

Zpravodaj Československého sdružení uživatelů TeXu

Michal Hoftich

Elektronické knihy a systém TeX4ebook

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 26 (2016), No. 1-4, 94–105

Persistent URL: <http://dml.cz/dmlcz/150249>

Terms of use:

© Československé sdružení uživatelů TeXu, 2016

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ*:
The Czech Digital Mathematics Library <http://dml.cz>

Článek pojednává o konverzi dokumentů vytvořených v T_EXu do formátů elektronických knih, především Epub 3. Na konkrétních příkladech jsou ukázány možnosti uzpůsobení průběhu konverze a vzhledu výsledného e-booku.

Klíčová slova: e-knihy, TeX4ht, HTML, konverze.

Elektronické knihy

Elektronické knihy (e-booky) se rozvíjejí především od 70. let 20. století, kdy byl spuštěn projekt Gutenberg, jehož cílem je digitalizace knih a textů, na které se nevztahují autorská práva. Prudký rozvoj nastal od 90. let, kdy se objevily osobní digitální asistenty (PDA) a masověji se rozšířily osobní počítače. Zpočátku je tvořili především nadšenci, komerční vydavatelství pouze pomalu ztrácela nedůvěru. V roce 2004 se objevila první čtečka s displejem s elektronickým inkoustem (e-ink), o tři roky později internetový obchod Amazon představil svojí čtečku Kindle a nastal velký komerční rozvoj.

Hlavními vlastnostmi e-booků jsou flexibilita zobrazení, neboť je třeba, aby šly zobrazit na řadě zařízení s různou velikostí a vlastnostmi displeje, možnost upravit vlastnosti zobrazení, jako je velikost nebo barva písma podle potřeb konkrétního čtenáře. Některé čtečky také umožňují předčítání textu technologií text to speech.

V průběhu vývoje se objevilo množství formátů e-booků, v dnešní době jsou nejrozšířenějšími Epub, z něho vycházející Epub 3 a Mobi, používaný Amazonem. Tyto formáty v podstatě vychází z formátů HTML a CSS, běžně používaných na WWW stránkách, a pomocných metadat. Formát Mobi se vytváří kompilací Epubu programem Kindlegen, který Amazon poskytuje pro Windows, Mac OS i Linux.

Výhodou Epubu je jeho široká podpora nejrozličnější škálou zařízení a programů, které ovšem často podporují pouze některé jeho vlastnosti, často mají problém například s CSS styly, takže formátování e-booku může být zcela rozdílné, než autor zamýšlel. Nejlepší využití Epubu je pro jednoduchý text bez složitějšího formátování, například pro beletrii nebo literaturu faktu.

Totéž platí v podstatě i o formátu Mobi, který vzniká konverzí z Epub. Vlastně se jedná o dva formáty v jednom, neboť Kindlegen vytváří verzi pro novější i starší verzi svých zařízení. Amazon pro autory připravil dokument *Kindle design*

guidelines, ve kterém konkrétně specifikuje, jakým způsobem by měl být zdrojový Epub soubor připraven, aby konverze proběhla nejlepším způsobem.

Protože formát Epub vznikal živelně a jeho implementace ve čtecích zařízeních je nejednotná, organizace IDPF (IDPF, 2016) vytvořila standard Epub 3. Ten by měl mimo jiné zlepšit podporu pro technické publikace nebo pro asistenční služby pro čtenáře se specifickými potřebami. Tento formát bude tedy pro uživatele \TeX u patrně nejzajímavější, ovšem je třeba zmínit, že přestože je tento formát standardizovaný, čtecí zařízení zdaleka nepodporují všechny jeho vlastnosti. V současné době vzniká jeho nová verze, Epub 3.1, která obsahuje opět poměrně zásadní změny, ovšem zatím nebyla představena finální verze, proto se jím nebudeme dále zabývat.

Užitečnými vlastnostmi Epub 3 pro technické publikace jsou podpora MathML pro sazbu matematiky,¹ podpora SVG pro vektorové obrázky, podpora JavaScriptu, která umožňuje tvorbu interaktivních prvků, vkládání fontů do dokumentu. Významnou změnou jsou nové HTML elementy pro značkování logické struktury dokumentu (například poznámky pod čarou, bibliografie, rejstřík). Formát SMIL umožňuje vkládání multimediálního obsahu.

Patrně nejzajímavější ukázkou možností Epub 3, která v současnosti existuje, je projekt Aeneas (PETARIN, 2016), který umožňuje synchronizovat čtenou verzi knihy s jejím textem. Takto připravené e-booky lze číst v aplikaci, Mene-strello (PETARIN, 2016), která zvýrazňuje právě předčítaný text. Je to vynikající pomůcka například pro procvičování výuky jazyků.

Konverze \TeX u na e-book

Kvůli stále rozšířenějšímu čtení textů z elektronických zařízení, ať již jsou to PC, mobilní telefony, tablety nebo e-ink čtečky, vzniká potřeba konverze z \TeX u do e-booků. Všechny zmíněné formáty jsou v podstatě balíčky WWW stránek s přidanými metadaty, takže tato úloha se v podstatě skládá z konverze z \TeX u do HTML, přidání nezbytných metadat a zabalení výsledných souborů do e-bookového formátu.

Pro konverzi \TeX u do HTML se dá použít řada nástrojů, nejrozšířenějšími jsou LaTeX2HTML (DRAKOS, 2016), LaTeXML (MILLER, 2016), Pandoc (MACFARLANE, 2016) a \TeX 4ht (GURARI, 1997). První tři z nich jsou založené na zpracování \TeX ového dokumentu externím programem, který se snaží rozpoznat použitá makra a převést je na HTML. LaTeXML je z nich nejpokročilejší, neboť emuluje \TeX ovou expanzi maker, a podporuje tudíž i uživatelská makra a balíčky. Zbývající dva systémy podporují především základní příkazy \LaTeX u

¹V ostatních formátech je třeba používat obrázky, což je především pro matematiku vloženou v textu nevhodné řešení.

a se složitějšími dokumenty mohou mít problém. Všechny podporují pouze \LaTeX , jiné formáty nikoli.

\TeX4ht je sada maker, která se automaticky načítá do dokumentu a upravuje definici jednotlivých příkazů tak, aby bylo možné vkládat formátovací značky podporovaných výstupních formátů (kromě HTML je to především OpenDocument Format, TEI nebo Docbook). Díky tomu, že pro překlad je používán samotný \TeX , neměl by být problém s užíváním uživatelských maker a balíčků. Kromě \LaTeX u podporuje i jiné formáty, například Plain, ovšem u nich může být třeba upravit dokument. Pro překlad je možné použít současné nepoužívanější \TeX ové enginy, tedy \pdfTeX , \XeTeX a \LuaTeX . Systém vytvářel Eitan Gurari, po jeho smrti vývoj převzal Karl Berry, CV Radhakrishnan a autor článku. Právě \TeX4ht je použit jako základ pro \TeX4ebook (HOFTICH, 2016A), systém pro konverzi \TeX u na e-booky.

\TeX4ebook

\TeX4ebook je systém pro konverzi \TeX u do formátů Epub, Epub 3 a Mobi, hlavní důraz je kladený na Epub 3. Hlavní výhodou oproti pouhé konverzi \TeX u do HTML a jeho následné konverzi na e-book externími nástroji je možnost získání různých metadat přímo ze zpracovávaného dokumentu. Systém emuluje základní postup kompilace používaný \TeX4ht a přidává několik kroků navíc. Celý postup je konfigurovatelný s použitím sestavovacího skriptu.

Základní postup je shodný s \TeX4ht a sestává ze tří kroků:

1. Kompilace dokumentu \TeX em. Ještě před načtením dokumentu je nahrán balíček `tex4ht.sty`, který řídí další postup zpracování dokumentu. Pro každý načtený balíček se testuje existence souboru s příponou `.4ht`, který, pokud existuje, je načten po skončení preamble dokumentu. V tomto konfiguračním souboru se mohou předefinovávat makra a vkládají se do nich takzvané *háčky*. Po nahrání všech `.4ht` souborů pro použité balíčky jsou nahrány soubory s konfigurací pro daný výstupní formát. Obsahují instrukce, které jsou vloženy do háček definovaných v předešlém kroku. Háčky je také možné konfigurovat pomocí konfiguračního souboru, který bude popsán dále.

Kompilace je spouštěna několikrát, protože jsou využívány pomocné soubory pro tvorbu hypertextových odkazů. Obvykle jsou třeba tři kompilace, aby všechny odkazy fungovaly správně, ve výjimečných případech může být třeba i více kompilací.

2. Zpracování DVI souboru příkazem `tex4ht`. Tento příkaz vytváří výstupní soubory, převádí různá vstupní kódování na UTF-8 a vytváří dva pomocné soubory: `.idv` soubor je speciální soubor ve formátu DVI, který obsahuje stránky, které mají být převedeny na obrázky. Jedná se například o prostředí

picture L^AT_EXu, nebo matematiku, pokud není převáděna do formátu MathML. soubor `.lg` obsahuje seznam výstupních souborů, CSS instrukce a instrukce pro kompilaci jednotlivých stránek v `.idv` souboru na obrázky.

3. Zpracování `.lg` souboru příkazem `t4ht`. V tomto kroku je vytvořen CSS soubor, dochází ke konverzi obrázků a výstupní soubory mohou být předány externím programům k dalšímu zpracování.

T_EX4ht používá pro kompilaci řadu skriptů, které se liší použitým T_EXovým formátem, enginem a výstupním formátem. Všechny provádí tři kompilační kroky popsané výše a předávají jednotlivým příkazům argumenty v uvozovkách. Například pro vytvoření dokumentu ve formátu XHTML a kódování UTF-8 je možné použít následující příkaz:

```
htlatex filename.tex "xhtml,charset=utf-8" " -utf8 -cunihtf"
```

Tento postup není úplně uživatelsky přívětivý, proto T_EX4ebook umožňuje použít pro základní konfiguraci přepínače, známé z běžných příkazů užívaných například v Linuxu.

```
tex4ebook -f epub3 filename.tex
```

Tento příkaz vytvoří soubor `filename.epub` ve formátu Epub 3, kódování UTF-8 je použito automaticky.

Hlavní rozdíl T_EX4ebook oproti T_EX4ht spočívá ve třetím kompilačním kroku. Příkaz `t4ht` je použit pouze pro vytvoření CSS souboru, konverzi obrázků a spouštění externích příkazů řídí T_EX4ebook sám. Kromě toho je možné spouštět příkazy mezi jednotlivými kompilacemi T_EXu, například po první kompilaci a tvorbě pomocných souborů je možné spustit příkazy pro tvorbu rejstříku a bibliografie.

Postup kompilace lze řídit pomocí sestavovacích skriptů Make4ht (HOFTICH, 2016B). To je sestavovací program pro T_EX4ht, který původně vznikl jako součást T_EX4ebook, ale postupně se vyvinul do samostatné aplikace a knihovny, která nahrazuje skripty používané pro kompilaci T_EX4ht.

Sestavovací skripty jsou programy v jazyce Lua:

```
local filter = require "make4ht-filter"
local process = filter{"hruletohr"}
Make:add("biber","biber ${input}")
Make:htlatex{}
Make:biber {}
Make:htlatex {}
Make:image("png$",
"dvipng -bg Transparent -T tight -o ${output}"..
"-pp ${page} ${source}")
Make:match("html$",process)
Make:match("html$",
"tidy -m -utf8 -asxhtml -q -i ${filename}")
```

Pomocí příkazu `Make:add("biber", "biber ${input}")` definujeme příkaz `Make:biber`, který můžeme použít pro tvorbu bibliografie s balíčkem Biblatex. Druhý argument příkazu `Make:add` je šablona s příkazem, který se spustí. Proměnné definované `Make4ht` se vkládají pomocí konstruktu `${jméno}`, proměnná `input` obsahuje název zpracovávaného souboru.

Argumenty ve složených závorkách pro příkazy `Make:htlatex` a `Make:biber` jsou tabulky, ve kterých můžete nastavit hodnoty proměnných užívaných v šablonách, pokud byste např. chtěli použít bibliografii z jiného souboru, lze použít: `Make:biber {input = "jinysoubor"}`

Příkazy `Make:image` a `Make:match` se spouští na výstupních souborech. Jejich prvním argument je regulární výraz, kterým se testují jména souborů, druhým je příkaz, který se spustí, pokud jméno souboru vyhovuje hledanému vzoru. Příkaz může být jednak šablona s programem, jednak Lua funkce.

Při konverzi obrázků se používají proměnné, které nejsou k dispozici v kompilačních příkazech, konkrétně `source` obsahuje jméno `.idv` souboru, `output` je název obrázku a `page` udává číslo stránky v `.idv` souboru.

Příkaz `Make:match` umožňuje měnit výstupní soubory. Proměnná `filename` obsahuje název výstupního souboru. V našem případě je tento příkaz spuštěn dvakrát, první příklad ukazuje použití filtrů, které jsou funkce v jazyce Lua. V tomto případě se jedná o filtr, který napravuje problém způsobený příkazem `\hrule`, jenž `TEX4ht` nemůže redefinovat, a v HTML souboru se projeví jako řada znaků `_`. Jako argument pro příkaz `filter` lze zadat více filtrů, kromě předdefinovaných, jako je `hruletohr`, lze zadat i vlastní funkce:

```
local changea = function(s) return s:gsub("a","z") end
local process = filter{"hruletohr", changea}
```

Tento příklad není úplně užitečný, neboť změni všechna písmena „a“ ve zpracovávaném dokumentu na „z“, včetně HTML tagů, ovšem jako ukázka snad postačí.

Pokud sestavovací soubor pojmenujete stejně jako název dokumentu s příponou `.mk4`, bude spuštěn automaticky, jinak ho lze zadat jako parametr `-e` pro `TEX4ebook`:

```
tex4ebook -f epub3 -e buildfile.mk4 filename.tex
```

Konfigurační soubor pro T_EX4ht

Pro vložení vlastních HTML tagů do konfigurovatelných háčků nebo vlastních CSS instrukcí můžeme použít konfigurační soubor pro `TEX4ht`. Možnosti konfigurace si ukážeme na jednoduchém `TEX`ovém souboru:

```
\documentclass{article}
\usepackage[T1]{fontenc}
```

```

\usepackage[utf8]{inputenc}
\usepackage[czech]{babel}
\begin{document}
Příliš žluťoučký kůň \textit{úpěl} \textbf{ďábelské ódy}
\end{document}

```

Po kompilaci se vygeneruje HTML soubor s následujícím obsahem:

```

<p class="noindent" >Příliš žluťoučký kůň <span
class="ecti-1000">úp</span><span
class="ecti-1000">ěl </span><span
class="ecbx-1000">ď</span><span
class="ecbx-1000">ábelsk</span><span
class="ecbx-1000">é </span><span
class="ecbx-1000">ódy </span>

```

Vygenerované soubory jsou ponechány i v adresáři s $\text{T}_{\text{E}}\text{X}$ ovým dokumentem, lze je tedy snadno zkontrolovat ve WWW prohlížeči. Slova, která byla zvýrazněna jiným než základním řezem písma, byla rozdělena do několika HTML elementů. Ve vykresleném dokumentu se text zobrazí v pořádku, nicméně měli bychom tento problém vyřešit. Jedná se o chybu při konverzi DVI souboru příkazem `tex4ht`, který se snaží přidat formátování podle písma detekovaného v DVI souboru. Díky této funkcionalitě není třeba vytvářet konfigurace pro všechna uživatelská makra, základní textové formátování, jako jsou řezy a velikost písma, je zachováno. Pokud však `TeX4ht` narazí na znak s diakritikou, uzavře aktuální element, přestože nedochází ke změně písma. Tento problém lze vyřešit jednoduchou konfigurací pro dotčená makra v konfiguračním souboru pojmenovaném například `myconfig.cfg`:

```

\Preamble{xhtml}
\Configure{textbf}{\NoFonts\HCode{<strong>}}
{\HCode{</strong>}\EndNoFonts}
\Configure{textit}{\NoFonts\HCode{<em>}}
{\HCode{</em>}\EndNoFonts}
\begin{document}
\EndPreamble

```

Konfigurační soubor můžeme načíst volbou `-c`:

```
tex4ebook -f epub3 -e buildfile.mk4 -c myconfig.cfg filename.tex
```

Výsledný HTML kód již vypadá mnohem lépe:

```

<p class="noindent" >Příliš žluťoučký kůň
<em>úpěl</em> <strong>ďábelské ódy</strong>

```

Konfigurační soubory pro `TeX4ht` mají pevnou základní strukturu:

```

% zde můžeme vkládat balíčky
\Preamble{xhtml,volby pro tex4ht.sty}
..

```

```
% Konfigurace háčků a kaskádových stylů
\begin{document}
..
\EndPreamble
```

Před příkazem `\Preamble` můžeme nahrávat dodatečné balíčky pomocí příkazu `\RequirePackage`. Příkaz `\Preamble` umožňuje specifikovat volby, pro které ovlivňují nahrávání konfigurací. Jsou oddělené čárkami a povinná je minimálně volba `html` nebo `xhtml`, která musí být na první pozici. Pro Epub je třeba používat druhou z nich.

Po příkazu `\Preamble` můžete přidávat konfigurace pro háčky, kaskádové styly nebo měnit definice příkazů definovaných v preambuli dokumentu, ať již ve vkládaných balíčcích nebo přímo v dokumentu. Použití `\begin{document}` je povinné, definice, které po něm následují, jsou spuštěny až na začátku dokumentu. Konfigurace končí příkazem `\EndPreamble`.

V našem příkladu užíváme dvě konfigurace pro příkazy `\textbf` a `\textit`. Příkaz `\Configure` má variabilní počet argumentů, pro jednotlivé konfigurace je třeba konzultovat dokumentaci².

V tomto konkrétním případě užívá příkaz `\Configure` dva argumenty, jedná se o kód, který se vloží na začátek a na konec konfigurovaného textu. Příkazy `\NoFonts` a `\EndNoFonts` zakážou a povolí zpracování fontů z DVI souboru, čímž vyřešíme problém s rozděleními slovy. Příkaz `\HCode` vkládá HTML kód do dokumentu.

Výsledný e-book vypadá poměrně stroze, pokud chcete vylepšit jeho vzhled, je možné použít kaskádové styly. Pro jednoduché úpravy je určen příkaz `\Css`, který lze použít v konfiguračním souboru:

```
\Preamble{xhtml}
% konfigurace písem vynechána
\Css{em{color:blue;}}
\begin{document}
```

Tímto příkazem obarvíme text v kurzívě modrou barvou.

Pro složitější změny je lepší vložit kompletní CSS styl. Existuje řada projektů stylů, které se zaměřují na dobrou typografii na webu. Můžeme využít například `Scale.css` (SALMINEN, 2012), který vytvořil finský designér Viljami Salminen. Tento styl je responzivní, což znamená, že velikost jednotlivých prvků dokumentu se přizpůsobuje velikosti zobrazované plochy tak, aby byla vždy optimální z hlediska dobré čitelnosti.

²Dokumentace není úplně silnou stránkou projektu `TEX4ht`. Základní informace o konfiguraci se nachází na WWW stránkách projektu, detailnější informace můžete získat z dokumentace vygenerované ze zdrojových kódů `TEX4ht` umístěných na stránkách <http://michal-h21.github.io/src4ht/>. Jedná se především o dokumenty `tex4ht-info` a `tex4ht-html4`.

Pro vložení externího stylu můžeme využít příkaz `\AddCss`, který poskytuje balíček `include4ht`. Tento balíček je jeden z těch, které tvoří projekt `helpers4ht`. Jeho cílem je zjednodušit práci se systémem `TEX4ht` a obsahuje balíčky pro vkládání kaskádových stylů, JavaScriptových programů, tvorbu rejstříků a další. Balíček pro vkládání fontů můžeme použít pro vložení fontů do našeho e-booku.

Epub 3 podporuje vkládání písem ve formátu OpenType nebo jeho zjednodušené verzi určené pro web, WOFF. Je vhodné využívat OpenSource písma, vložním komerčního písma bychom se mohli dopustit jeho ilegálního šíření. Vhodná písma jsou například EB Garamond, Linux Libertine nebo Latin Modern. Pro každý řez písma je třeba stáhnout odpovídající OpenType soubor a umístit ho do adresáře s dokumentem, stejně tak je třeba umístit zde externí kaskádové styly. Upravený konfigurační soubor:

```
\RequirePackage{include4ht}
\RequirePackage{addfont4ht}
\Preamble{xhtml}
\AddCss{scale.css}
..
\NormalFont{EBGaramond}{EBGaramond12-Regular.woff}
\BoldFont{EBGaramond}{EBGaramond12-Italic.woff}
\ItalicFont{EBGaramond}{EBGaramond12-Italic.woff}
\Configure{@HEAD}{\HCode{<style type='text/css' >
  \Hnewline body{font-family:rmfamily,
    "EBGaramond", sans-serif;}\Hnewline
  </style>}}
\begin{document}
\EndPreamble
```

Příkazy `\NormalFont`, `\BoldFont` a `\ItalicFont` vloží fonty. První parametr určuje jméno rodiny, na které se dále odkazujeme. Příkaz `\Configure{@HEAD}` vkládá kód do hlavičky HTML dokumentu. V tomto případě je použit pro vložení kaskádového stylu, který vybírá námi definovanou rodinu *EBGaramond* jako hlavní písmo dokumentu.

Matematika a e-booky

Velký problém e-booků je matematika, stejně jako na webu, kde zobrazování matematiky řeší formát MathML. Alespoň teoreticky, protože jeho podpora ve WWW prohlížečích je špatná, všechny ho podporují pouze částečně.

Formát MathML je součástí specifikace Epub 3, teoreticky by ho tedy měla čtecí zařízení podporovat, realita je ovšem odlišná³. Pokud MathML podporují,

³Tabulku shrnující podporu v různých čtecích zařízeních lze nalézt na stránkách knihovny MathJax (THE MATHJAX CONSORTIUM, 2016).

je to za pomoci knihovny MathJax. Ta dokáže výborně vykreslit matematiku na WWW stránkách, ovšem je poměrně náročná na výkon a na tabletech nebo mobilních telefonech trvá delší chvíli, než se stránky obsahující matematiku vykreslí.

V případě ostatních formátů je situace ještě horší, v případě Epub i Mobi lze matematiku zobrazit pouze za pomoci obrázků, což pro $\text{T}_{\text{E}}\text{X}4\text{ht}$ není problém, naopak je to jeho výchozí nastavení. Problémem je, že tyto obrázky nebudou sedět na účaří a velikost a typ písma nebudou odpovídat okolnímu textu. Zatím jsem nenalezl uspokojivé a univerzální řešení těchto problémů, proto doporučuji pro matematické e-booky používat Epub 3 s podporou MathML, jehož podpora ve čtecích zařízeních se může zlepšit.

Podpora MathML se může zapnout pomocí přepínače `mathml` umístěného v příkazu `\Preamble` v konfiguračním souboru nebo ve druhém argumentu příkazu `tex4ebook`:

```
tex4ebook -f epub3 filename mathml
```

Konfigurace vlastních balíčků

Příkazem `\Configure` můžeme konfigurovat pouze příkazy, do kterých byly pro toto použití vloženy háčky. Pokud chceme přidat podporu pro nový balíček, je třeba vytvořit pro něj `.4ht` soubor. Jeho tvorbu si ukážeme na jednoduchém příkladu.

Mějme balíček pro tvorbu vzdělávacích dokumentů, který umožňuje tvorbu dvou verzí dokumentu – pro studenty a pro učitele, s poznámkami, které se ve studentské verzi nezobrazují. Balíček bude mít přepínač, kterým se bude vybírat mezi zobrazeními, pro každou verzi bude vytvořen soubor, který bude sloužit pro kompilaci a který bude vkládat samotný text dokumentu.

Budeme také využívat rozšířených metadat, která poskytuje formát Epub 3. Vzniklo několik profilů, které specifikují metadata podle určení dokumentu. Pro tvorbu vzdělávacích dokumentů vznikl profil *Epub for Education* (THE EPUB WORKING GROUP, 2016), který využijeme v našem dokumentu.

Balíček `edupub.sty` je poměrně prostý:

```
\ProvidesPackage{edupub}
\RequirePackage{kvoptions}
\RequirePackage{etoolbox}
\newbool{teacher}
\boolfalse{teacher}
\newcommand\teacherinfo[1]{}
\DeclareVoidOption{teacher}{%
  \renewcommand\teacherinfo[1]{%
    \edupub@print@teacherinfo{##1}}
```

```

\booltrue{teacher}}
\newcommand\edupub@print@teacherinfo[1]{#1}
\ProcessKeyvalOptions*

Definuje pouze příkaz \teacherinfo, který vypíše poznámky pro učitele,
pokud je balíček nahrán s volbou teacher. Nyní vytvoříme konfigurační soubor
pro TEX4ht, edupub.4ht:

\ifbool{teacher}{%
  \Configure{OpfMetadata}
  {\HCode{<dc:type>teacher-edition</dc:type>}}
}{}%
\NewConfigure{teacherinfo}{2}

```

```

\let\old:teacherinfo\edupub@print@teacherinfo
\renewcommand\edupub@print@teacherinfo[1]{
  \a:teacherinfo
  \old:teacherinfo{#1}
  \b:teacherinfo
}
\Configure{teacherinfo}
{\HCode{<span epub:type="answer">}}
{\HCode{</span>}}

```

Konfigurace `\Configure{OpfMetadata}` vloží informace do souboru s meta-
daty pro dokument. V tomto případě označí verzi dokumentu pro učitele.

Příkaz `\NewConfigure` deklaruje konfigurační háčky, které se vkládají do
redefinovaných příkazů. První argument je název konfigurace, druhý je počet
háčků, které budou vytvořeny. Pokud chceme vkládat nějaký kód před pří-
kaz a po jeho skončení, stačí nám dva háčky. Háčky jsou příkazy ve formátu
`\[a-i]:název háčku`, v našem případě jsou vytvořeny dva, `\a:teacherinfo`
a `\b:teacherinfo`. Pomocí příkazu `\let` si můžeme uložit původní definici pří-
kazu, který chceme upravit, a poté použít například příkaz `\renewcommand` pro
úpravu definice.

Nakonec konfigurujeme naše háčky pomocí `\Configure{teacherinfo}`, po-
známky pro učitele označíme elementem `span` s atributem `epub:type`, který
Epub 3 používá pro sémantické značkování textu. Tento konkrétní typ je součástí
profilu *Epub for Education*.

Naše ukázková publikace tedy bude mít dvě verze, jednu pro studenty a druhou
pro učitele. Pro každou vytvoříme řídicí dokument, který bude obsahovat pouze
hlavičku dokumentu, samotný text se bude vkládat a bude pro obě verze stejný.

Řídicí soubor pro učitele `teacher.tex`:

```

\documentclass{article}
..

```

```
\usepackage[teacher]{edupub}
\begin{document}
\include{text}
\end{document}
```

Verze pro studenta bude totožná, pouze balíček `edupub` vložíme bez volby `teacher`. Samotný text dokumentu může mít následující obsah:

```
\newcommand\odpoved[1]{\teacherinfo{\ #1}}
\begin{enumerate}
\item Jak se jmenuje největší had
      vyskytující se v~České republice?
\odpoved{Užovka stromová}
\end{enumerate}
```

Závěr

Článek představil základní použití nástroje pro tvorbu elektronických knih \TeX -ebook, shrnul rozdíly mezi jednotlivými formáty elektronických knih a zaměřil se na jeden z nich, Epub 3. Tento formát je vhodný i pro publikaci technicky a matematicky zaměřených publikací, tudíž je zajímavý i pro uživatele \TeX u. \TeX 4ebook byl úspěšně použit pro tvorbu materiálů pro výuku matematiky i několika menšími vydavateli.

Samotný \TeX 4ebook je nadstavba pro \TeX 4ht; pro upravení vzhledu výsledného dokumentu je možné využít konfigurační nástroje, které \TeX 4ht nabízí. Více informací lze nalézt v článku původního autora \TeX 4ht (GURARI, 2004) nebo ve čtvrté kapitole knihy *L^AT_EX Web Companion* (GOOSENS A KOL., 1999).

Reference

- DRAKOS, NIKOS. *LaTeX2HTML* [on-line]. Ver. 98.1. 2016. Dostupné na: <http://www.latex2html.org/>.
- GOOSENS, MICHEL ET AL.. *The L^AT_EX Web Companion: Integrating TeX, HTML, and XML*. Boston : Addison-Wesley Professional, 1999. 560 s. ISBN 978-0201433111.
- GURARI, EITAN. *TeX4ht* [on-line]. Ver. 200. 1997. Dostupné na: <http://www.tug.org/tex4ht/>.
- GURARI, EITAN. \TeX 4ht: HTML Production [on-line]. *TUGboat*, 2004, vol. 25, no. 1., s. 39–47. [cit. 2016-14-10]. (ISSN 0896-3207.) Dostupné na: <https://www.tug.org/TUGboat/tb25-1/gurari.pdf>.
- HOFTICH, MICHAL. *TeX4ebook* [on-line]. Ver. 0.1d. 2016a. Dostupné na: <https://www.ctan.org/pkg/tex4ebook>.

- HOFTICH, MICHAL. *Make4ht* [on-line]. Ver. 0.1b. 2016b. Dostupné na: <https://www.ctan.org/pkg/make4ht>.
- MILLER, BRUCE. *LaTeXML* [on-line]. Ver. 0.8. 2016. Dostupné na: <http://dlmf.nist.gov/LaTeXML/>.
- MACFARLANE, JOHN. *Pandoc* [on-line]. Ver. 1.17. 2016. Dostupné na: <http://pandoc.org/>.
- PETTARIN, ALBERTO. *Aeneas* [on-line]. Ver. 1.6. 2016. Dostupné na: <https://www.readbeyond.it/aeneas/>.
- PETTARIN, ALBERTO. *Menestrello* [on-line]. Ver. 3.0.0. 2016. Dostupné na: <https://www.readbeyond.it/menestrello/index.html>.
- SALMINEN, VILJAMI. *Scale.css* [on-line]. Ver. 0.1. 2012. Dostupné na: <https://github.com/viljamis/Scale>.
- THE EPUB WORKING GROUP. *EPUB for Education* [on-line]. 2016. [cit. 2016-14-10]. Dostupné na: <http://www.idpf.org/epub/profiles/edu/spec/>.
- THE INTERNATIONAL DIGITAL PUBLISHING FORUM. *EPUB 3* [on-line]. 2016. [cit. 2016-14-10]. Dostupné na: <http://idpf.org/epub/30>.
- THE MATHJAX CONSORTIUM. *EPUB3 Reading systems overview* [on-line]. 2016. [cit. 2016-14-10]. Dostupné na: <http://docs.mathjax.org/en/latest/misc/epub.html>.

Summary: E-books and the T_EX4ebook System

This article describes the process of conversion of a T_EX document to e-book using T_EX4ebook system. Concrete examples of configuration and caveats are provided.

Keywords: e-books, TeX4ht, HTML, conversion.

Michal Hoftich, michal.h21@gmail.com