ %
564 \hfil\vbox{
565 %\vskip 0 pt
566 \bgroup %
567 \catcode9=4 \catcode13=13 %
568 \def\linka^^M{\noalign{\hrule}}
569 \def\dlinka^^M{\noalign{\hrule\break\vskip 1pt\hrule}}
570 \def^^M{ \cr } %
571 \offinterlineskip{ %
572 \halign{\strut #1 \cr %
573 #2 \cr} } } \egroup
574 \egroup \vskip 0pt}
575 }

```

Zbývá dodefinovat makra pro začátek a konec obou tabulkových prostředí, která společně s předchozími makry a makrem `\begtabx` vytváří kompletní tabulkové prostředí.

```

576 \def\begtabx{\vskip 0pt\bgroup\catcode'\^^M=13 \catcode'\^^I=4 }
577 \def\zactab{\begtabx \tabulex }
578 \def\zacmat{\begtabx \tabuley}

```

Nakonec jsou samozřejmě definována makra pro reference na tabulku:

```

579 \def\tablabel #1{
580   \immediate\write\REF{\string\tabREF{\the\bookno#1}{\thebook\the\cistab}}
581 \def\tabref #1{\expandafter\ifx\csname tab:\the\bookno#1\endcsname \relax
582   \message{Warning: Undefined tab number [#1]}{??}%
583   \else\csname tab:\the\bookno#1\endcsname\fi}
584 \def\tabREF #1#2{\expandafter\ifx\csname tab:#1\endcsname \relax
585   \expandafter\def \csname tab:#1\endcsname {#2}
586   \else \errmessage{Error: Double tab mark [#1]}\fi}

```

4. 15. Obrázky

Vkládání obrázků do dokumentů dvi, resp. ps se děje pomocí souboru maker `epsf`, které je standardní součástí instalace \TeX U. Do dokumentů pdf je obrázek vkládán způsobem standardním v Han The Thahově $\text{pdf}\TeX$ U, používá se přípona `\obrprrip` definovaná při volání makra `\obrazekprrip`. Obrázek je vkládán vlastně do tabulky — v prostředí `\halign` se snadno umístí včetně jeho popisku:

```

587 \input epsf % soubor epsf.tex by mel byt standartni soucasti texu
588 \newcount\cisobr
589 \cisobr=0
590 \def\obrazekprrip #1{\gdef\obrprrip{#1}}
591 \obrazekprrip{pdf}
592 \newdimen\obrazekx
593 \newdimen\obrazeky
594 \obrazekx Opt \obrazeky Opt
595 \def\obrazek#1#2{\vskip 6pt \global\advance\cisobr by1
596   \centerline{\vbox{\halign{\strut\hfil ##\hfil\cr
597     \ifx\pdfoutput\undefined
598       \epsfxsize \obrazekx \epsfysize \obrazeky
599       \epsfbox{#1.eps} \epsfxsize Opt \epsfysize Opt \cr
600     \else
601       \pdfimageresolution=96
602       \ifdim \obrazekx = Opt
603         \ifdim \obrazeky = Opt \immediate\pdfximage {#1.\obrprrip}
604         \else \immediate\pdfximage height \obrazeky {#1.\obrprrip} \fi
605       \else
606         \ifdim \obrazeky = Opt
607         \immediate\pdfximage width \obrazekx {#1.\obrprrip}
608         \else
609         \immediate\pdfximage width \obrazekx height \obrazeky {#1.\obrprrip}
610         \fi
611       \fi
612       \pdfrefximage \pdflastximage \cr
613     \fi
614     \if #2\relax \else
615       \vbox to8pt{}
616     \line{\hfil{\it Obrázek \thebook\the\cisobr. #2}\hfil}\cr\cr \fi }}}
617   \vskip Opt}

```

A reference na obrázky:

```

618 \def\obrlabel #1{
619   \immediate\write\REF{\string\obrREF{\the\bookno#1}{\thebook\the\cisobr}}
620 \def\obrref #1{\expandafter\ifx\csname obr:\the\bookno#1\endcsname \relax
621   \message{Warning: Undefined pic number [#1]}{??}%
622   \else\csname obr:\the\bookno#1\endcsname\fi}
623 \def\obrREF #1#2{\expandafter\ifx\csname obr:#1\endcsname \relax
624   \expandafter\def \csname obr:#1\endcsname {#2}
625   \else \errmessage{Error: Double pic mark [#1]}\fi}

```

4. 16. Verbatim

Prostředí verbatim jsem v souboru maker věnoval velkou pozornost vzhledem k předpokládanému využití při dokumentování zdrojových kódů programů.

4. 16. 1. Nečíslovaný verbatim

Tato základní verze verbatim prostředí pouze definuje kategorii speciálních znaků jako 12, tedy z nich udělá „obyčejné“ znaky, dále definuje odstavec (`\par`) takovým způsobem, aby se provedlo i vícenásobné zařádkování a nastaví takový význam mezer, že jsou interpretovány i mezery na začátku řádku.

```

626 {\obeyspaces \gdef\activespace{\obeyspaces\let =\ }}
627 {\obeyspaces \gdef\activespace{\obeyspaces\let =\ }}
628 \def\setverb{\def\do##1{\catcode'##1=12}\dospecials}
629 \def\verb{\bgroup \setverb\activespace%
630         \everypar{\hskip 10pt}
631         \def\par##1{\endgraf\ifx##1\par\leavevmode\fi ##1}
632         \parskip=0pt\obeylines\startverb}
633 {\catcode'\|=0 \catcode'\|=12
634  |gdef|startverb#1\everb{|tt#1|egroup}}

```

4. 16. 2. Verbatim s číslováním řádek

Pro číslování řádek ve verbatim prostředí byla zavedeno další makro `\cverb`, které navíc vkládá při každém odstavci (tj. v našem případě na každém zlomu řádku) číslo řádky, které se počítá od začátku dokumentu nebo od volání makra `\begcfile`, které nuluje počítadlo řádek `\cverblines`. Dále se vyhodnocuje definování makra `\verblinesfile`, které pokud je 1, použije se skutečné číslo řádek ve vstupním souboru, jinak se čísluje výše uvedeným způsobem závislým pouze na volání makra `\begcfile`

```

635 \def\setcverb{\def\do##1{\catcode'##1=12}\dospecials}
636 \def\cverb{%
637     \if1\verblinesfile\cverblines=\inputlineno\else\fi
638     \vskip 0pt %\par
639     \bgroup\setcverb\activespace%
640     \everypar{\global\advance\cverblines by1%
641             \hbox to24pt{\hfil\the\cverblines}\qquad}
642     \def\par##1{\endgraf\ifx##1\par\leavevmode\fi ##1}
643     \parskip=0pt\catcode47=12\obeylines\startcverb}
644 {\catcode'\|=0 \catcode' /=12 \catcode'\|=12
645  |gdef|startcverb#1\ecverb{|tt#1|egroup|vskip0pt}}
646 \newcount \cverblines
647 \def\begcfile{\cverblines = 0}
648 \def\verblinesfile{0}

```

4. 16. 3. Vkládání C++ a Tcl zdrojových souborů

Pro tuto funkci je znak `/` resp. `#` definován jako aktivní (catcode 13) s tím, že funguje jako makro (`\ccomment`) a je-li následován dalším lomítkem (v C++), tato sekvence je ignorována. V opačném případě se vysází lomítko i následující znak. Ke vložení souboru slouží 2 makra. `\csrctldocinput` navíc definuje `\verblinesfile` jako 1 a řádky jsou tedy číslovány „skutečně“. Makro `\srcdocinput` pouze vynuluje čítač řádek prostředí verbatim a zdrojový kód C++ programu je číslován, jako by v C++ souboru dokumentace chyběla.

```

649 {\catcode47=13
650 \gdef\ccomment#1{\ifx#1/\relax\else{\char47{ }#1}\fi}
651 \gdef\csrctldocinput #1 {{\def\verblinesfile{1}
652         \def\startcverb{\startcppverb}
653         \catcode47=13 \def/{\ccomment} \input #1 }}
654 \gdef\srcdocinput #1 {{\begcfile
655         \def\startcverb{\startcppverb}
656         \catcode47=13 \def/{\ccomment} \input #1 }}}
657 {\catcode47=12 \catcode'\|=0 \catcode'\|=12
658  |gdef|startcppverb#1//\ecverb{|tt#1|egroup}}

```

Podobně fungují i makra pro vkládání zdrojových Tcl souborů, ingorovány jsou naprosto všechny znaky `#` a navíc bylo nutno změnit `\catcode` znaku `|`, který zde zastupuje znak `#` v definicích maker. Dále jsem musel předefinovat definice `\startcverb`, aby verbatim prostředí obsahovalo i poslední znak `#` před každým ukončením verbatimů.

```

659 {\catcode35=13 \catcode124=6
660 \long\gdef\tclcomment{\relax}
661 \gdef\tcsrctldocinput |1 {{\def\verblinesfile{1}
662         \def\startcverb{\starttclverb}
663         \catcode35=13 \def#/{\tclcomment} \input |1 }}}
664 \gdef\tclsrcdocinput |1 {{\begcfile
665         \def\startcverb{\starttclverb}
666         \catcode35=13 \def#/{\tclcomment} \input |1 }}}
667 {\catcode'\|=12 \catcode'\|=6 \catcode' /=0 \catcode'\|=12
668  /gdef|starttclverb|1#\ecverb{|tt|1|egroup}}

```


4. 17. Parametry T_EXtu

Další definice se týkají definování používaných typů fontů, rozměrů zrcadla pro sazbu, parametrů pro sazbu mezer a penalt pro zalamování v různých částech dokumentu.

```

669 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% DEFINICE FONTŮ PRO rm10 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
670 % je-li definováno \aFive, použijí se parametry pro formát A5
671 \expandafter\ifx\csname aFive\endcsname \relax
672 \font\rm = csr10
673 \font\bf = csbx10
674 \font\it = csti10
675 \font\tt = cstt10
676 \font\sl = cssl10
677 \font\sss = csss12
678 \font\maly = csr8
679 \font\malit = csti8
680 \font\malytt = cstt9
681 \font\big = csr10 scaled\magstep1
682 \font\bigit = csti10 scaled\magstep1
683 \font\bigbf = csbx10 scaled\magstep1
684 \font\bigg = csr10 scaled\magstep2
685 \font\biggg = csr10 scaled\magstep3
686 \font\bigggg = csr10 scaled\magstep4
687 \font\biggggss = csss12 scaled\magstep4
688 \else
689 \font\rm = csr8
690 \font\bf = csbx8
691 \font\it = csti8
692 \font\tt = cstt8
693 \font\sl = cssl8
694 \font\sss = csss10
695 \font\maly = csr7
696 \font\malit = csti7
697 \font\malytt = cstt8
698 \font\big = csr8 scaled\magstep1
699 \font\bigit = csti8 scaled\magstep1
700 \font\bigbf = csbx8 scaled\magstep1
701 \font\bigg = csr8 scaled\magstep2
702 \font\biggg = csr8 scaled\magstep3
703 \font\bigggg = csr8 scaled\magstep4
704 \font\biggggss = csss8 scaled\magstep4
705 \fi
706 \rm
707 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ROZMĚRY TeXTU PRO A4 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
708 \expandafter\ifx\csname aFive\endcsname \relax
709 \hsize 16 cm % šířka
710 \vsize 23.5 cm % výška
711 \hoffset 0 cm % horizontální offset
712 \voffset 0.3 cm % vertikální offset
713 \parindent 0 pt % odsazení začátku odstavce
714 \baselineskip 12 pt % řádkování
715 \parskip 10 pt % proklad při \par
716 \raggedbottom % vynechávat místo na stránce
717 \emergencystretch=4em % natahovatelnost mezer
718 \lineskip=0pt % proklad řádků (vzdálenost boxů)
719 \bigskipamount=12pt % deformovatelné výplňky - velký
720 \medskipamount=6pt % ... - střední
721 \smallskipamount=3pt % ... - malý
722 \widowpenalty=10000 % penalta za vdovy
723 \clubpenalty=10000 % penalta za sirotky
724 \expandafter\ifx\csname pdfoutput\endcsname \relax \else
725 \pdfhorigin=2.53cm
726 \pdfvorigin=2.83cm
727 \pdfpagewidth=21cm
728 \pdfpageheight=29.7cm
729 \pdfdestmargin=2mm
730 % \pdfthreadmargin=12pt%1em
731 % nasledující bude predefinováno v pdfcolor.tex, pokud bude toto ovsem
732 % dostupne... Musi byt definováno az za definice softinput!
733 \def\Blue{} \def\Black{}
734 \softinput{pdfcolor.tex}
735 \fi
736 \else
737 \hsize 12 cm % šířka
738 \vsize 17.0 cm % výška
739 \hoffset -1.0 cm % horizontální offset
740 \voffset -0.8 cm % vertikální offset
741 \parindent 0 pt % odsazení začátku odstavce
742 \baselineskip 10 pt % řádkování
743 \parskip 8 pt % proklad při \par
744 \raggedbottom % vynechávat místo na stránce
745 \emergencystretch=4em % natahovatelnost mezer

```

```

746 \lineskip=0pt           % proklad řádků (vzdálenost boxů)
747 \bigskipamount=10pt    % deformovatelné výplňky - velký
748 \medskipamount=5pt     % ... - střední
749 \smallskipamount=2.5pt % ... - malý
750 \widowpenalty=10000    % penalta za vdovy
751 \clubpenalty=10000    % penalta za sirotky
752 \expandafter\ifx\csname pdfoutput\endcsname \relax \else
753 \pdfhorigin=2.5cm
754 \pdfvorigin=2.5cm
755 \pdfpagewidth=148.5mm
756 \pdfpageheight=21.0cm
757 \pdfdestmargin=1.5mm
758 % \pdfthreadmargin=12pt%1em
759 % nasledujici bude predefinovano v pdfcolor.tex, pokud bude toto ovsem
760 % dostupne... Musi byt definovano az za definice softinput!
761 \def\Blue{} \def\Black{}
762 \softinput{pdfcolor.tex}
763 \fi
764 \fi

```

4. 18. Záhloví a pata stránky

Definice záhlaví a paty stránky je rozdělena do dvou částí:

- pro různé sudé a liché stránky (oboustranný tisk)
- pro všechny stránky stejné

Která varianta se použije je dáno „ručním přepínačem“ — podmínkou `\if` (viz. kód níže).

Definice si kontrolují, zda se kompiluje jen jedna kniha nebo větší celek a podle toho dosazují do záhlaví název knihy nebo hlavní kapitoly, u oboustranného režimu se střídá umístění v pravé a levé části záhlaví. Pata stránky vypadá vždy stejně: je na ní pouze číslo stránky, u oboustranného režimu umístěné střídavě vpravo a vlevo.

Pomocné makro `\headkapname` kontroluje, zda nezačala nová kapitola na aktuální stránce a podle toho dosadí správný řetězec (při kompilaci jediné knihy by měl být v záhlaví jen název první kapitoly na stránce).

```

765 \def\headkapname{\ifnum \lastpagekap = \pageno \lpagefkap\else\lpagekap\fi}
766
767 \if 10 % ,,ruční přepínač“ režimu
768 \headline{\ifodd \the\pageno % Ruzne sude a liche
769 \expandafter\ifx\csname maindoccs\endcsname \relax
770 {\malit\hrulefill~\lower2.3pt\hbox{\headkapname}}
771 \else {\malit\hrulefill~\lower2.3pt\hbox{\thebook\bookname}}
772 \fi
773 \else \expandafter\ifx\csname maindoccs\endcsname \relax
774 {\malit\lower2.3pt\hbox{\headkapname}~\hrulefill}
775 \else {\malit\lower2.3pt\hbox{\thebook\bookname}~\hrulefill}\fi\fi}
776 \footline{\ifodd \pageno \hfill{\bf \the\pageno}
777 \else {\bf \the\pageno}\hfill \fi}
778 \else\headline{ % Vsechny stranky stejne
779 \expandafter\ifx\csname maindoccs\endcsname \relax
780 {\malit\hrulefill~\lower2.3pt\hbox{\headkapname}}
781 \else
782 {\malit\hrulefill~\lower2.3pt\hbox{\thebook\bookname}}
783 \fi}
784 \footline{{\bf\hfill~\the\pageno}}
785 \fi

```

4. 19. Načtení referencí

V tomto místě je již možno načíst reference z předchozího běhu \TeX u, protože jsou již definována všechna makra všech typů referencí. Po jejich načtení je soubor `jobname.ref` smazán, resp. otevřen jako prázdný pro další zápis referencí v aktuálním průchodu \TeX procesoru.

```

786 \softinput{jobname.ref}
787 \immediate\openout\REF=jobname.ref

```

4. 20. Titulní stránka knihy

Jako titulní stránku knihy vysází makro `\titulknihy` uprostřed samostatné stránky tělo maker `\nazev` a `\podnazev`, dále vpravo dole obsah makra `\autor`. Na titulní stránce je vypnuta hlavička i patička stránky i očíslování číslem stránky. Makro `\titulknihy` je voláno makrem `\kniha`, které definuje všechny tři potřebné údaje pro jeho sazbu.

```

788 \def\titulknihy{%
789 \headline{\hfil} \footline{} \nopagenumbers \vbox to3cm{ } \vfil
790 \centerline{\bigg\nazev}\vskip 6pt \centerline{\big\npodnazev}
791 \vfil \vbox to5cm{ } \rightline{\autor} \eject}}

```

4. 21. Titulní stránka dokumentace

Titulní stránka celé dokumentace složené z více knih se sází podobně jako titulní stránka knihy s tím rozdílem, že je uživatelem přímo voláno makro `\maindoc`. Nejdříve v něm definuji kontrolní sekvenci `maindoccs` a v případě, že byla tato již definována, ohlásí T_EX chybu. Dále je definován `\konecknihy` jako `\endinput`, tedy ukončení souboru s knihou (v případě kompilace pouze jedné knihy je toto makro definováno jako `\end`, aby se ukončil kompletně běh T_EXu). Na kontrolní sekvenci `maindoccs` pak testují různá makra, zda se mají chovat jako v jediné knize nebo větším dokumentačním celku.

```

792 \def\maindoc #1#2#3{%
793 \expandafter\ifx\csname maindoccs\endcsname \relax
794 \expandafter\def \csname maindoccs\endcsname {}
795 \else \errmessage{Error: Nelze mit dvakrat maindoc}\fi
796 \def\konecknihy{\endinput}

```

Následující část makra `\maindoc` sází titulní stránku dokumentace podobným způsobem jako makro `\titulknihy`,

```

797 {\headline{\hfil}\footline{} \nopagenumbers
798 ~ \vfil\newline
799 \centerline{\vbox{\halign{\strut\hfil##\hfil\cr
800 {\bigg #1}\cr~\cr{\big #2}\cr}}}
801 \vfil\eject\pageno = 1
802

```

ale s tím rozdílem, že místo `\autor` se sází na samostatné stránce anotace (umísťují se zde např. autoři, vydavatel, copyright...), zde je použito prostředí `\halign`:

```

803 \bgroup
804 ~\vfil\newline\baselineskip 10 pt \malit
805 \leftline{\vbox{\halign{ ##\hfil\cr #3 \cr\cr}}}
806 \egroup
807 \vskip 1.5cm\eject}}

```

Tím popis je algoritmů dokumentačních maker ukončen. V následující kapitole se ještě zmíním o možnostech samodokumentování tohoto balíku.

5. Systém autodokumentace

Sám sebe dokumentující systém mě napadl při zevrubném seznámení se s balíkem `docstrip`. Rozhodl jsem se tedy něco podobného navrhnout při dokumentování mého systému `Mf`.

5. 1. Princip autodokumentace

Dokumentace funguje podobně jako prostředí verbatim, začátek a konec zdrojového kódu se označuje makry `\defverb` a `\edefverb` a to, co není mezi těmito značkami by mělo být „schováno“ za procentem na začátku řádku, které je při „dokumentačním“ průchodu \TeX em ignorováno a při běžném načítání souboru `mdef.tex` se považuje za běžný komentář. Tím je docíleno, že při prvním průchodu souboru `mdef.tex` se celá dokumentace ignoruje (Uživatelská část je postavena do podmínky `\if`, kde jsou dvě za sebou jdoucí definice stejného makra, jedna z nich je v komentáři a tudíž se definuje pouze při samodokumentování, u popisu maker je použito zmíněné procento všude). Při souhlasném `\jobname` s názvem definičního souboru maker se předpokládá generování dokumentace sebe sama a po dodefinování výše zmíněných souvislostí zavolá definiční soubor sám sebe, což způsobí, že text, který byl původně komentářem je dokumentací a označený zdrojový text definic formátů a příkazů je vypsán jako číslovaný verbatim.

Zahrnutí popisu tohoto dokumentačního systému do jiné dokumentace jako kniha je možné pomocí makra `\autodoc` (resp. `\autodocsrc` nebo `\autodocman`, viz. kapitulu 2. 13. *Ostatní příkazy*) v případě, že soubor `mdef.tex` se nachází v aktuálním adresáři (z nějž je \TeX spuštěn).