

OpBible

automatická sazba studijních Biblí T_EXem

Verze 1.0

Petr Olšák, Šimon Konečný

Revidováno 25. 2. 2023

<https://www.overleaf.com/project/62e3832c89a3bfe389d592ba>

*Nestačí, abys byl mým služebníkem,
který má pozvednout Jákobovy kmeny
a přivést zpátky ty z Izraele, kdo byli ušetřeni;
dal jsem tě za světlo pronárodům,
abys byl spása má do končin země.*

Iz 49:6

Kolekce maker OpBible může být volně kopírována, šířena a používána v souladu s GNU General Public Licence (Verze 2, 1991).

Můžete si přenést části tohoto softwaru do svých vlastních maker a/nebo učinit je součástí jiných balíčků, avšak balíček, jakkoliv se liší od původní distribuce, se nesmí jmenovat OpBible.

Adaptace specifických implementací (např. fontů) jsou považovány za doplňující soubory a jejich začlenění neporušuje licenci.

Předmluva

Verš z Iz 49 na titulní straně je inspirující: Bylo by málo napsat a vysázet \TeX em jenom jeden komentář k Bibli coby něčí celoživotní dílo. OpBible nabízí možnost napsat komentující poznámky k jakékoliv Bibli komukoliv kdekoliv; jakmile je poznámkový aparát dokončen, studijní Bible je vysázena.

Dlouholeté vize Šimona Konečného o použití \TeX u k automatické sazbě studijních Biblí došly svého naplnění spoluprací navázanou s Petrem Olšákem. Vycházejíce z realistického předpokladu, že ti, kdo budou OP-Bibli nejvíce využívat, budou zároveň těmi, kdo mají s \TeX em nejméně zkušeností, snažili jsme se učinit obsluhu co možná nejjednodušší.

Nicméně nemá smysl předstírat, že sazba studijní Bible je něco triviálního. Prohlásit, že je to programátorský „oříšek“, je těžce podsazené. Je to spíše kokosová palma, obsypaná zralými plody.

Náš záměr umožnit vytvoření jednoho komentářového aparátu pro více biblických překladů současně byl navíc komplikován skutečností, že různé verze Bible mají v některých místech různá číslování veršů, jinak rozlámané odstavce, jinou strukturu poetických pasáží, jiné rozmístění nadpisů, doplněných překladateli; že některé překlady mají zcela jiné názvy knih (a tudíž i jejich zkratky), že v názvech souborů nelze mít diakritiku, ale v odkazech na knihy ano a bezpočtem dalších podobných specialitek.

Všechny problémy, na které jsme narazili, se podařilo překonat a všechno funguje k naší spokojenosti. Výsledná kolekce nástrojů je značně komplikovaná, pročež použití jejich definic je až nedůtklivě choulostivé na prohřešky proti syntaxi \TeX u.

Proto OpBible obsahuje i nástroje k snadnějšímu vyhledání míst, kde zůstaly překlady, zapomenuté závorky apod.; a v této dokumentaci navíc tam, kde jsme to uznali za vhodné, připojujeme odstavčky nadepsané **Na co si dát pozor a Doporučujeme**.

Petr Olšák
Šimon Konečný

Obsah

Předmluva	3
1 Čím je OpBible specifická?	5
2 Co potřebujeme k provozu OpBible	6
2.1 T _E Xová distribuce na lokálním počítači	6
2.2 Varianta: vzdálený přístup na serveru OverLeaf.com	6
3 Spuštění T _E Xu	6
4 Struktura souborů zpracovávaných T _E Xem	7
4.1 Hlavní soubor s údaji o všech dalších souborech	7
4.2 Soubor deklarující varianty překladu	9
4.3 Soubor s názvy knih	10
4.4 Formát základního textu, tzv. .txs soubory	11
4.5 Údaje určující formátování základního textu v souborech fmt-*.tex	12
4.6 Poznámky v souborech notes-*.tex	14
5 Poznámky a další objekty s vazbou na základní text	14
5.1 Poznámky vázané na fráze základního textu: příkaz \Note	14
5.2 Příkazy pro vložení dalších objektů	15
5.3 Vložení obrázků	15
5.4 Vložení článků	16
5.5 Vložení citací do horní části stránky	16
5.6 Vložení citací do okraje v textu článku	16
6 Různé (ale podobné) verze základního textu	17
6.1 Deklarace variant překladu a použití pomoci příkazu \x	17
6.2 Variantní deklarace frází pro spárování poznámek s textem	18
6.3 Větvění zpracování textu podle variant překladu	19
6.4 Přechíslování veršů podle variant překladu	19
7 Metody vytváření hyperlinkových odkazů	20
7.1 Základní pravidlo s úplnými údaji	20
7.2 Specifikátor odkazu	21
7.3 Výjimka pro formát úplného údaje u některých knih	21
7.4 Neúplný údaj	21
7.5 Formát pro rozsah veršů a pro úsek ve verši	22
7.6 Zatajení údaje	22
7.7 Přechíslování odkazu	22
7.8 Redukce údajů	22
7.9 Značka pro knihu se může vytisknout jinak	23
7.10 Nepodařené odkazy, tj. odkazy na neexistující místo	23
7.11 Trasování odkazů	23
7.12 Odkazy na kapitoly a na knihy	23
7.13 Odkazy na stránky	23
8 Mapy, obrázky a jejich legendy	23
8.1 Varianty překladu	23
8.2 Makro \town pro symbol města na mapě	24
8.3 Nakloněný text	25
8.4 Nápis podél křivky	25
8.5 Částečně průhledné pozadí souvislého textu	25
9 Nástroje na zařazení časových os	26
9.1 Obrázek či text přes dvě strany	26
9.2 Příkazy k vytvoření časové osy	26
10 Varianty formátování stránek	27
11 Možnosti hledání chyb	27
12 Generování výchozích souborů notes, fmt, intro	28
13 Shrnutí uživatelských definic	29
14 Rejstřík	29

1 Čím je OpBible specifická?

Mezi hlavní přednosti OpBible oproti tradiční sazbě patří:

- **Cena:**
 - Software samotný je zdarma jako public domain pod licencí Open Source (viz Předmluva).
 - Mnohem významnější však je cena, kterou ušetříte za sazeče, kterého byste jinak museli platit několik let, než studijní Bibli vysází ručně.
- **Čas:** Sazba celé studijní Bible netrvá řádově roky, nýbrž minuty. Opatřit úplný text Bible poznámkovým aparátem, který rozláme stránky tak, aby poznámky skončily na stejných stranách s verši, které komentují, trvá na systému s čtyřjádrovým procesorem Ryzen 3 kolem deseti minut. Pracujete-li na konkrétní knize a zpracováváte jen tu jednu, bývá to otázka sekund.
- **Flexibilita:** Výsledkem není jedna jediná verze Bible, do které již není možné provést žádné další zásahy. Pokud se rozhodnete některé poznámky upravit, doplnit, připsat další apod., máte tuto novou revidovanou verzi okamžitě vysázenou a připravenou k tisku.
- **Variabilita:** Výsledkem nemusí být jen jedna Bible. Poznámky lze psát způsobem, který umožňuje vysázet se stejným poznámkovým aparátem tolik Biblí, kolik v daném jazyce existuje různých překladů. Vyjadřuje-li se Vaše poznámka k babylonskému králi Nabuchodonozorovi, poznámka uvede jméno Nabuchodonozor, pokud sázíte studijní Bibli kralickou, ale pokud přepnete na Ekumenický překlad, poznámka změní jméno krále na Nebúkadnesar. V češtině máme nejméně 6 kompletních překladů Bible (z toho nejméně 4 snadno stažitelné); svůj komentář můžete napsat tak, že výsledkem bude 6 různých studijních Biblí naráz. Podrobněji viz kapitola 6.
- **Interaktivní výstupní formát.** Výsledkem zpracování je PDF soubor, který je bohatě propojen aktivními odkazy. Ty jsou nedocenitelným pomocníkem pro proof-reading (korektury) před odevzdáním do tisku (viz níže).
- **Preciznost:**
 - Průměrná studijní Bible obsahuje kolem 20.000 poznámek a v nich kolem 80.000 odkazů na biblické verše a jiné poznámky. Poznámky píšou lidé; je nevyhnutelně zákonité, že budou obsahovat chyby z přehlédnutí či překlepu. Prakticky není v lidských silách je všechny najít a opravit. Je to nicméně v silách stroje. OpBible vypisuje varovná hlášení, pokud například poznámka odkazuje na neexistující poznámku nebo verš.
 - Kontrolu správnosti odkazů na verše, a poznámky, které sice existují (takže nevyvolají chybové hlášení), ale z nějakého důvodu to nebyly ty správné, umožňuje skutečnost, že všechny odkazy jsou aktivní linky, které při zastavení kurzoru zobrazí bezprostřední kontext odkazovaného místa (pokud PDF prohlížíte například správnou verzí programu Evince). A při kliknutí tam rovnou skočí.
 - Fráze, kterou poznámka komentuje, se zvýrazní odlišným řezem fontu (např. tučně). Zároveň se však i vyhledá v biblickém textu příslušného verše a stránky se budou zalamovat tak, aby poznámka byla na stejné stránce se svou frází, nikoliv jen podle čísla verše. Může se stát, že dvě poznámky ke stejnému verši komentují dvě různá slova onoho verše a že stránkový zlom bude vycházet mezi těmito slovy. V takovém případě budou poznámky následovat své fráze.
 - Biblické texty se připraví (například stažením z internetu) do zvláštních souborů a není třeba do nich vůbec nic přidávat. Všechny poznámky k nim a další pokyny k sazbě se zapisují do oddělených souborů. Pak to \TeX spojí dohromady.

Na co si dát pozor: Ačkoliv záměr tohoto programu je šířit Boží Slovo a dobrou zprávu v něm obsaženou, OpBible sama *neodpouští nic!* (Co byste taky čekali, je to software. Pro odpuštění jděte k Bohu.) Jak se dočtete dále, OpBible načte do paměti vždy jednu celou biblickou knihu (tzv. základní text) a pak teprve začne chroustat Vaše poznámky a skládat je dohromady na stránky s příslušnými verši. Když se Vám stane, že se dopustíte chyby v \TeX ovské syntaxi (např. zapomenete uzavírací složenou závorku apod.), \TeX chybu uvidí někde uprostřed této přiřazovací procedury a tomu taky bude odpovídat nepřehledné chybové hlášení, ve kterém je vysoce pravděpodobné, že se neorientujete.

OpBible s výskytem podobných situací počítá a nabízí užitečné nástroje, jak Vám pomoci z podobných brynd, ale raději buďte pozorní a důslední. Neprohloupíte, když s první napsanou poznámkou projedete všechny překlady, které plánujete používat, abyste se přesvědčili, že všechno funguje, jak má. Pak stačí pracovat jenom s jedním z nich, ale je radno spouštět \TeX po dokončení každé nové poznámky, ať podle jeho protestů snadno najdete tu, ve které máte chybu.

Doporučujeme: Používejte operační systém Linux. OpBible, coby soubor maker pro \TeX , poběží na kterémkoliv systému se stejnými výsledky, ale instalace na Linuxu má několik nepřehlédnutelných výhod. Mezi ty nejvýznamnější je nutno zařadit PDF prohlížeč Evince, jehož novější (píše se rok 2023) verze dokáže zobrazit bezprostřední kontext hyperlinkového odkazu při pouhém zastavení kurzoru bez nutnosti na něj klikat. To je nedocenitelná pomůcka pro kontrolu správnosti odkazů na biblické pasáže nebo poznámky. Jiné systémy tuto vymoženost (zatím) nenabízejí. Hyperlinky v nich sice skočí na příslušné místo po kliknutí, ale už netrefí zpátky, odkud byl vyslán signál ke skoku, což je únavné. V této dokumentaci předpokládáme instalaci na Linuxu.

A taky samozřejmě doporučujeme – nebo spíše i považujeme za nezbytnou podmínku – alespoň základní gramotnost v programu \TeX . Jestli s \TeX em nemáte vůbec žádné zkušenosti, zkuste začít třeba [tady](#). Čas, investovaný do tohoto preliminárního vzdělání, nebude promarněný; čím více budete \TeX u rozumět, tím lépe (a radostněji) budete psát své poznámky k Bibli.

2 Co potřebujeme k provozu OpBible

Je třeba mít nějaký textový editor, který nezanechává v textu žádné skryté formátovací údaje (určený k editování například programů). Záleží na uživateli, jaký editor mu vyhovuje. Za ideální lze považovat takové, které podle přípony zdrojového souboru poznají programovací jazyk a barví slova podle jeho syntaxe. Např. Vi, Emacs, ... [vyberte si](#), který by Vám vyhovoval.

Dále je třeba mít \TeX ovou distribuci s programem Lua \TeX a s výchozími makry Op \TeX a konečně je třeba mít nějaký prohlížeč výstupu, tedy PDF souborů (doporučujeme novou verzi Evince).

Na jakém operačním systému to budete provozovat, není podstatné, doporučujeme ale [Linux](#).

Je také možné pracovat v online režimu bez nutnosti cokoli instalovat, viz sekci [2.2](#).

2.1 \TeX ová distribuce na lokálním počítači

Doporučujeme nejnovější [\$\text{\TeX}\$ live](#). Distribuci lze instalovat na libvolném operačním systému a to přímo z webu. Je též součástí běžných linuxových distribucí.

\TeX live obsahuje program Lua \TeX , který bude zpracovávat Vaše vstupní soubory a vyrábět výstupní PDF. Také obsahuje výchozí sadu maker [Op \$\text{\TeX}\$](#) , kterou makra OpBible potřebují a rozšiřují ji o možnosti popsané v tomto manuálu. Výchozí sada maker (často nazývaná také formát) vymezuje způsob značkování dokumentů a řídí formátování. \TeX ové distribuce obsahují i jiné výchozí sady maker. Nejznámější je zřejmě \LaTeX , ale ten OpBible nepotřebuje.

2.2 Varianta: vzdálený přístup na serveru OverLeaf.com

Overleaf je webové prostředí pro sdílenou přípravu \TeX ových dokumentů třeba i více uživateli. Nemusíte mít \TeX live instalován a může jej provozovat online prostřednictvím [Overleaf](#). Je k dispozici i **TODO: výchozí projekt** obsahující soubory pro zpracování pomocí OpBible. Jako uživatelé Overleafu si jej můžete zkopírovat (klonovat) do svého projektu a tam pokračovat v činnosti. Práce s \TeX ovou distribucí přímo v počítači je ale výrazně rychlejší a pohodlnější.

3 Spuštění \TeX u

Máte-li nainstalovanou \TeX ovou distribuci na svém počítači (například \TeX live 2022) a máte-li v systému k dispozici příkazový řádek, pak \TeX spustíte pomocí:

```
optex main.tex
```

kde `main.tex` je název hlavního souboru (může se jmenovat i jinak). Příkaz `optex` spustí Lua \TeX s makry Op \TeX . Výsledkem zpracování je soubor `main.pdf` a zpráva o zpracování je v log souboru `main.log`.

Můžete vyzkoušet, jestli Vám to funguje (zatím bez maker OpBible). Vytvořte si jednoduchý soubor `main.tex` v textovém editoru s tímto obsahem:

```
\fontfam[1m]
Ahoj světe!
\bye
```

a spusťte příkaz `optex main.tex`. Výsledný PDF soubor `main.pdf` si prohlédněte PDF prohlížečem.

4 Struktura souborů zpracovávaných T_EXem

4.1 Hlavní soubor s údaji o všech dalších souborech

Hlavní soubor je soubor, který se T_EXu předloží jako první. Například je uveden na příkazovém řádku pro spuštění T_EXu. V něm jsou informace, jaké další soubory si má T_EX přečíst. Nakonec T_EX vytvoří soubor PDF stejného názvu, jako je název hlavního souboru.

Hlavní soubor pro použití OpBible (například `main.tex`) může vypadat takto:

```
\load[opbible] % makra OpBible
\cslang          % inicializace českých vzorů dělení slov

% Varianty překladu:
\def\tmark      {BKR}      % Bible kralická
%\def\tmark      {PSP}      % Pavlíkův studijní překlad
%\def\tmark      {CSP}      % Český studijní překlad
%\def\tmark      {CEP}      % Český ekumenický překlad
%\def\tmark      {B21}      % Bible pro 21. století
%\def\tmark      {SNC}      % Slovo na cestu

\input {Cze-vars.tex} % Deklarace frází pro různé varianty překladu
\input {Cze-books.tex} % Tituly knih a jejich značky \amark

\def\txsfile      {sources/Cze\tmark-\amark.txs}      % Umístění txs souborů
\def\fmtfile      {formats/fmt-Cze\tmark-\amark.tex} % Umístění fmt souborů
\def\notesfile    {notes/notes-\amark.tex}            % Umístění souborů s poznámkami
\def\introfile    {others/intro-\amark.tex}           % Umístění soubotů s úvody ke knihám
\def\articlefile  {others/articles-\amark.tex}        % Umístění soubor s články

\def\printedbooks {%
  Gn Ex Lv Nu Dt Joz Sd Rt 1Sa 2Sa 1Kr 2Kr 1Pa 2Pa Ezd Neh
  Est Jb Ž Př Kaz Pís Iz Jr Pl Ez Da Oz Jl Am Abd Jon Mi
  Na Abk Sf Ag Za Mal
  Mt Mk Lk Jn Sk Ř 1K 2K Ga Ef Fp Ko 1Te 2Te 1Tm 2Tm
  Tt Fm Žd Jk 1Pt 2Pt 1Jn 2Jn 3Jn Ju Zj
}
\processbooks % Generuje dokument s knihami deklarovanými v \printedbooks
\bye
```

Pojďme se nyní podívat, co který z těch řádků dělá a které budou vyžadovat úpravy z Vaší strany pro konkrétní potřeby Vašeho projektu.

Pomocí `\load[opbible]` načte T_EX makra balíčku OpBible. To je ten nejdůležitější program, který se stará o sazbu.

Příkaz `\cslang` nastaví české vzory dělení slov, předpokládá se tedy český text. Přitom `cs` je ISO zkratka jazyka, můžete použít i jiné jazyky: `\enlang` pro angličtinu, `\delang` pro němčinu, `\eslang` pro španělštinu atd. Všechny tyto jazykové možnosti jsou uvedeny v [dokumentaci k OpT_EXu](#).

Příkaz `\def\tmark {{značka}}` definuje makro `\tmark` jako značku použitého překladu (můžete si pamatovat Translation mark). Značky všech dostupných překladů jsou uvedeny v souboru `Cze-vars.tex`. Jednu z nich je třeba vybrat jako označení zrovna zpracovávaného překladu. Např. BKR je značka pro překlad Bible kralické. V ukázce jsou uvedeny všechny obvyklé možnosti pro definici překladové značky v případě českých Biblí. Jen jedna možnost (ta skutečně vybraná) nemá před sebou znak uzavírací komentář `%`.

Pokud prohlížíte právě rozpracovanou knihu v překladu BKR, ponechte soubor `main.tex` ve výše uvedené podobě. Až budete chtít přepnout do řekněme ekumenického překladu, znakem procenta zakomentujete (tj. zneviditelníte pro T_EX) řádek s Biblí kralickou, ale zviditelníte (odkomentujete) řádek s ekumenickým překladem. Pak úsek s definicemi `\tmark` bude vypadat takto:

```
% Variants of translation:
%\def\tmark      {BKR}      % Bible kralická
%\def\tmark      {PSP}      % Pavlíkův studijní překlad
```



```

%\def\tmark      {CSP}      % Český studijní překlad
\def\tmark       {CEP}      % Český ekumenický překlad
%\def\tmark      {B21}      % Bible pro 21. století
%\def\tmark      {SNC}      % Slovo na cestu

```

Vždy však musí být jeden z překladů aktivní, čili `\tmark` musí být definována. Kdybyste zapomněli napsat procento před řádek, který chcete zrušit, svět se nezboří; bude platit poslední definice, kterou \TeX načte a kterou předefinuje případné předchozí.

`\input {Cze-vars.tex}` přečte konfiguraci o variantách překladu ze souboru `Cze-vars.tex`. Soubor je v ukázce nazván `Cze-vars.tex`, protože obsahuje informace o českých překladech. Podrobněji viz sekci 4.2. Nesahat na tento řádek, byť asi budete editovat soubor `Cze-vars.tex`, tímto řádkem volaný.

`\input {Cze-books.tex}` přečte informace o značkách (zkratkách) knih Bible a jsou jim přiřazeny názvy knih. Podrobněji o těchto údajích pojednává sekce 4.3.

Makrem `\txsfile` (definovaným pomocí příkazu `\def`) je určeno umístění `.txs` souborů v adresářové struktuře. Pro každou knihu Bible musí být připraven jeden `.txs` soubor obsahující základní text pro příslušnou knihu. Názvy souborů se liší podle značky knihy a máme-li více překladů, obsahuje název souboru také značku překladu. V makru `\txsfile` lze použít `\tmark` jako značku překladu a `\amark` nebo `\bmark` jako značku knihy. O značkách knih se píše v sekci 4.3, o formátu `.txs` souborů se píše v sekci 4.4. V ukázce je vymezeno, že `.txs` soubory jsou umístěny v adresáři `sources/` a mají názvy `Cze<značka-překladu>-<značka-knihy>.txs`, například tedy `CzeBKR-Gn.tex`.

Zmíněné české překlady jsou již připraveny k použití, nemusíte je vytvářet sami. Pokud byste potřebovali nějaký jiný existující překlad, musíte ho dostat do formátu, použitelném pro OpBibli, tedy do podoby, v jaké jsou soubory `*.txs` v adresáři `sources/`. S přípravou jednotlivých `.txs` knih z existujícího zdroje Vám pomůže skript `maketxs`, viz 4.4. Pokud tvoříte zbrusu nový překlad a plánujete ho pro použití s OpBibli, asi neuškodí ho rovnou psát po jednotlivých souborech pro každou knihu přímo v požadovaném formátu, viz taktéž 4.4.

Makrem `\fmtfile` je vymezeno umístění souborů upřesňující formátování základního textu. Každá kniha Bible každého použitého překladu má svůj formátovací soubor. To je něco, co nemůže být společné všem překladům (na rozdíl od poznámek), protože odstavce a doplněné nadpisy má každý překlad své vlastní. Naším záměrem byla tzv. non-destruktivní editace, jinými slovy, formátování biblického textu bez zásahu do něho. O těchto souborech se píše v sekci 4.5.

Makrem `\notesfile` je vymezeno umístění souborů s poznámkami. Do nich budete psát svůj komentář. Každá kniha Bible má svůj soubor s poznámkami. Poznámky odkazují na místo v základním textu a úkolem \TeX u je vytvořit stránky se základním textem a poznámkami k němu. Podrobněji o způsobu psaní souborů s poznámkami se píše v sekci 5.1. Všimněte si, že soubory s poznámkami jsou společné všem překladům, tj. nejsou zde zvlášť soubory odlišené pomocí `\tmark`. Pravidla psaní poznámek totiž umožní obsáhnout vyjadřovací možnosti různých překladů společným zápisem, ovšem v jednom jazyku (například češtině). Chcete-li psát poznámky pro zcela jiný jazyk, je třeba založit nový projekt (nejlépe v jiném adresáři) s jinými `.txs` soubory, jinými formátovacími a poznámkovými soubory.

Makrem `\introfile` se specifikují soubory, kde jsou napsány úvody k jednotlivým knihám. Ke každé knize je možné vytvořit samostatný soubor s úvodem.

Makrem `\articlefile` se určují názvy souborů kde, jsou teologické články, rozmístěné různě po biblických knihách.

Makro `\printedbooks` obsahuje značky těch knih, které chcete \TeX em zpracovat. V ukázce je požadavek na zpracování celé Bible, tedy všech 66 knih protestantského kánonu. Pokud děláte třeba jen testovací tisky, můžete zpracovat jen některé biblické knihy a mít v hlavním souboru tedy alternativní definici, například `\def\printedbooks{Da}`. Stačí ji uvést za definici pro celou Bibli, protože pozdější definice stejného makra má přednost.

Příkaz `\processbooks` spustí zpracování všech knih specifikovaných v makru `\printedbooks`. Pro každou knihu \TeX přečte příslušný základní text z `.txs` souboru, formátuje jej pomocí údajů z příslušného formátovacího souboru a připojí k nim poznámky z příslušného poznámkového souboru. Tady už nepotřebujete nic měnit.

Příkaz `\bye` ukončí činnost \TeX u. Cokoliv napíšete za toto rozloučení s \TeX em, bude ignorováno.

Do hlavního souboru před `\processbooks` můžete přidávat také vlastní makra a nastavení, která ovlivní kompletně celou sazbu.

Je možné tam například dát deklaraci `\ChapterPre{<kód>}` anebo `\ChapterPost{<kód>}`. Tyto kódy se pak provedou před a za každou kapitolou.

4.2 Soubor deklarující varianty překladu

Pracujeme-li s jedinou variantou překladu, není nutné tento soubor vytvářet a používat. Pak stačí odstranit pokyn k jeho čtení z hlavního souboru.

V ukázce v sekci 4.1 je čten soubor `Cze-vars.tex`, který by měl obsahovat deklaraci značek variant překladů pomocí `\variants`:

```
\variants <počet-variant> {<značka>} {<značka>} ... {<značka>}
```

kde `<počet-variant>` je počet variant překladů (do společného jazyka, například češtiny) a pak jsou vyjmenovány všechny značky variant překladů. Například

```
\variants 6 {BKR} {PSP} {CSP} {CEP} {B21} {SNC}
```

deklaruje zkratky pro 6 variant překladů: BKR: Bible kralická, PSP: Pavlíkův studijní překlad, CSP: Český studijní překlad, CEP: Český ekumenický překlad, B21: Bible pro 21. století, SNC: Slovo na cestu.

Takto definované varianty překladu musejí odpovídat definicím `\tmark` v hlavním souboru `main.tex` včetně velkých či malých písmen.

Dobře zvažte předem, jaký počet překladů budete chtít (měnit to později bude značně obtížné, byť přeci jen možné) a hlavně pořadí překladů: Jak to bude uvedeno v definici `\variants`, tak to bude platit pro celý projekt. Do všech poznámek, komentujících nějakou frázi, která zní v různých překladech různě, budete její různá znění uvádět právě v tomto pořadí.

Pokud víte, že se nějaká fráze či slovo bude vyskytovat častěji, než jen v jediné poznámce, můžete ji nadefinovat rovnou v tomto souboru `Cze-vars.tex` pomocí příkazu `\vdef`. Frází uvedených za `\vdef` musí být přesně tolik, jako je `<počet-variant>`, jsou uzavřeny do složených závorek a korespondují s jednotlivými variantami překladu ve stejném pořadí, jak jsou varianty uvedeny v deklaraci `\variants`. Například:

```
\vdef {Joakim} {Jehójákím} {Jójákím} {Jójakím} {Joakim} {Jójakím}
```

deklaruje, že jméno Judského krále Joakima je přepisováno jinak v různých variantách překladu. Například v CEP nebo v SNC to je Jójakím, Pavlík jej nazývá Jehójákímem atd. Až budeme psát poznámky týkající se tohoto krále do poznámkového souboru, napíšeme do textu poznámky jen `\x/Joakim/` (první varianta překladu v definici `\variants`) a toto se promění v odpovídající frázi používanou v právě zpracovávaném překladu, který jsme deklarovali v hlavním souboru pomocí `\def\tmark{...}`.

Takže po změně `\def\tmark{...}` v hlavním souboru se všechny výskyty `\x/Joakim/` v textech poznámek mohou začít automaticky chovat jinak a přizpůsobí se frazeologii dané varianty překladu. Pak lze taková slova třeba i skloňovat: Zápis `\x/Joakim/ovi` dá v poznámce pod Českým ekumenickým překladem podobu Jójakímovi. Podrobněji o tom pojednává sekce 6.

Příkaz `\variants` deklarující zkratky variant překladu je v souboru pro varianty jediný, zatímco příkazů `\vdef` definujících variantní fráze může být v souboru více, protože pochopitelně existuje více frází, které se v různých variantách překladu různě přepisují, není to jen král Joakim.

Celé pasáže textu mohou být zpracovány rozdílným způsobem v závislosti na nastavené variantě překladu. K tomu slouží příkaz větvení `\switch`. Podrobněji se o něm píše v sekci 6.3. Například názvy jednotlivých překladů (které se pak použijí v záhlaví stránek) je možné deklarovat pomocí `\def\bibname` rozdílně pro různé překlady:

```
\switch {BKR}{\def\bibname{Bible kralická}}%
        {B21}{\def\bibname{Bible pro 21. století}}%
        {CEP}{\def\bibname{Český ekumenický překlad}}%
        {CSP}{\def\bibname{Český studijní překlad}}%
        {PSP}{\def\bibname{Pavlíkův studijní překlad}}%
        {SNC}{\def\bibname{Slovo na cestu}}%
```

Taková deklarace je součástí již připraveného souboru `Cze-vars.tex`.

4.3 Soubor s názvy knih

V hlavním souboru je pokyn na čtení souboru s názvy knih, například `\input {Cze-books.tex}`. Tento soubor musí obsahovat příkazy `\BookTitle` ve formátu:

```
\BookTitle <a-značka> <b-značka> {\<nezkrácený titul knihy>}
```

Mezi značkami a titulem knihy musí být aspoň jedna mezera. Začátek takto čteného souboru může vypadat třeba takto:

```
\BookTitle Gn    Gen    {První Mojžíšova (Genesis)}
\BookTitle Ex    Exod   {Druhá Mojžíšova (Exodus)}
\BookTitle Lv    Lev    {Třetí Mojžíšova (Leviticus)}
\BookTitle Nu    Num    {Čtvrtá Mojžíšova (Numeri)}
\BookTitle Dt    Deut   {Pátá Mojžíšova (Deuteronomium)}
\BookTitle Joz   Josh   {Jozue}
\BookTitle Sd    Judg   {Soudců}
...
```

V prvním sloupci za `\BookTitle` jsou *<a-značky>*, které jsou dále používány v textu poznámek a jsou pomocí nich tvořeny odkazy na různá místa v Bibli.

Ve druhém sloupci jsou *<b-značky>*, které mohou být stejné jako *<a-značky>*, ale mohou se také lišit. Je třeba možné, že názvy `.txs` souborů vznikly exportem z nějakého softwaru a značky knih jsou tam jiné, než potřebujeme použít v textu našich poznámek. Pak je možné v hlavním souboru deklarovat umístění `.txs` souborů s využitím `\bmark` místo `\amark`, tedy

```
\def\txsfile    {sources/Cze\txmark-\bmark.txs}
```

a mít soubory `CzeBKR-Gen.tx`s, `CzeBKR-Exod.tx`s, zatímco v textu používáme značky `Gn`, `Ex`, atd., nikoliv `Gen`, `Exod`.

Makro `\amark` obsahuje *<a-značku>* právě zpracovávané knihy a dále makro `\bmark` zahrnuje *<b-značku>* právě zpracovávané knihy.

Poznamenejme, že makro `\printedbooks` (v souboru `main.tex`) se značkami všech knih, které chceme zpracovat (viz sekci 4.1), obsahuje *<a-značky>*, nikoli *<b-značky>*.

Ve třetím parametru za `\BookTitle` jsou názvy knih ve složených závorkách.

Soubor s 66 údaji `\BookTitle` je vygenerován automaticky po extrakci základních textů z *Swordu* pomocí `mod2tex` a `maketxs` (viz sekci 4.4). Je možné jej použít, ovšem *<a-značky>* a tituly knih je potřeba asi manuálně upravit podle zvyklostí daného překladu, jak je předvedeno v ukázce výše.

Do souboru s názvy knih je možné přidat další informace o jednotlivých knihách pomocí příkazů `\BookException`, `\BookPre`, `\BookPost`. Mají tuto syntaxi:

```
\BookException <a-mark> {\<text-výjimky>}
\BookPre       <a-mark> {\<text-před-knihou>}
\BookPost      <a-mark> {\<text-po-knize>}
```

Přitom *<text-výjimky>* se vloží před čtením všech souborů knihy vymezené značkou *<a-mark>* uvnitř cyklu pro čtení všech knih příkazem `\processbooks`, dále *<text-před-knihou>* se vloží po načtení souborů knihy, ale před zpracováním prvního verše. Konečně *<text-po-knize>* se vloží za posledním vešem knihy.

Příklad použití `\BookException`: pokud chcete mít v názvech souborů *<a-značky>* a ty obsahují diakritická znaménka, je možné upravit hodnotu makra `\amark` tak, aby neobsahovalo diakritiku takto:

```
\BookException Ž    {\def\amark{Z}}
\BookException Př    {\def\amark{Pr}}
\BookException Pís    {\def\amark{Pis}}
\BookException Ř     {\def\amark{R}}
\BookException Žd     {\def\amark{Zd}}
```

Pak třeba soubory `notes-\amark.tex` jsou ve skutečnosti pojmenovány `notes-Z.tex`, `notes-Pr.tex` atd. Přitom v textu poznámek dále normálně používáte značky knih `Ž`, `Př`, `Pís` atd.

V Bibli je pět knih, které mají jen jednu kapitolu (Abdijáš, Filemon, 2 a 3 list Janův a Judův). Vzhledem k tomu, že odkazy do nich se nepíší s číslem kapitoly (Fm 1:4), nýbrž jen číslem verše

(Fm 4), musíme \TeX naučit, které to jsou, aby u nich číslo kapitoly neočekával, ale přitom aby věděl, že je to číslo 1, které se nikam nepíše. Při odkazování na takové knihy se pak použije jiný způsob interpretace odkazu, viz sekci 7.3. Dosáhneme toho definováním makra `\nochapbooks`, které musí obsahovat *<a-značky>* těchto knih: `\def\nochapbooks {Abd Fm 2Jn 3Jn Ju}`, samozřejmě totožné s těmi, které jsou již uvedeny v definicích `\BookTitle` (v souboru `Cze-books.tex`).

4.4 Formát základního textu, tzv. .txs soubory

Předpokládá se, že základní text Bible je uložen v souborech s příponou `.txs` (text source). Každá z 66 knih Bible je uložena ve svém `.txs` souboru. Názvy `.txs` souborů a jejich umístění musejí odpovídat deklaraci `\def\txsfile` v hlavním souboru `main.tex` (viz sekci 4.1).

Každý řádek `.txs` souboru obsahuje jeden verš Bible uvozený `#<číslo-kapitoly>:<číslo-verše>`. Verše musejí být uvedeny ve správném pořadí. Například začátek souboru `CzeBKR-Da.tx` vypadá takto (části textu jsou v ukázce vynechány):

```
#1:1 Léta třetího kralování Joakima krále Judského, ... a oblehl jej.
#1:2 I vydal Pán v ruku jeho Joakima ... do domu pokladu boha svého.
...
```

Základní texty Bible lze získat například z modulů Swordu <https://www.crosswire.org/sword/modules/ModDisp.jsp?modType=Bibles>. Jednotlivé `.txs` soubory je pak možné vygenerovat následujícím postupem (na Linuxu):

ZIP stažený z uvedené [www](https://www.crosswire.org/sword/modules/ModDisp.jsp?modType=Bibles) stránky (tzv. modul) někam rozbalte. Je třeba mít dále v počítači instalován balíček `libsword-dev` a dále program `mod2tex`, který je součástí `OpBible`. Pomocí `installmgr -l` zjistíte seznam stažených modulů. Máte-li nastaven aktuální adresář v místě, kam jste rozbalili ZIPy, a vznikl tam adresář `modules`, pak jsou takové moduly nalezeny. Moduly obsahují texty v binárním formátu, my je potřebujeme převést do textového formátu. K tomu stačí napsat do příkazového řádku:

```
mod2tex modul > soubor
```

kde `modul` je název modulu. Ve výsledném souboru máte kompletní základní text daného překladu (modulu). Například po

```
mod2tex CzeBKR > CzeBKR.out
```

je v souboru `CzeBKR.out` kompletní překlad Bible kralické. Ten nyní můžete rozdělit do `.txs` souborů příkazem

```
maketxs CzeBKR.out
```

Uvedený příkaz kromě 66 `.txs` souborů vytvoří soubor `CzeBKR-books.tex`, ve kterém jsou tituly a zkratky jednotlivých knih, tedy je tam:

```
\BookTitle Gen Gen {Genesis}
\BookTitle Exod Exod {Exodus}
\BookTitle Lev Lev {Leviticus}
\BookTitle Num Num {Numbers}
\BookTitle Deut Deut {Deuteronomy}
\BookTitle Josh Josh {Joshua}
\BookTitle Judg Judg {Judges}
...
```

Tyto tituly nejsou v češtině, protože je zdroj Sword neobsahuje. Je tedy třeba tento soubor manuálně upravit a vložit místo anglických české názvy. Zkratky knih jsou uvedeny dvakrát stejně. První pozměňte podle zvyklostí používání zkratk v českých Biblích a druhou zkratku můžete ponechat. Pak deklarace

```
\def\txsfile {sources/Cze\mark-\bmark.tx}
```

v hlavním souboru způsobí, že se vytvořené `.txs` soubory budou hledat v adresáři `sources/` a předpokládají se jejich názvy `CzeBKR-Gen.tx`, `CzeBKR-Exod.tx`, `CzeBKR-Lev.tx`, atd.

Pokud by měl nastat ten vzácný případ, že byste skládali Bibli z několika různých zdrojů, řekněme Starý Zákon byste chtěli v překladu Dr. Jana Hejčla a Nový Zákon v překladu Františka Žilky,

budete muset trochu žonglovat s názvy souborů, aby výsledná definice `\tmark` byla stejná pro celou Bibli. Oba *.out soubory (např. HEJCL.out a ZILKA.out)¹ by bylo nutno spojit v jeden a ten pak pojmenovat třeba CzeHecjZilka.out a pak teprve provést `\maketxs CzeHecjZilka.out`. Pak lze mít v hlavním souboru `main.tex` definici `\def\tmark{HecjZilka}` a výsledná studijní Bible bude mít Hejčlův Starý Zákon a Žilkův Nový.

Se soubory .txs nadále nebudete potřebovat nijak manipulovat, není potřebné je editovat ani doplňovat. V adresáři `sources/` tedy můžete mít „datový sklad“ všech základních textů Bible pro všechny použité varianty překlady najednou. V případě šesti variant překladu tam tedy máte $6 \times 66 = 396$ souborů.

Máte-li v .txs souborech něco, co chcete nakonec formátovat jinak, je možné v hlavním souboru použít `\cnvtext{<co>}{<jak>}`. TeX v každém verši z .txs souboru vyhledá všechny výskyty `<co>` a nahradí je `<jak>`. Například máte-li v .txs souborech úseky textu v hranatých závorkách, tedy `[něco takového]` a chcete je tisknout kuzívou, do hlavního souboru napište:

```
\cnvtext{[ ]{\bgroup\it} \cnvtext{ ]}{\egroup}}
```

Může se stát, že .txs soubor nepoužívá správné typografické uvozovky (tj. v češtině „...“), ale místo nich tam jsou programátorské uvozovky "...". Bez zásahu do .txs souboru je možné toto opravit přidáním pokynu do hlavního souboru:

```
\quotationmarks{„ }{“}
```

Pak se budou automaticky nahrazovat programátorské uvozovky v .txs souboru českými typografickými. Podobně můžete v hlavním souboru deklarovat nahrazování anglickými uvozovkami či jinými, například deklarací `\quotationmarks{" "}`. Programátorská uvozovka je pak implicitně nahrazena otevírací typografickou (první parametr v deklaraci), ale pokud za ní následuje mezera, konec verše, konec odstavce, tečka nebo čárka, je nahrazena zavírací typografickou uvozovkou.

4.5 Údaje určující formátování základního textu v souborech `fmt-*.tex`

Základní text v .txs souborech neobsahuje žádné formátovací ani doplňkové údaje, jakými jsou například názvy kapitol nebo místa, kde se má ukončit odstavec či přejít z formátování v bloku do formátování na střed řádků a zpět.

Protože do základního textu nechceme nijak zasahovat,² je třeba tyto doplňkové údaje deklarovat k příslušným veršům pomocí speciálních příkazů `\fmtadd`, `\fmtpre` a `\fmtins`. Tyto příkazy jsou typicky v souborech `fmt-***.tex`, například `fmt-CzeBKR-Da.tex`. Je vhodné udržovat tyto formátovací soubory závislé jednak na knize (Daniel v uvedeném příkladu), ale také na použitém překladu (Bible kralická v příkladu). Je sice možné začít s jedním souborem pro každou knihu a soubory pro ostatní překlady pořídit jako kopie výchozích, ale nakonec bude asi potřebné pro rozličné varianty základního textu formátovací pokyny mírně modifikovat v souladu s použitým překladem.

Syntaxe použití uvedených příkazů je následující:

```
\fmtpre{<číslo-kapitoly>: <číslo-verše>}{<fmt-pokyn>}
\fmtadd{<číslo-kapitoly>: <číslo-verše>}{<fmt-pokyn>}
\fmtins{<číslo-kapitoly>: <číslo-verše>}{<fráze>}{<fmt-pokyn>}
```

kde `<fmt-pokyn>` je „vzkaz“, který je předán TeXu k formátování. Například `\endgraf` značí konec odstavce. `\begcenter` otvírá pasáž s centrováním textem a ta musí být (někde později) uzavřena pomocí `\endcenter`. Nebo `\chaptit{<text>}` vloží `<text>` jako titul kapitoly, kdežto `\schaptit{<text>}` vloží nadpis perikopy jinak, než před první verš kapitoly a udělá nad a pod nadpisem patřičné místo.

Příkaz `\fmtpre` vkládá `<fmt-pokyn>` na začátek uvedeného verše (ještě před případně vytištěné číslo verše horním indexem). Příkaz `\fmtadd` vkládá `<fmt-pokyn>` na konec uvedeného verše. Konečně `\fmtins` vkládá `<fmt-pokyn>` dovnitř verše za první výskyt stanovené `<fráze>`, která ve verši musí doslovně existovat. Jinak TeX vypíše varování a `<fmt-pokyn>` nezařadí vůbec.

Jak může vypadat použití příkazů `\fmt*` je vidět například v souboru `fmt-BKR-Da.tex`.

Příkaz `\fmtfont{<číslo-kapitoly>: <číslo-verše>}{<fráze>}{}` slouží k označení vybrané fráze zvoleným fontem. Například `\fmtfont{1:26}{lidé}{\em}` v knize Genesis způsobí, že slovo

¹ Jejich moduly nejsou na Swordu, *.out soubory by musely vzniknout stažením např. z <https://obohu.cz> a nějakým šikovným skriptem (nebo ručně?) převedením do požadované podoby.

² Řekněte si, že to jsou svatá Písma, tudíž „nedotknutelný“ text. Jediné, co Vás oprávní do základního textu zasáhnout, je ta málo pravděpodobná možnost, že byste ve zdroji ze Swordu (nebo odkud jste ho získali) objevili chybu vůči tištěné verzi. Tu pak bude skutečně lepší odstranit na místě.

„lidé“ bude vytištěno kurzívou, protože `\em` je inteligentní přepínač do kurzívy (automaticky doplňuje za slovo italickou korekci `\/`, o kterou se nemusíte starat, což byste museli, kdybyste do kurzívy přepínali obyčejným `\it`). Namísto `\em` je možno použít jakýkoli jiný přepínač fontu.

Mimo `\begcenter` a `\endcenter` je také možné použít řízené odsazování pomocí `\ind<číslo>` (jako indent). V místě vložení se ukončí řádek a další řádek začíná odsazen o `<číslo>` odstavcových zarážek. Vkládání takových `\ind` příkazů prostřednictvím `\fmtins` nebo `\fmtpre` může být ale docela pracné a nepřehledné, přitom Bible se zhusta hemží poetickými pasážemi, které vyžadují spousty různě odsazených řádků. K tomu účelu může sloužit příkaz `\fmtpoetry`, jehož použití nejprve ukážeme na příkladě pro překlad CEP knihy Jeremiáš 23:9–15:

⁷ „Hle, přicházejí dny, je výrok Hospodinův, kdy se už nebude říkat: „Jakože živ je Hospodin, který vyvedl syny Izraele z egyptské země“, ⁸ nýbrž: „Jakože živ je Hospodin, který vyvedl a přivedl potomstvo domu Izraelova ze severní země a ze všech zemí, kam je rozehnal“. Usadí se ve své zemi.“

⁹ O prorocích:

„Mé srdce je zlomeno v mém nitru,
všechny mé kosti se chvějí,
jsem jako opilý člověk,
jako muž zmožený vínem,
kvůli Hospodinu, kvůli jeho svatým slovům.

¹⁰ Poněvadž země je plná cizoložníků,
truchlí pod kletbou
a pastviny na stepi vyschly;
oni však běhají za zlem
a zmužile si vedou v tom, co není správné.

¹¹ Vždyť jak prorok, tak kněz se rouhají,
i ve svém domě nalézám jejich zlé činy,
je výrok Hospodinův.

¹² Proto bude jejich cesta kluzká,
budou vyhnáni do temnoty
a v ní padnou:
přivedu na ně zlo,
rok jejich trestu,
je výrok Hospodinův.“

¹³ Na samařských prorocích jsem viděl tuto nepatřičnost:
Prorokovali ve jménu Baalově
a sváděli Izraele, můj lid.

¹⁴ Také u jeruzalémských proroků jsem viděl hroznou věc:
cizoložství a neustálé klamání.
Posilují ruce zlovolníků,
aby se nikdo neodvrátil od svých zlých činů.
Jsou pro mne všichni jako Sodoma
a obyvatelé města jako Gomora.

¹⁵ Proto Hospodin zástupů praví proti těm prorokům toto: „Hle, nakrmím je pelyňkem a napojím je otrá-

Poetická část této ukázky byla vysázena takto:

```
\fmtpre{23:9}{\endgraf}
\fmtpoetry{23:9}{prorocích:// nitru,/// chvějí,/// člověk,/// vínem,///}
\fmtpoetry{23:10}{// cizoložníků,/// kletbou// vyschly;/// zlem///}
\fmtpoetry{23:11}{// rouhají,/// činy,///}
\fmtpoetry{23:12}{// kluzká,/// temnoty// padnou;/// zlo,/// trestu,///}
\fmtpoetry{23:13}{// nepatřičnost:/// Baalově///}
\fmtpoetry{23:14}{// věc:/// klamání.// zlovolníků,/// činů./// Sodoma///}
\fmtins{23:9}{prorocích:}{\bigskip}
\fmtpre{23:13}{\medskip}
\fmtpre{23:14}{\medskip}
\fmtpre{23:15}{\bigskip}
```

`\fmtpoetry{<číslo-kapitoly>:<číslo-verše>}{<formátovací-údaje>}` stanoví formátování konkrétně uvedeného verše. Ve `<formátovacích-údajích>` se vyskytují slova z konců řádků následovaná jedním nebo více lomítky. Počet lomítek udává, o kolik odstavcových zarážek bude odsazen další řádek za uvedeným slovem. `<formátovací-údaje>` musejí nutně skončit jedním nebo více lomítky a mohou (ale nemusí) začínat jedním nebo více lomítky. Pokud začínají, je i začátek verše odsazen o příslušný počet odstavcových zarážek, ale číslo verše je vysazeno vystrčené mírně doleva do prostoru odstavcové zarážky.

Chceme-li při použití `\fmtpoetry` navíc vložit vertikální mezery, je to možné, ale až po příkazu `\fmtpoetry`, jak je vidět i v ukázce, kde je vložena mezera polovičního řádku `\medskip` a mezera celého řádku `\bigskip`. Platí totiž pravidlo, že pokud na stejné místo pomocí `\fmtins` nebo `\fmtpre` vložíte více formátovacích pokynů, ty se nakonec provedou v opačném pořadí, než byly do `.fmt` souboru vloženy. Takže například u verše 23:13 výše uvedené ukázky se nejprve provede na začátku verše `\medskip` a pak se následující řádek odsadí o dvě odstavcové zarážky podle `\fmtpoetry` (`\fmtpoetry{23:13}{// nepatřičnost:...}`).

4.6 Poznámky v souborech notes-*.tex

Tyto soubory (umístěné podle deklarace `\notesfile` v hlavním souboru) obsahují mimo jiné poznámky k jednotlivým veršům nebo částem veršů. K psaní poznámek slouží zejména příkaz `\Note`. Protože toto je hlavní důvod, proč byl vytvořen balíček OpBible, je tomu věnována celá sekce 5.1.

5 Poznámky a další objekty s vazbou na základní text

5.1 Poznámky vázané na frázi základního textu: příkaz `\Note`

Hlavním smyslem nástroje OpBible je vytvořit PDF ze základního textu nejen se samotným textem Bible, ale s propojenými poznámkami. K tomu slouží mimo jiné soubory notes-*.tex (například notes-Gn.tex pro knihu Genesis), ve kterých jsou zapsány poznámky k jednotlivým frázím základního textu dle následujícího smlouveného formátu. Jednotlivá poznámka je uvozena příkazem `\Note` a má tvar:

```
\Note <číslo-kapitoly>:<číslo-verše> {<fráze>} <text-poznámky>
<prázdný řádek>
```

Například:

```
\Note 1:2 {nádobí} Odkaz na nádobí z vypleněného chrámu, nikoliv na deportaci zajatců.
```

Příklad je ze souboru notes-Da.tex, tedy ze souboru s poznámkami ke knize Daniel. Konkrétně se poznámka váže na kapitolu první, verš druhý a na frázi „nádobí“. Tato fráze musí být bezpodmínečně ve stanoveném verši základního textu obsažena. Pak ji \TeX propojí s odpovídajícím místem v základním textu, tj. zajistí, že výskyt fráze i samotné poznámky je na společné stránce. Jinými slovy, stránky se nelámou podle čísla verše a jemu odpovídajícího čísla poznámky; lámou se podle komentované fráze. Když stránkový zlom vyjde uvnitř verše, v němž komentujeme více než jednu frázi, poznámky následují své fráze, nejen začátek verše. V ukázkové knize Daniel je to vidět v překladech CEP a B21 na verši 9:27 – jedna fráze na jedné straně, druhá na druhé, a stejně tak i poznámky k nim.

Povšimněme si, že fráze se váže na překlad Bible kralické, zatímco v překladu ekumenickém je použit ve stejném verši termín „nádob“. Jak zařídit, aby se stejná poznámka automaticky mírně modifikovala podle použitého překladu, je popsáno v sekci 6.2.

Pokud `<fráze>` není ve stanoveném verši základního textu doslovně obsažena, \TeX během zpracování oznámí do logu a na terminál o tom varování a poznámku zařadí k danému verši, jakoby `<fráze>` byla na začátku verše.

Vytištěná poznámka (v závislosti na typografickém nastavení) obsahuje například zopakování čísla kapitoly a verše, pak následuje komentovaná fráze (například tučně) a za ní je vlastní text poznámky.

Někdy je třeba v základním textu vyhledat mírně jinou frázi, než jakou pak chceme zapsat do vytištěné poznámky (např. v základním textu je slovo v jiném pádě nebo je to mírně jinak formulovaná fráze). Pak je možné těsně za `{<fráze>}` napsat rovnítko následované `{<fráze-k-tisku>}`, tedy

```
\Note <číslo-kapitoly>:<číslo-verše> {<fráze>}={<fráze k tisku>} <text-poznámky>
<prázdný řádek>
```

V takovém případě se v základním textu hledá `<fráze>`, ale ve vlastní poznámce se vytiskne jako heslo poznámky `<fráze-k-tisku>`. Například:

```
\Note 1:2 {vydal Pán}={Pán vydal} Porážka Izraele Babylónem není vysvětlitelná
jen pouhou vojenskou a politickou analýzou oné doby. Bůh vždy jednal svrchovaně
v záležitostech národů. Babylóňany použil jako nástroj potrestání svého vlastního
lidu za porušení smluvních závazků.
```

V tomto příkladě se hledá ve verši druhém základního textu „vydal Pán“ ale v poznámce je vytištěno heslo „Pán vydal“.

Povšimněte si, že tato poznámka se rovněž váže na verš druhý kapitoly první knihy Daniel (protože je v souboru notes-Da.tex). Je tedy možné mít více poznámek k různým frázím, které se váží na stejný biblický verš.

Jednotlivé poznámky ve zdrojovém souboru poznámek jsou odděleny prázdnými řádky. To je nutné, jinak by \TeX při jejich čtení nepoznal, kde končí text poznámky. Také se tím zvyšuje přehlednost zdrojového souboru. Další řádky poznámky můžete (ale nemusíte) odsadit.

Pokud se poznámka váže k celému verši (tj. bez specifikované fráze), pište {}, tedy prázdnou frází k vyhledání. Například:

```
\Note 1:1-21 {}={Uchování rituální čistoty} Prorok uvozuje kontext své knihy
    vyprávěním osobní historie (své i svých přátel) zajetí, vzdělání, věrnosti Bohu
    a služby králi Nabuchodonozorovi.
```

Tato ukázka navíc demonstrovuje, že je také možné uvést celý rozsah čísel veršů. Je-li *<fráze>* pro vyhledání prázdná (jako v tomto příkladě), pak se poznámka umístí na stejnou stranu, kde je začátek prvního verše z uvedeného rozsahu. Je-li *<fráze>* neprázdná, musí se vyskytovat v prvním verši z uvedeného rozsahu veršů. Rozsah veršů bude stejně vytištěn v úvodu poznámky. Symbol pro rozsah „–“ je jediný znak „mínus“ běžně dosažitelný na klávesnici. Má kód 45. Nesmí to být žádný speciální znak vypadající třeba jen podobně jako vodorovná čárka.

Pořadí vytištěných poznámek k jednomu verši odpovídá pořadí frází, na které se vážou, v základním textu. Na pořadí, ve kterém jsou zapsány ve zdrojovém souboru to tedy nemá vliv. Poznámky, které se vážou pomocí prázdného parametru {} k celému verši, jsou řazeny jako první a je-li jich více, pak jsou řazeny podle pořadí ve zdrojovém souboru.

5.2 Příkazy pro vložení dalších objektů

Do souboru `notes-*.tex` je možné kromě příkazů `\Note` zapisovat pokyny pro vložení dalších objektů s vazbou na konkrétní verš v základním textu. Takový objekt se umístí³ na spodní část strany pod dva sloupce poznámek nebo citáty se umístí na horní část strany. Vazbu na místo textu vymezíme pomocí údaje *<číslo-kapitoly>*:*<číslo-verše>* podobně jako v případě poznámek `\Note`. Objekt se umístí na stejnou stranu, kde začíná příslušný verš. Pokud se na stránku pod verš objekt nevejde, umístí se na spodní část následující stránky.

Obrázky vložíte pomocí `\putImage` (sekce 5.3), články pomocí `\putArticle` (sekce 5.4), citace pomocí `\putCute` (sekce 5.5), obrázky přes dvě stránky v otevřené knize pomocí `\putSpanImage` nebo `\putSpanText` (sekce 9.1).

Pořadí, v jakém do souboru `notes-*.tex` příkazy na vození těchto objektů napíšete, nemá na konečný vzhled po vytištění žádný vliv. Můžete mít všechny objekty jednoho typu soustředěny na jednom místě zdrojového souboru, ale taky je můžete mít umístěny mezi poznámkami `\Notes` typicky podle čísla verše, ke kterému jsou objekty vázány.

5.3 Vložení obrázků

Obrázky s vazbou na konkrétní verš umístíme příkazem `\putImage`:

```
\putImage <číslo-kapitoly>:<číslo-verše> {<titulek>} [<lejblík>] (<parametry>) {<soubor>}
```

Například:

```
\putImage 2:1 {Danielovy vize} [danielovyvize] () {Nabuco.pdf}
```

- *<číslo-kapitoly>*:*<číslo-verše>* označuje místo, ke kterému se obrázek umístí.
- *<titulek>* je titulek (popisek) k obrázku. V implicitním typografickém nastavení se vytiskne do proužku, který odděluje horní část strany od samotného obrázku.
- *<lejblík>* udává lejblík k obrázku pro případné reference. Můžete pak na něj odkazovat pomocí ...viz obrázek `\ref [<lejblík>]` na straně `\pg`, což vytiskne v případě naší ukázky ...viz obrázek [Danielovy vize](#) na straně 123. Přitom text v tomto odkazu bude propojen hyperlinkem s místem, kde se vyskytuje obrázek. Když nechcete *<lejblík>* použít, pište [] (tj. jakoby prázdný lejblík).
- *<parametry>* určují případný další pokyn pro formátování obrázku. Mohou chybět, jak naznačuje ukázka. Implicitně se obrázek roztáhne na celou šířku stránky. Pokud ho chcete mít třeba menší, pište do místa pro *<parametry>* třeba `\picw=9cm` což způsobí, že obrázek bude mít šířku 9 cm a bude umístěn na střed.
- *<soubor>* je úplné jméno souboru s obrázkem. Může mít příponu pdf (pro vektorové obrázky) nebo png či jpg pro bitmapové obrázky. Soubory s obrázky musejí být uloženy v adresáři `images/`. Jsou-li umístěny jinde, je třeba nastavit pro toto umístění parametr `\picdir`, například `\picdir={bibleImages/}`. Toto je rozumné nastavit v hlavním souboru, viz sekci 4.1.

³ Podle implicitní definice vzhledu jednotlivých stran, to je možné předefinovat.

5.4 Vložení článků

Článek s textem (typicky delším než poznámka) lze umístit podobně jako obrázky, tj. při implicitním typografickém nastavení na spodní stranu stránky, na níž začíná verš specifikovaný pomocí `<číslo-kapitoly>: <číslo-verše>`. Nevejde-li se tam, vloží se na spodní část následující strany. Nevejde-li se na jedinou stranu, vloží se jeho další úsek na spodní část následující strany (atd. až do chvíle, kdy je celý článek do textu vložen).

Text článků ke každé knize Bible zapisujete do souboru nazvaném a uloženém podle makra `\articlefile`, například v souboru `articles-Gn.tex` jsou všechny články pro knihu Genesis. Jak obsah tohoto souboru vypadá, popíšeme za chvíli.

V souboru `notes-*.tex` je třeba uvést požadavek na umístění článku pomocí `\putArticle`:

```
\putArticle <číslo-kapitoly>: <číslo-verše> {<titulek>} [<číslo-článku>] (<parametry>)
```

Například:

```
\putArticle 6:1 {Kdo byl Darius Médský?} [6] ()
```

- `<číslo-kapitoly>: <číslo-verše>` označuje místo, ke kterému se obrázek umístí.
- `<titulek>` je titulek ke článku. Vytiskne se podobně jako v případě obrázku.
- `<číslo-článku>` je nějaké číselné označení článku, které musí být v rámci jedné knihy jednoznačné. Typicky se články vážou na kapitoly, pak by `<číslo-článku>` mělo znamenat číslo odpovídající kapitoly. Podle tohoto čísla se článek do sazby neumísťuje (k tomu slouží údaj `<číslo-kapitoly>: <číslo-verše>`), nicméně je toto číslo použité v odkazech. Můžete například psát ...viz <"článek" Da 6>a a vytiskne se ...viz [článek Da 6](#). Formát a vlastnosti takových odkazů popisuje sekce 7. Navíc se podle `<číslo-článku>` vyhledá vlastní text článku v příslušném souboru `articles-*.tex`.
- `<parametry>` jsou parametry přesněji určující formátování článku. **TODO...**

Text článků ke vložení musí být uveden v souboru `articles-*.tex`. Tento soubor musí obsahovat řádek uvozený `\Article` takto:

```
\Article [<číslo-článku>]
```

Za ním následuje text článku. Pak může pokračovat další `\Article [<číslo-článku>]` následovaný další textem atd. V jednom souboru jsou tak shomážděny všechny články k dané knize.

Pokud zadáte požadavek `\putArticle` a příslušný soubor `article-*.tex` nebo příslušný řádek v něm `\Article [<číslo-článku>]` neexistuje, zpracování \TeX em skončí s chybou.

5.5 Vložení citací do horní části stránky

Příkaz `\putCite <číslo-kapitoly>: <číslo-verše> {<text>}` vloží při implicitním typografickém nastavení `<text>` ve formě citátu na začátek stránky, na které se nachází `<číslo-kapitoly>: <číslo-verše>`.

Samotný `<text>` může obsahovat na konci `\quotedby {<autor>}`. V takovém případě je vytištěn `<autor>` na novém řádku (při citátu umístěném na levé stránce) nebo za předpokladu, že je dost místa, je na posledním řádku posunutý více vpravo (při citátu umístěném na levé stránce).

Pro řízený přechod na nový řádek můžete v `<textu>` použít příkaz `\nl` (new line).

5.6 Vložení citací do okraje v textu článku

Text článků je v implicitním nastavení sazby formátován do dvou sloupců. Vnější sloupec je možné přerušit a do vzniklého místa vsunout např. citát či jinou myšlenku. Tento text bude vyčuhovat do vnějšího okraje.

Uvnitř článku (tedy těsně po příkazu `\Article [<číslo-článku>]`) je třeba vložit deklaraci citovaného textu pomocí: `\Cite <písmeno> {<text>}`. Zde `<písmeno>` je typicky A. Pokud ale chcete do jednoho článku vložit více citátů, je třeba je odlišit dalšími písmeny, tedy B, C, atd. a všechny citáty napište na začátek článku.

Příkazem `\Cite` je citát pouze deklarován. Jeho skutečné vložení do textu článku provedete pomocí `\insertCite <písmeno>\left` a současně na jiném místě v textu článku použijte `\insertCite <písmeno>\right`. Je-li článek na levé stránce dvoustrany, citát se umístí jen podle `\insertCite <písmeno>\left`, jinými slovy `\insertCite <písmeno>\right` je ignorováno. Je-li článek na pravé stránce otevřené dvoustrany, umístění se řídí jen podle `\insertCite <písmeno>\right`.

Hned pod řádkem, ve kterém je příslušné `\insertCite` uvedeno, se citát skutečně objeví. Řádek samotný se kvůli tomu nerozdělí (jinak řečeno, odstavec se kvůli vložení citátu neukončí).

Důvod, proč je nutné uvést dvě lokace `\insertCite` je následující. Dopředu nevíme, na které stránce (liché nebo sudé) se článek s vloženým citátem objeví. Protože citát má zasahovat do vnějšího okraje, je třeba jej na levé stránce umístit do prvního sloupce, na pravé stránce do druhého sloupce, tedy někde poněkud jinak. Lokace `\insertCite <písmeno>\left` by tedy měla odpovídat nějakému řádku v prvním sloupci a `\insertCite <písmeno>\right` jinému řádku v druhém sloupci článku.

Je rozumné si odladit (tj. dopředu vyzkoušet), jak vypadá umístění citátu pro obě varianty, které mohou nastat (vlevo/vpravo). Pokud se chcete podívat, jak to dopadne při variantě, která neodpovídá správné poloze stránky, můžete použít na začátku textu článku příkaz `\swapCites`, který jejich pozice vzájemně prohodí. Toto by ale nemělo zůstat zapnuté pro definitivní tisk, proto aktivované `\swapCites` způsobí varování na terminál a do log souboru.

6 Různé (ale podobné) verze základního textu

Úvod do této problematiky je v sekci 4.2. Zde je k jednotlivým možnostem podrobná dokumentace.

6.1 Deklarace variant překladu a použití pomocí příkazu `\x`

Nejsou-li varianty deklarovány příkazem `\vdef`, pak příkaz `\x/<fráze>/` použitý v textu vypíše `<fráze>`. Je ovšem možné deklarovat varianty překladu. Počet variant je nutné stanovit příkazem `\variants` (viz sekci 4.2). To se provede jedinkrát v celém dokumentu. Pak mohou následovat příkazy `\vdef` vždy s tolika parametry, kolik variant bylo specifikováno příkazem `\variants`. Například

```
\variants 6 {BKR}      {PSP}      {CSP}      {CEP}      {B21}      {SNC}
\vdef      {Baltazar} {Béltešaccar} {Beltšasar} {Beltšasar} {Baltazar} {Beltšasar}
\vdef      {Darius}  {Dárjáveš}   {Dareios}  {Darjaveš}  {Darjaveš}  {Darjaveš}
\vdef      {Daria}   {Dárjáveše}  {Dareia}   {Darjaveše} {Darjaveše} {Darjaveše}
\vdef      {Daniel}  {Dánijjél}   {Daniel}   {Daniel}   {Daniel}   {Daniel}
\vdef      {sedm let} {sedm dob}   {sedm časů} {sedm let}  {sedm období} {po dobu sedmi let}
```

Pokud je nyní v hlavním souboru deklarováno `\def\tmark{<varianta>}`, pak se `\x/<fráze>/` promění na `<frázi>` specifikované `<varianty>`. Přitom parametr `<fráze>` příkazu `\x` musí být shodný s první frází uvedenou v příkazu `\vdef`. Je-li v našem případě v hlavním souboru `\def\tmark{BKR}`, pak

```
\x/Baltazar/      vypíše Baltazar
\x/Darius/         vypíše Darius
\x/sedm let/       vypíše sedm let
\x/cokoli/         vypíše cokoli
```

Je-li ale v hlavním souboru `\def\tmark{CEP}`, pak

```
\x/Baltazar/      vypíše Beltšasar
\x/Darius/         vypíše Darjaveš
\x/sedm let/       vypíše sedm let
\x/cokoli/         vypíše cokoli a na terminálu bude varování,
                    že fráze /cokoli/ nemá deklarován překlad.
```

Například poznámka v souboru `notes-Da.tex` (pro knihu Daniel) by mohla znít:

```
\Note 5:31 {Darius}={\x/Darius/ Médský} Některé školy tvrdí, že tento a další
(<6:1>, <6:6>, <6:9>, <6:25>, <6:28>; <9:1>; <11:1>)
odkazy na \x/Daria/ Médského v knize \x/Daniel/ jsou historické omyly.
```

a tato poznámka se při `\def\tmark{CEP}` vypíše ve znění:

5:31 Darjavéš Médský Některé školy tvrdí, že tento a další (6:1, 6:6, 6:9, 6:25, 6:28; 9:1; 11:1) odkazy na Darjavéše Médského v knize Daniel jsou historické omyly.

Vybrané parametry příkazu `\vdef` mohou být prázdné (zapsané jako `{}`), což značí nedefinovanou frázi pro daný jazyk. Bude-li potřeba takovou frázi použít pomocí `\x/. . . /`, vypíše se varování. Dále parametr může obsahovat jediný znak `"`, což znamená, že se použije stejná fráze, jako v předchozím parametru. Takže naše výše uvedená ukázka může také vypadat takto:

<code>\vdef</code>	<code>{Baltazar}</code>	<code>{Béltešaccar}</code>	<code>{Beltšasar}</code>	<code>{"}</code>	<code>{Baltazar}</code>	<code>{Beltšasar}</code>
<code>\vdef</code>	<code>{Darius}</code>	<code>{Dárjáveš}</code>	<code>{Dareios}</code>	<code>{Darjaveš}</code>	<code>{"}</code>	<code>{"}</code>
<code>\vdef</code>	<code>{Daria}</code>	<code>{Dárjáveše}</code>	<code>{Dareia}</code>	<code>{Darjaveše}</code>	<code>{"}</code>	<code>{"}</code>
<code>\vdef</code>	<code>{Daniel}</code>	<code>{Dánijjél}</code>	<code>{Daniel}</code>	<code>{"}</code>	<code>{"}</code>	<code>{"}</code>
<code>\vdef</code>	<code>{sedm let}</code>	<code>{sedm dob}</code>	<code>{sedm časů}</code>	<code>{sedm let}</code>	<code>{sedm období}</code>	<code>{po dobu sedmi let}</code>

Na co si dát pozor: Uvozovky na místě fráze nelze použít v definicích \ww (viz kapitola 6.2), které předcházejí \Note. Fungují pouze v definici \vdef v souboru s variantami, platnými pro celou Bibli.

6.2 Variantní deklarace frází pro spárování poznámek s textem

Před příkazem `\Note` může bezprostředně předcházet deklarace vyhledávaného slova podle variant překladu pomocí `\ww` (to je zkratka za watchword). Příkaz `\ww` má tolik parametrů, kolik je variant překladu deklarovaných příkazem `\variants` a tyto parametry mohou být jednoduché (ve formátu `{\hledaná-fráze}`) nebo složené (ve formátu `{\hledaná-fráze}={\co-vytisknout}`). Bezprostředně následující `\Note` pak bude ignorovat svůj parametr pro hledanou frázi a použije parametr z `\ww` odpovídající zpracovávané variantě jazyka. Například:

```

\ww {liternímu umění a jazyku} %BKR
{písmu a jazyku Kasdím} %PSP
{chaldejskému písemnictví a jazyku} %CSP
{kaldejskému písemnictví a jazyku} %CEP
{babylonskému písemnictví a jazyku} %B21
{kaldejský jazyk}={kaldejský jazyk ... s národním písemnictvím} %SNC
\Note 1:4 {} Babylónská literatura byla ...

```

Zde uvedená poznámka Note 1:4 vyhledá ve verši 1:4 text „liternímu umění a jazyku“, když je zpracovávána varianta překladu BKR, a vyhledá text „kaldejskému písmenictví a jazyku“ při zpracování varianty překladu CEP. Ukázka předpokládá, že bylo pomocí příkazu \variants deklarováno šest variant překladu ve stanoveném pořadí. Můžete též stanovit rozdílnou frázi pro vyhledání a pro vtištění v poznámce, jak ukazuje následující příklad:

```

\ww {Baltazar} %BKR
{Béltešaccar}={Béltešaccar ... Šadrach ... Měšach ...i Avéd-negó} %PSP
{Beltšasar}={Beltšasar ... Šadrak ... Měšak ...i Abed-nego} %CSP
{Beltšasar}={Beltšasar, Chananjášovi Šadrak, Mišaelovi Měšak a Azarjášovi Abed-nego}
{Baltazar}={Baltazar, Chananiášovi Šadrach, Mišaelovi Mešach a Azariášovi Abednego}
{Beltšasarem}={Beltšasarem, Chananjáše Šadrakem, Mišaele Měšakem a Azarjáše Abed-negem}

\Note 1:7 {} Přesný význam těchto jmen je předmětem diskusí...

```

Fráze pro hledání a nahrazení se použijí přesně tak, jak jsou napsány v parametrech příkazu `\ww`. To neplatí pro poznámky `\Note`, které nemají před sebou `\ww`. Pak při použití

\Note *⟨verš⟩* {*⟨hledaná-fráze⟩*} *⟨text⟩* *⟨prázdný řádek⟩*

nebo

`\Note <verš> {<hledaná-fráze>}={<co-vytisknout>} <text> <prázdný řádek>`

se *⟨hledaná-fráze⟩* nejprve transformuje podle dat z \vdef. Teprve pokud tato data pro hledanou frázi neexistují, použije se *⟨hledaná-fráze⟩* tak, jak je.

Doporučujeme: Pro přehlednost se vyplatí mít každý překlad na novém řádku, a za komentujícím procentem jeho označení, abychom bez tápání věděli, kam co patří. Poslední řádek souboru s poznámkami ke konkrétní knize, který `TEX` načte, by měl obsahovat jediný příkaz `\endinput`. Cokoliv bude následovat pod tímto pokynem na dalších řádcích, `TEX` už neuvidí. (Ale nezaměňujte ho s `\end` nebo `\bye`, aby v tomto místě neskočil svůj běh, ale pokračoval čtením dalších souborů.)

Pod příkazem `\endinput` tedy můžeme mít připraveno několik řádků např. v této podobě:

```

\ww {}={} % BKR
{}={} % PSP
{}={} % CSP
{}={} % CEP
{}={} % B21
{}={} % SNC
\Note 1:1 {}

```

a tyto řádky pak stačí zkopírovat na místo nové poznámky, upravit číslo kapitoly a verše za \Note, popřípadě umazat ={}, kde nejsou zapotřebí. Neztratíme přehled, kam patří které fráze, ať už je vypisujeme ručně nebo kopírujeme z biblického programu či on-line zdroje.

6.3 Větvení zpracování textu podle variant překladu

Pomocí příkazu \switch můžete větvit zpracování vstupního textu v závislosti na nastavené hodnotě parametru \tmark, tedy v závislosti na aktuálně zpracovávané variantě jazyka. Příkaz má tuto syntaxi:

```
\switch {<seznam variant>} {<co provést>}%
      {<seznam variant>} {<co provést>}% ... atd
      {<seznam variant>} {<co provést>}
```

Dvojic {<seznam variant>}{<co provést>} může být uvedeno libovolně mnoho. Za každou dvojici {<seznam variant>}{<co provést>} (s výjimkou poslední dvojice) musí bezprostředně a bez mezer následovat další taková dvojice, proto je nutné při přechodu na další řádek napsat za uzavírací závorku procento, které zakryje mezeru z konce řádku. Mezery na začátku dalšího řádku nevadí. Procento za dvojicí můžete číst jako „pokračuje další dvojice“.

<seznam variant> je jedna překladová varianta nebo seznam překladových variant oddělených čárkou a bez mezer. T_εX pak pracuje takto: vyskytuje-li se v <seznam variant> varianta definovaná v parametru \tmark, provede se následující <co provést>. Nevyskytuje-li se tam taková varianta, následující <co provést> se přeskočí. Příklad:

```
\switch {BKR} {Balsazar}%
      {PSP} {Bélšaccar}%
      {CSP,CEP,B21,SNC} {Belšasar}
```

Příklad ukazuje, jak vytisknout slovo Balsazar v závislosti na zpracovávané variantě jazyka.

Jakmile T_εX najde shodu a provede <co provést>, pak případně následující údaje v rámci stejného příkazu \switch přeskočí. Dále platí pravidlo, že pokud je <seznam variant> prázdný, provede se <co provést> vždy, pokud tedy není přeskočeno dle předchozího pravidla. Takže prázdný <seznam variant> na konci dvojic parametrů příkazu \switch se vyhodnotí jako „ostatní případy“. Příklad uvedený výše se tedy dá zapsat také takto:

```
\switch {BKR} {Balsazar}%
      {PSP} {Bélšaccar}%
      {} {Belšasar}
```

Příkaz \switch se dá použít nejen na jednotlivé fráze uvnitř poznámek \Note, ale třeba také na celé úseky vstupního textu obsahující třeba i několik poznámek \Note, několik definic atd.

Příkaz \switch se nedá použít v parametrech jiných maker. Tam funguje jen příkaz \x/{fráze}/.

6.4 Přechíslování veršů podle variant překladu

Některé varianty překladu mají jiné číslování veršů. V takovém případě je možné použít příkaz \renum takto:

```
\renum <zkratka-knihy> <výchozí-č-kap>:<výchozí-č-verše> = <překlad> <č-kap>:<č-od>-<č-do>
```

kde <překlad> je značka pro konkrétní překlad. Místo <výchozí-č-kap>:<výchozí-č-verše> se použije při \def\tmark{<překlad>} ve skutečnosti <č-kap>:<č-od>. Takové přechíslování se netýká jen tohoto verše, ale celého úseku veršů vymezeného rozsahem <č-od>-<č-do>.

Například:

```
\renum Da 6:1 = CSP 6:2-29
\renum Da 6:1 = CEP 6:2-29
\renum Da 6:1 = B21 6:2-29
\renum Da 6:1 = SNC 6:2-29
```

Pavlíkův studijní překlad má v na přelomu 5. a 6. kapitoly číslování shodné s Bibli kralickou, takže není nutno ho přechíslovat. Ostatní překlady uvádějí obsah verše BKR 6:1 (Libilo se pak Dariovi...) pod číslem 6:2 a celá kapitola 6 je o jedno číslo posunutá vůči BKR. BKR a PSP 6:28

je v CSP, CEP, B21 a SNC 6:29. Poznámky číslováme podle překladu, uvedeného jako první parametr definice `\variants` v souboru `Cze-vars.tex`, tedy v naší ukázce BKR, a to včetně odkazů, např. Viz pozn. 5:31. Přechíslované překlady změni číslo poznámky podle skutečného čísla verše, který poznámka komentuje, včetně odkazu, který se vytiskne jako Viz pozn. 6:1.

Má-li se přechíslování týkat jediného verše, je třeba uvést shodné `<č-od>` i `<č-do>`, jako např.

```
\renum Da 5:31 = CSP 6:1-1
\renum Da 5:31 = CEP 6:1-1
\renum Da 5:31 = B21 6:1-1
\renum Da 5:31 = SNC 6:1-1
```

Po takové deklaraci se nakonec poznámka

```
\Note 5:31 {Darius}={\x/Darius/ Médský} Některé školy tvrdí, že tento a další
(<6:1>, <6:6>, <6:9>, <6:25>, <6:28>; <9:1>; <11:1>)
odkazy na \x/Daria/ Médského v knize \x/Daniel/ jsou historické omyly.
```

vytiskne takto:

6:1 Darjavěš Médský Některé školy tvrdí, že tento a další (6:2, 6:7, 6:10, 6:26, 6:29; 9:1; 11:1) odkazy na Darjavěše Médského v knize Daniel jsou historické omyly.

Lze si poradit i s takovými lahůdkami, jako je posun čísla o pouhou část verše. BKR Da 2:28 „jest Bůh na nebi, kterýž zjevuje tajné věci“ je v SNC „Bůh, který je v nebesích, odhaluje tajemství.“ pod číslem 2:27; před poznámkou k této frázi tedy uvedeme `\renum Da 2:28 = SNC 2:27-27`. Ale pak potřebujeme poznámku i k frázi „v potomních dnech“ z téhož verše, avšak – světe, div se – fráze „posledních dnů“ už je v SNC ve verši 2:28! Nepotřebujeme spekulovat, proč na nás překladatelé SNC zkoušejí takové triky, že půl verše pošlou pod jiné číslo, než všichni ostatní; postačí nám, že víme, jak si s tím poradit: Před tuto novou poznámku napíšeme `\renum Da 2:28 = SNC 2:28-28` a všechno funguje, jak má: Kde se číslování rozchází, tam se přechísluje; kde se shoduje, zůstane shodné.

7 Metody vytváření hyperlinkových odkazů

Odkaz je úsek textu, podle kterého čtenář i po vytištění pozná, na jaké jiné místo v textu (nebo v internetu) se může podívat. Obsahuje tedy typicky číselný údaj o stránce nebo o čísle kapitoly, sekce atd. Navíc, pokud čtenář pracuje s PDF prohlížečem, pak tento úsek textu je *aktivní*, tj. po najetí myším kurzorem nad tento text je možné kliknout a PDF prohlížeč přejde na stanovené místo v dokumentu (nebo do internetu).

Bible je neměnným způsobem strukturovaný text. Obsahuje (v protestantském kánonu) 66 knih s ustálenými značkami pro tyto knihy, každá kniha má své kapitoly číslované od jedné a každá kapitola má verše číslované od jedné. Není tedy potřeba nechat \TeX , aby generoval tato čísla automaticky (jako to dělá při sazbě třeba odborného textu členícího se na kapitoly a sekce) a není tedy vesměs potřeba používat lejblíky ve zdrojovém dokumentu (které \TeX při zpracování přiřadí k vygenerovaným číslům) a odkazovat pomocí těchto lejblíků, jak je popsáno v sekci 1.4.3 v dokumentaci $\text{Op}\TeX$ u. Je daleko účelnější odkazovat přímo do konkrétního místa v Bibli, které má už po mnohá staletí pevně označení knihy, čísla kapitoly a čísla verše. (Canterburský arcibiskup Stephen Langton počátkem 13. století, když vyučoval na univerzitě v Paříži [a ještě nebyl arcibiskupem] rozdělil Bibli na kapitoly. V polovině 16. století pak francouzský knihtiskař Robert Estienne rozdělil Nový Zákon na verše a k tomu připojil Starý Zákon, který židovští písmáci o staletí dříve byli rozdělili na verše, avšak rafinovaně nikoliv na kapitoly. Od roku 1553, kdy Estienne vydal první francouzskou Bibli takto očíslovanou, tento systém používáme dodnes.)

Odkazy píšeme na konkrétní místo v Bibli mezi `< a >`. Text mezi těmito znaky se vytiskne tak, jak je napsán (až na výjimky posané níže). \TeX ale musí umět odkaz správně interpretovat, aby jej mohl udělat aktivní s možným proklikem na správné místo v Bibli. K tomu slouží poměrně značné množství pravidel, kterým se věnuje tato sekce.

7.1 Základní pravidlo s úplnými údaji

Odkaz mezi znaky `< a >` má tvar `"<text>" <údaj>` nebo jen `<údaj>`. Přitom úplný `<údaj>` má tvar `<kniha> <kapitola>: <verš>`. Zde `<kniha>` je zkratka knihy (za ní musí být mezera), `<kapitola>` je číslo kapitoly a `<verš>` je číslo verše. Příklad:


```
... viz také verš <Jr 8:13>
... viz také <"verš" Jr 8:13>
```

V prvním případě se vytiskne ... viz také verš [Jr 8:13](#) a ve druhém ... viz také [verš Jr 8:13](#). Aktivní (umožňující proklik) bude v obou případech oblast zde vyznačena modře a proklik vede na Jr 8:13.

7.2 Specifikátor odkazu

Za koncovým znakem odkazu > může těsně následovat specifikátor odkazu, což je jedno z písmen:

- n ... odkazuje se na poznámku (note),
- g ... odkazuje se na glosu (která přijde ke slovu, vznikne-li někdy další verze OpBible s dvou-sloupcovou sazbou základního textu),
- a ... odkazuje se na článek (article),
- i ... odkazuje se na úvod (introduction).

Specifikátor odkazu se nevytiskne, je to jen interní informace, kam má směřovat aktivní odkaz (tj. proklik myši). Nenásleduje-li za ukončujícím znakem údaje > žádný z uvedených specifikátorů, jedná se o odkaz na verš (to bude asi nejčastější případ). Příklad odkazu na poznámku:

```
... viz <"poznámku k" Jr 7:4>n, kde je uvedeno více informací.
```

vytiskne ... viz [poznámku k Jr 7:4](#), kde je uvedeno více informací. Proklik vede na první poznámku k verši Jr 7:4, nikoli k verši samotnému.

V případě odkazu na článek (specifikátor a), má úplný údaj formát *<kniha>* *<kapitola>*, tj. chybí informace o verši, protože články můžeme chápat jako úvody ke kapitolám. V případě odkazu na úvod ke knize (specifikátor i) má úplný údaj formát *<kniha>* (chybí informace o kapitole i verši), protože to jsou úvody ke knihám. V ostatních případech má úplný údaj formát, jak bylo řečeno v sekci 7.1 s výjimkou popsanou v sekci 7.3

7.3 Výjimka pro formát úplného údaje u některých knih

Knihy Abd, Fm, 2Jn, 3Jn, a Ju nejsou členěny na kapitoly. V odkazu na verš, poznámku k verši nebo glosu do některé z těchto knih tedy chybí informace o kapitole a formát úplného odkazu vypadá takto: *<kniha>* *<verš>*. Aby \TeX věděl, že má uplatnit tuto výjimku, je třeba seznam zkratk těchto knih definovat v makru `\nochapbooks`. Například v souboru `Cze-books.tex` je řečeno

```
\def\nochapbooks{Abd Fm 2Jn 3Jn Ju}
```

Protože pro různé jazyky existují různé zkraky knih, je třeba toto makro definovat v závislosti na použitém jazyku.

7.4 Neúplný údaj

Někdy dokáže umístění verše čtenář stanovit z kontextu, takže *<údaj>* nemusí být úplný. V *neúplném údaji* může chybět *<kniha>* nebo *<kniha>* *<kapitola>*: nebo všechno. Například:

```
... analogii vidíme též ve <"verších" Jr 8:13>, <9:7> a <11:3>
... viz verše <Jr 8:13>, <15>, <17>
... viz všechny <"verše" Jr 8:13>--<22>,
... (srv. <Jr 8:13> a <"jeho poznámku">n).
```

Vytiskne se: ... analogii vidíme též ve [verších Jr 8:13, 9:7 a 11:3](#) ... viz verše [Jr 8:13, 15, 17](#) ... viz všechny [verše Jr 8:13–22](#). ... (srv. [Jr 8:13](#) a [jeho poznámku](#)).

Údaje 9:7, 11:3, 15, 17, 22 a poslední prázdný údaj v těchto příkladech jsou neúplné. Čtenář ví, že se týkají knihy Jeremjáš a že v případě, kdy není uvedena ani kapitola, se týkají kapitoly 8 knihy Jeremjáš. V posledním příkladě s prázdným údajem čtenář ví, že se jedná o poznámku k Jr 8:13. Ví to i \TeX a přidělí ke všem těmto neúplným odkazům správné interní odkazy (podle kterých se provede proklik), protože neúplný údaj přebírá nespecifikované informace z předchozího údaje. Toto pravidlo se vztahuje lokálně na jeden objekt textu: poznámku, článek, úvod atd. Je-li hned první údaj v některém objektu textu neúplný, nespecifikované informace se nahradí zkratkou aktuálně zpracovávané knihy, případně číslem aktuální kapitoly, případně verše.

Je-li neúplný údaj uvozen znakem \, přebírá nespecifikované informace z aktuálně zpracovávané knihy, případně kapitoly a případně verše bez ohledu, jaký údaj předchází. Například:

\Note 4.5 {} Myšlenka se opakuje v <Jr 4:5>. Ale také v \<8:3> najdete o tom zmínku.

Zde odkaz 8:3 vede na verš 3 kapitoly 8 aktuální knihy. Kdyby tam znak \ nebyl, pak by tento odkaz vedl na 8:3 knihy Jeremjáš.

Neúplný údaj se vytiskne tak, jak je, jako neúplný. Shora uvedená pravidla na jeho doplnění se provedou jen interně, aby aktivní odkaz správně fungoval po kliknutí myší.

Existuje ještě jeden odlišný formát neúplného údaje: chybí číslo verše. To se pozná tak, že je přítomná zkratka knihy a v údaji chybí dvojtečka a nejedná se o knihu ze seznamu výjimek \nochapbooks. Tedy údaj má tvar <knihy> <kapitola> (oddělené mezerou). T_EX v takovém případě vytvoří interní odkaz na první verš dané kapitoly. Příklady:

```
... viz <"kapitolu" Da 7>
... Josefův příběh (<Gn 39-41>)
```

Ve druhém příkladě je uveden rozsah kapitol a T_EX vytvoří interní odkaz na první verš kapitoly 39. Srovnajte též se sekci 7.5.

7.5 Formát pro rozsah veršů a pro úsek ve verši

V každém údaji je možné místo <verš> nebo místo <kapitola>: <verš> psát rozsah veršů ve formátu <od>-<do>. T_EX vytvoří interní link jen na první verš rozsahu a spojovník (znak -, ASCII 45) uvedený v rozsahu promění v pomlčku. Příklady:

```
<Jr 8:3-7>,
<Jr 8:3-9:5>,
<3-7>,
<8:3-7>.
```

Třeba první odkaz z tohoto příkladu se vytiskne jako [Jr 8:3-7](#) a nabízí proklik jen na Jr 8:3.

Někdy je potřeba odkazovat na úsek verše, nikoli na celý verš. To se dělá připojením písmene bezprostředně za číslo verše. Například

```
... viz <Da 9:11b>
```

Takové písmeno můžete připojit k úplnému i neúplnému odkazu. Pro potřebu hyperlinku se tato písmena ignorují, ale vytisknou se. Tedy uvedený příklad vytiskne ... viz [Da 9:11b](#), ale proklik vede na Da 9:11.

7.6 Zatajení údaje

Když nepostačuje žádné z pravidel pro vytvoření interního linku z odkazu uvedené výše, můžete vše, co se má v odkazu vytisknout, uzavřít do "... " a následný údaj, podle kterého se má vytvořit interní odkaz, zatajit. K tomu stačí těsně za závírací uvozovku " připojit znak podtržítka _ následovaný údajem. Tento údaj se netiskne. Například

```
<"První kniha Samuelova"_1Sa 1:1>
```

vytiskne pouze text [První kniha Samuelova](#), který interně odkazuje na první verš této knihy.

7.7 Přechíslování odkazu

Pokud odkaz směřuje na verš, který má v konkrétním jazykovém překladu odlišné číslování specifikované makrem \renum, pak do odkazu zapište údaj podle výchozího číslování a T_EX si jej sám přepočítá podle dat uvedených v \renum. Vytiskne přepočítaný údaj a použije jej pro interní odkaz. Srv. 6.4.

7.8 Redukce údajů

Možná budete chtít do závorek pro odkaz psát úplné údaje a požadovat jejich automatickou redukci na neúplné, odkazují-li na aktuální knihu. Neodkazují-li na aktuální knihu, údaj zůstane úplný. Toho můžete docílit přidáním \re před otevírací závorkou údaje (pravidlo případné redukce se uplatní jen na tento jediný údaj) nebo pomocí příkazu \reduceref. Když jej použijete v poznámce (nebo v T_EXové skupině), pravidlo redukce se uplatní pro všechny následující úplné údaje této poznámky (této T_EXové skupiny). Když jej použijete v deklaraci sazby, pravidlo případné redukce se uplatní na všechny úplné údaje v dokumentu. Příklad:

`\re<"verš" Da 7:3>`

se vytiskne jako [verš Da 7:3](#), je-li tento odkaz uveden mimo knihu Daniel. Když je ale tento odkaz uveden v knize Daniel, vytiskne se jen [verš 7:3](#), který interně odkazuje na Da 7:3.

Pravidlo redukce odkazů nastavené pomocí `\reduceref` lze vypnout příkazem `\noreduceref`. Od něj až do konce poznámky (T_EXové skupiny) se odkazy chovají, jakoby `\reduceref` nebylo zapnuté.

7.9 Značka pro knihu se může vytisknout jinak

Je-li značka pro knihu deklarovaná v různých překladových variantách různě pomocí `\vdef`, pak v odkazech používejte výhradně údaj `<kniha>` podle první varianty. Je-li ale pomocí `\tmark` nastavena nějaká alternativní překladová varianta, nakonec se odkaz vytiskne podle údaje `\vdef` této překladové varianty. Vnitřně se ale odkazy propojují podle základní varianty. Tuto vlastnost využijete, pokud deklarujete pomocí `\vdef` třeba pro značky 1Pa a 2Pa (Paralipomenon) alternativní text 1Let a 2Let (Letopisů) při variantě B21 (Bible pro 21. století). Když pak napíšete odkaz `<"viz" 1Pa 2:3>`, dostanete [viz 1Pa 2:3](#) v obvyklých překladových variantách, ale vytiskne se [viz 1Let 2:3](#), použijete-li překladovou variantu B21.

7.10 Nepodařené odkazy, tj. odkazy na neexistující místo

Je-li uveden odkaz na neexistující verš nebo neexistující poznámku, pak nastanou dvě možnosti. Je-li to odkaz na knihu, která se záměrně netiskne (protože například pracujeme s testovacím výtiskem jen vybraných knih Bible, viz též `\printedbooks` v sekci 4.1), pak odkaz je sice obarven, jakoby byl aktivní, ale aktivní není a v logu při zpracování T_EXem se neobjeví žádné varování. Pokud ale odkaz směřuje na neexistující verš nebo poznámku tištěné části Bible, pak odkaz je aktivní, proklik směřuje na poslední stránku PDF souboru a v logu se objeví o tom varování, že odkaz není správný. Při prvním zpracování T_EXem jsou ale neošetřeny všechny odkazy a v logu je velké množství varování o nesprávných odkazech. Teprve při následujícím zpracování T_EXem jsou odkazy směřující do existujícího místa správně prolinkovány.

7.11 Trasování odkazů

Implicitně je zapnuto vypisování podrobného trasování odkazů v log souboru. Vypnete to pomocí `\no-tracinglinks` a zpětně zapnete pomocí `\tracinglinks`. Navíc pomocí `\tracingouterlinks` vypnete potlačení varování o odkazech na neexistující knihy, což umožní v logu najít případné neexistující odkazy z důvodu překlepu ve značce knihy.

7.12 Odkazy na kapitoly a na knihy

Chcete-li odkázat na knihu s proklikem na začátek této knihy, pište `\cref[<kniha>]`, například `\cref[Gn]` vytiskne [Gn](#) s proklikem na začátek knihy Genesis. Podobně `\cref[Gn 2]` vytiskne [Gn 2](#) s proklikem na začátek kapitoly Den odpočínutí. Chcete-li vytisknout něco jiného, pak za údajem `[. . .]` musí těsně následovat `{<text>}`, kde `<text>` je text, který se má vytisknout a stát se aktivním odkazem. Takže třeba `\cref[Gn 2]{Den odpočínutí}` vytiskne [Den odpočínutí](#) s proklikem na odpovídající kapitolu.

7.13 Odkazy na stránky

Ve svém textu můžete umístit netištěný cíl odkazu na stránku pomocí `\pglabel[<lejblík>]` a pak můžete odkazovat na stránku pomocí `\pgref[<lejblík>]`. V místě `\pgref[<lejblík>]` se vytiskne číslo strany, které je možné prokliknout. Podobně jako v případě `\cref` se dá pomocí příkazu `\pgref[<lejblík>]{<text>}` vytisknout jiný `<text>` než číslo strany, který proklikne na místo, kde je umístěný `\pglabel[<lejblík>]`.

8 Mapy, obrázky a jejich legendy

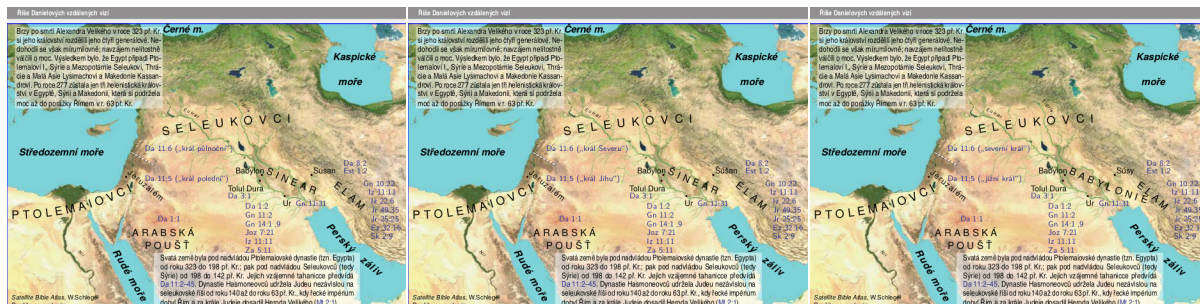
8.1 Varianty překladu

Podobně jako lze měnit znění hledané fráze v poznámce, aby odpovídalo skutečné podobě aktuálního základního textu, lze připravit popis mapy (a podobných grafických objektů, zobrazujících výrazy, které se v různých překladech liší) tak, aby vždy odpovídal aktuální verzi Bible.

Za předpokladu, že máme k dispozici slepou mapu, kterou chceme popsat svou legendou, se jakýkoliv text umisťuje přes obrázek např. takto:

```
\puttext 5mm 62mm{{\bi Středoziemní moře}}
```

První rozměr (5mm) určuje horizontální posun na ose x, druhý pak vertikální na ose y, kdy souřadnice 0mm 0mm má levý dolní roh obrázku.



Mapa z Úvodu k Danielovi: BKR

CEP

B21

Například si můžete všimnout, jak se v ukázce výše mění nejen fráze „král půlnocní“ a „král polední“ (BKR) na „král Severu“ a „král Jihu“ (CEP) a „jižní král“ a „severní král“ (B21), což je běžný aktivní link jako v poznámkách, ale také samotný popis mapy, např. jméno města Susan (kousek nad Perským zálivem) z králičtiny na Šušán v CEP a na Súsý podle B21. Podobně jméno oblasti Sinear metamorfuje na Šineár a na Babylonii. Postarají se o to definice \vdef v souboru Cze-vars.tex.

Obrázek musí být načten před zahájením popisu, a všechno, co na něm má být, musí být uvnitř definice \insertBot. Výše uvedená ukázka byla zapsána jako

```
\insertBot {Říše Danielových vzdálených vizí}[mapa](){
\inspic{fertile-crescent-crop.pdf}% slepá mapa
\Heros \cond \setfontsize{at 9pt}\rm %font
\vskip=1mm
\putstext 2mm 108mm {\vtop{\hspace{6.5cm}%šířka boxu \vtop
\baselineskip10pt %řádková rozteč uvnitř boxu
\noindent %šetříme místem, nepotřebujeme odsazovat
\leftskip=3pt \rightskip=3pt %o kolik bude poloprůhledný stín přesahovat text
Brzy po smrti Alexandra Velikého ... až do porážky Římem v r. 63 př. Kr.}%konec\vtop
}%konec \putstext
\LMfonts \sans \setfontsize{at 9pt}\rm
\puttext 145mm 29mm {"Sk 2:9" _Sk 2:9)}
\puttext 145mm 32.5mm {"Ez 32:16" _Ez 32:16)}
.
.
.
\puttext 2mm 5mm{{\Heros \setfontsize{at 7pt}\it Satellite Bible Atlas,\ / \rm W.Schlegel}}
\puttext 2mm 2mm{{\Heros \setfontsize{at 7pt}\rm Použito se svolením.}}
}%konec \insertBot
```

Na co si dát pozor: Uvnitř \insertBot se nesmí vyskytnout prázdný řádek.

8.2 Makro \town pro symbol města na mapě

Města Jeruzalém, Babylon, Tolul Dura, Susan a Ur jsou na mapě vidět jako drobné kroužky s červeným středem a černým obvodem. Vlastnosti tohoto kroužku lze nastavit makrem \townparams, jehož implicitní hodnoty jsou tyto:

```
\def\townparams{
\hhkern=.8pt % poloměr kroužku
\lwidth=.5pt % tloušťka obrysové čáry
\fcolor=\Red % barva kroužku
\lcolor=\Black % barva obrysové čáry
}
```

Samotné makro `\town` umísťuje tuto značku súradnicami, podobne ako makro `\puttext`, ale bez ďalšieho textu, teda napr.

```
\town 101.5mm 53mm %mesto Babylon
```

8.3 Naklonený text

V ukázke máme meno mesta Jeruzalém vytištené šikmo, aby nekolidovalo s „kráľom Jihu“ a bolo dobre vidieť. Toho lze dosáhnout přidáním dalších parametrů k definici `\puttext`:

```
\puttext 48mm 55mm {\c[-40/\kern1pt\pdfrotate{0}]{Jeruzalém}}
```

O naklonění nápisu se postaralo číslo -40. Další údaje viz 8.4.

8.4 Nápis podél křivky

Některé nápisy na mapě vyžadují „prohnout“ podle terénu, hlavně názvy rozlehlých oblastí, v našem případě Ptolemaiovcí a Seleukovci, či Perský záliv, nebo třeba také drobnější jména řek (Eufrat, Tigris, Nil). Budeme měnit doplňující parametry k `\puttext`, jak už jsme je viděli při umísťování Jeruzaléma do mapy v 8.3.

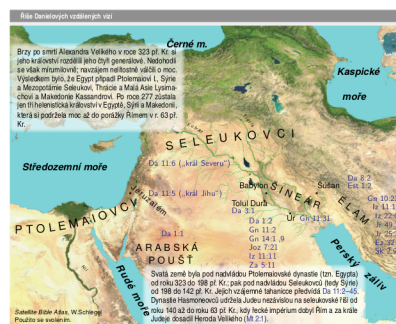
Příkaz `\kern` určí rozteč mezi písmeny; číslice v definici `\pdfrotate` pak sílu zakřivení. Záporná hodnota prohne nápis konkávně (jako duhu), kladná konvexně (jako mísu). Dynastie Alexandrových potomků jsou na naší mapě zapsány takto:

```
\puttext 62mm 70mm {\c[10/\kern7pt\pdfrotate{-1}]{SELEUKOVCI}}
\puttext 2mm 37mm {\c[0/\kern4pt\pdfrotate{2.5}]{PTOLEMAIOVCI}}
```

Kdybychom potrebovali nápis, který by vlnil do tvaru písmene S (byl by současně konkávní i konvexní), musíme ho seskládat ze dvou či více příkazů `\puttext`, slepených k sobě, aby vypadaly jako jeden souvislý text.

8.5 Částečně průhledné pozadí souvislého textu

V ukázce 8.1 byl dvojí souvislý text v mapě původně umístěn do světlejších obdélníků, připravených grafickým editorem (obrázek vlevo). Textem pak bylo nutno se do nich trefit; pokud to nevyšlo, musel se předělat obrázek a zkoušet se trefit znovu.



Šlo to, ale bylo to únavně pracné a časově náročné. OpBible nabízí nápaditější řešení: Když místo `\puttext` (s příslušnými parametry, viz 8.4) napíšeme `\putstext` (Put-Shadowed-Text), pak jakkoliv dlouhý text, který napíšeme, bude automaticky podložen částečně průhledným bílým pozadím; obrázek slepé mapy nepotřebuje žádné předpřipravené plochy. Před prvním použitím `\putstext` lze nastavit úroveň průhlednosti bílého stínu úpravou implicitní hodnoty `\def\shadowparameter{.1}` (obrázek vpravo). `\def\shadowparameter{1}` znamená solidní neprůhledné bílé pozadí; menší číslo znamená větší průhlednost. Poté se však tato hodnota uloží do tzv. page-resources výstupního PDF a na všech dalších stránkách se používá stejná, nelze ji tedy měnit a mít na různých místech jednoho dokumentu různou. Kdyby snad měla vzniknout nečekaně nadměrná poptávka po možnosti úroveň průhlednosti za jízdy měnit, může se to stát podnětem pro implementaci v případné další verzi OpBible. V tuto chvíli se nám nezdálo nutné komplikovat makra zakládáním dalších a dalších page-resources, pročez uživatel nechť se spokojí s možností nastavit průhlednost stínu pod texty na mapách pro celou Bibli jednotně.

9 Nástroje na zařazení časových os

9.1 Obrázek či text přes dvě strany

Pokud chceme vložit obrázek nebo text přes dvě stránky v otevřené dvoustraně, můžeme použít

- `\insertSpanImage`: vloží připravený PDF obrázek, lze použít v úvodu knihy,
- `\insertSpanText`: vloží text (například časovou osu), lze použít v úvodu knihy,
- `\putSpanImage`: vloží připravený PDF obrázek, kotví se vzhledem k číslu kapitoly a verše, lze použít v souboru s poznámkami,
- `\putSpanText`: jako `\putSpanImage`, ale místo obrázku vkládá text.

Příkazy `\insertSpanImage` a `\insertSpanText` umístí obrázek či text do spodní části dvou stránek podle následujícího pravidla. Předpokládejme, že příkaz samotný je spuštěn v okamžiku, kdy \TeX vytváří aktuální stránku s číslem c . Pak

- je-li c sudé a obrázek či text se na výšku do aktuální strany vejde, bude umístěn na stranách c a $c + 1$,
- je-li c sudé a obrázek či text se na výšku do aktuální strany nevejde, bude umístěn na stranách $c + 2$ a $c + 3$,
- je-li c liché, obrázek či text se umístí na stránky $c + 1$, $c + 2$.

Tím je zajištěno, že obrázek či text bude vždy viditelný na dvojstraně otevřené knížky.

Příkazy `\putSpanImage` či `\putSpanText` pracují podle stejného pravidla, jen číslo c v tomto případě odpovídá číslu strany, na které se objeví začátek verše specifikovaného v parametru příkazu.

Uvedené příkazy mají tyto parametry:

`\insertSpanImage` {Titul} [*lejblik*] (*parametry*) {*jméno souboru*} Zde se *Titul* použije v záhlaví obrázku a *lejblik* je možné stanovit, chcete-li se na obrázek odvolávat (a tím vytvořit aktivní odkaz) pomocí `\ref`. Nechcete-li použít lejblik, nechte parametr prázdný: []. Dále *parametry* mohou upřesnit způsob umístění obrázku, typicky je tento údaj prázdný: (). Konečně *jméno souboru* je jméno souboru s obrázkem včetně přípony. Typicky se jedná o PDF soubor, tj. má příponu .pdf.

`\insertSpanText` {Titul} [*lejblik*] (*parametry*) {*text*} Parametry jsou stejné jako má `\insertSpanImage`, jen poslední parametr se liší. Je v něm text, který se má vytisknout přes dvě strany. Typicky tam může být použita sada příkazů `\timeline`, `\timelinewidth`, `\arrowtext`, `\tlput`, `\tline`, `\tlines` pro vytvoření časové osy, viz sekci 9.2.

`\putSpanImage` *kapitola*:*verš* {Titul} [*lejblik*] (*parametry*) {*jméno souboru*} Navíc je zde *kapitola*:*verš* specifikující verš, jehož začátek je na straně c a obrázek se umístí podle pravidla výše. Ostatní parametry jsou stejné jako u `\insertSpanImage`.

`\putSpanText` *kapitola*:*verš* {Titul} [*lejblik*] (*parametry*) {*text*} se chová stejně, jako `\insertSpanText`, ale navíc parametr *kapitola*:*verš* má stejný význam jako v příkaze `\putSpanImage`.

9.2 Příkazy k vytvoření časové osy

Tyto příkazy je možné použít v parametru *text* v příkazech `\insertSpanText` nebo `\putSpanText`

Nejprve je třeba stanovit počet roků (či jiných jednotek) na celou šířku osy. V těchto jednotkách se budou zadávat všechny ostatní údaje. V dalším textu budeme těmto jednotkám říkat roky. Nastavení parametrů časové osy provedeme pomocí `\timeline` a `\timelinewidth`:

```
\timeline <počet roků>
\timelinewidth <šířka osy>
```

Například po

```
\timeline 500
\timelinewidth 25cm
```

bude osa široká 25 cm a v této šířce bude zahrnuto 500 let, takže jeden rok bude představovat šířku 0,5 milimetru. Ovšem je obvyklejší zadat šířku osy jako nějaký díl celkové šířky strany (či dvojstranného obrázku, použije-li se osa v příkaze `\insertSpanText` nebo `\putSpanText`). Například takto: `\timelinewidth=.95\hsize`.

Časovou osu sestavujeme po řádcích. Příkazy pro texty či úsečky, které jsou ve stejném řádku, píšeme pod sebe v libovolném pořadí. K posunu na další řádek slouží příkaz `\vskip` *<rozměr>*, přitom rozměr můžete udávat v násobcích výšky řádku při použití jednotky `\l`, tedy třeba `\vskip 1.5\l` znamená posun o jeden a půl řádku dolů.

Text vložíme pomocí `\tput` *<příznak>* *<pozice>* *<praporek>* (*<nastavení>*) *<text>*. Parametr *<příznak>* může být `a`, chceme-li, aby text byl nad místem aktuální polohy sazby, nebo `b`, chceme-li mít text pod místem aktuální polohy sazby. Když je *<příznak>* `a`, víceřádkový text se roztahuje směrem nahoru a při `b` směrem dolů. Parametr *<pozice>* je místo na časové ose (v rocích), ke kterému má být text připojen. Od tohoto místa bude text přetékat doprava, je-li *<praporek>* specifikován pokynem `\rlap`, bude přetékat doleva, je-li *<praporek>* specifikován pokynem `\llap`. Konečně, je-li *<praporek>* prázdný, bude mít text centrované řádky a *<pozice>* pak odpovídá tomuto středu řádků. Parametr *<nastavení>* může být prázdný, tedy `()`, nebo obsahuje nastavení fontu, barvy fontu atd. pro následující *<text>*. A *<text>* obsahuje text k vytištění. V případě víceřádkového textu oddělujte jednotlivé řádky příkazem `\cr`. Příklad:

```
\tput b 25 (\it) {Abraham\cr má 100 let}
```

Text má dva řádky, jejich společný střed je pod bodem 25 časové osy.

`\tline` *<od>*..*<do>* vytvoří vodorovnou úsečku začínající v místě *<od>* a končící v místě *<do>*. Údaje jsou v rocích.

`\tlines` *<w1>*|*<w2>*|*<w3>*| (a případně další) vloží do osy krátké svislé čárky. Počet parametrů *<w>*| může být libovolný, každý z nich reprezentuje jednu svislou čárku a čísla mezi nimi udávají vzdálenost mezi sousedními čárkami v rocích. Například

```
\tline 0..100
\tlines 20|20|20|20|
```

vytvoří stoletou vodorovnou linku a na ní svislé linky označující 20, 40, 60 a 80 roků.

`\arrowtext` *<od>*..*<do>* (*<nastavení>*) *<text>* vytiskne vodorovnou úsečku z místa *<od>* do místa *<do>* (údaje jsou v rocích) a prostředek úsečky je přerušen a je v tomto místě napsán *<text>*. Na okrajích úsečky jsou šipky směřující směrem ven s úsečky.

10 Varianty formátování stránek

V implicitní podobě je nastavena jednosloupcová sazba pro hlavní text Bible a pro úvody ke knihám a pro anotace. Dvousloupcová sazba je nastavena pro poznámky k veršům.

Implicitně jsou čísla kapitol velká a ve vnějším okraji. Ve vnějším okraji jsou též zvětšené značky uvozovek připojené k citacím. Příkazem `\normalchapnumbers` dojde ke změně tohoto nastavení: čísla kapitol pak jsou vsazena do levého horního rohu prvního odstavce a zvětšené značky uvozovek jsou zrušeny. Vnější okraj je pak zmenšen, protože v něm už není žádný tiskový materiál.

Implicitně zaujímá každá poznámka `\Note` nový odstavec ve dvousloupcové sazbě. Pomocí `\mergednotes` je možné zajistit, že všechny poznámky vztahené k jedinému verši jsou vždy spojeny do společného odstavce. Ale je to za cenu toho, že se nezaručí, aby výskyt frází v hlavním textu byl na stejné straně jako fráze v poznámce. Celý odstavec spojené poznámky se propojí pouze se začátkem verše, tj. začátek verše a začátek poznámky k verši jsou na stejné straně, samotné fráze se neprohledávají.

Další varianty formátování stránek jsou zatím jen v plánu a v této verzi OpBible nejsou implementovány.

11 Možnosti hledání chyb

Může se stát, že uděláte nějaký překlep v souborech `notes-*`, `intro-*`, `articles-*`. Kdybyste soubor rovnou zařadili do celého zpracování Bible, TeX ohlásí chybu ve zcela jiném okamžiku, než když čtli příslušný soubor a zpětné dohledávání takové chyby je velmi obtížné, navíc se typicky chyba projeví v jiné fázi zpracování a tedy je oznámena často velmi nesrozumitelným a zavádějícím způsobem.

Je ovšem možné nově napsané soubory nejprve zpracovat přímo, ale bez propojení s hlavním textem a v takovém módu jsou chyby hlášeny přímočařeji. Přímé hledání chyb provedete příkazem `\checksyntax` *<seznam souborů>* `{}` Zde seznam souborů jsou názvy kontrolovaných souborů bez přípony `.tex`, takže například můžete mít na konci hlavního souboru:

```
\checksyntax intro-Dan articles-Dan notes-Dan {}
```

provede přímou kontrolu vyjmenovaných souborů. Na výstupu je text těchto souborů bez důrazu na jakékoli formátování. Syntaktické chyby v souborech se ale projeví přímočaře.

Poznámka: použití `\checksyntax` deaktivuje příkaz `\processbooks`, který v takovém případě neprovede nic, takže základní text Bible se vůbec nenačte.

12 Generování výchozích souborů notes, fmt, intro

Pokud soubory `notes`, `fmt`, `intro` neexistují, OpBible se nezlobí a chová se stejně, jakoby byly prázdné. Možná ale budete chtít mít tyto soubory pro všechny knihy Bible již připraveny v odpovídajících adresářích a s krátkým úvodním obsahem. Šablony pro takové soubory si můžete napsat do zvláštního souboru (např. `templates.tex`) a pak tento soubor zpracovat příkazem `optex templates`. Tím vygenerujete výchozí soubory podle specifikovaných šablon pro všechny knihy Bible. Pokud soubor s daným jménem už existuje, zůstává zachován (tj. není přepsán výchozím souborem) a objeví se o tom varování na terminálu a v log souboru.

Nejprve potřebujete soubor `books.tex` s názvy a zkratkami všech knih Bible, jak se o něm píše v sekci 4.3. Ten může být vygenerován třeba z Swordu skriptem `mod2tex`, viz sekci 4.4.

V souboru `templates.tex` je třeba pomocí `\input` nejprve načíst makro `opbible.opm`, dále soubor `books.tex` a potom můžete stanovit šablony souborů příkazem `\filegen` takto:

```
\filegen {<název-souboru>}
<obsah souboru>
\endfile
```

přitom `<název souboru>` musí obsahovat a `<obsah souboru>` může obsahovat dvojznak `@@`, který se automaticky zamění za zkratku knihy `<bmark>` stanovenou v `bible.tex` a vygeneruje se tolik výchozích souborů, kolik je v `books.tex` deklarováno titulů knih. Například `templates.tex` může vypadat takto:

```
\input opbible.opm
\input books.tex

\filegen {intro-@@.tex}
% Úvod do knihy @@
\endfile

\filegen {notes-@@.tex}
% Poznámky ke knize @@

%\ww {} {} {} {} {} {}
% BKR PSP CSP CEP B21 SNC
%\Note 1:1 {} text ukončen prázdným řádkem
\endfile
```

Příkaz `optex templates` v tomto případě vygeneruje 66 souborů `intro-1Jn.tex`, `intro-1kor.tex`, atd. se specifikovaným jednořádkovým obsahem a dále 66 `notes` souborů se specifikovaným pětiřádkovým obsahem.

V `<obsahu souboru>` můžete použít také trojznak `@@@`, který se zamění ze celý titul knihy. Takže po specifikaci `% Úvod do knihy @@@` dostanete soubory s textem `% Úvod do knihy První Mojžíšova (Genesis)`, dále `% Úvod do knihy Druhá Mojžíšova (Exodus)` atd.

V ukázce jsou názvy souborů voleny tak, že se vygenerují v aktuálním adresáři. Máte-li v tomto adresáři připraveny podadresáře odpovídajících názvů, je možné soubory generovat přímo do nich, píše třeba `\filegen {intros/intro-@@.tex}`.

Chcete-li vygenerovat jiný seznam souborů, než odpovídá seznamu `<bmark>` uvedeném v souboru `books.tex`, definujte po přečtení `books.tex` makro `\genbooks` a Vámi zvolenými zkratkami. Například po

```
\def\genbooks {Gn Ex Lv Nu Dt Joz}
```

bude příkaz `\filegen` generovat jen šablony pro pět knih s uvedenými zkratkami. Definovat `\genbooks` můžete také před každým `\filegen`, pak budete mít zkratky knih pro různé typy souborů různé.

13 Shrnutí uživatelských definic

Tím se myslí \amark, \tmark atd? Nestačí rejstřík, který je připojen?

14 Rejstřík

\amark 8, 10	\genbooks 28	\putSpanImage 15, 26
\arrowtext 27	\ind 13	\putSpanText 26
\Article 16	\input {Cze-books.tex} 8	\putStanText 15
\articlefile 8, 16	\input {Cze-vars.tex} 8	\quotedby {<autor>} 16
\begcenter 12	\insertBot 24	\re 22
\bigskip 13	\insertCite 16–17	\reduceref 22–23
\bmark 8, 10	\insertSpanImage 26	\ref 15
\BookException 10	\insertSpanText 26	\renum 19
\BookPost 10	\introfile 8	\rlap 27
\BookPre 10	\l 27	\schaptit{<text>} 12
\BookTitle 10	\llap 27	\swapCites 17
\bye 8	\load[opbible] 7	\switch 9, 19
\ChapterPost 9	\medskip 13	\timeline 26
\ChapterPre 9	\nl 16	\timelinewidth 26
\chaptit{<text>} 12	\nochapbooks 11, 21	\tline 27
\checksyntax 27–28	\noreduceref 23	\tlines 27
\Cite 16	\normalchapnumbers 27	\tlput 27
\cnvtext 12	\Note 14–15, 18, 27	\tmark 7–8, 19
\cr 27	\notesfile 8	\town 25
\cref 23	\notracinglinks 23	\townparams 24
\cslang 7	\pg 15	\tracinglinks 23
\endcenter 12	\pglabel 23	\tracingouterlinks 23
\endgraf 12	\pgref 23	\txsfile 8
\filegen 28	\picdir 15	\variants 9, 18
\fmtadd 12	\printedbooks 8, 10, 23	\vdef 9, 17
\fmtfile 8	\processbooks 8, 28	\vskip 27
\fmtfont 12	\putArticle 15–16	\ww 18
\fmtins 12	\putCite 16	\x 17
\fmtpoetry 13	\putCute 15	
\fmtpre 12	\putImage 15	