

Zpravodaj Československého sdružení uživatelů TeXu

Vít Novotný

Vysokoúrovňové jazyky pro TeX

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 32 (2022), No. 1-4, 35–48

Persistent URL: <http://dml.cz/dmlcz/151107>

Terms of use:

© Československé sdružení uživatelů TeXu, 2022

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>

T_EX je strojový kód světa digitální sazby, který od spisovatelů a grafiků vyžaduje netriviální programátorské dovednosti a programátorům poskytuje minimum vysokoúrovňových abstrakcí. V článku představuji vybrané značkovací, programovací a stylové jazyky pro T_EX, které umožňují dělbou práce mezi spisovatele, vývojáře a grafiky a usnadňují proces přípravy elektronických dokumentů. Článek je přepis mé přednášky na valném shromáždění ζ TUGu 14. května 2022 [1].

Klíčová slova: vysokoúrovňové jazyky, programovací jazyky, značkovací jazyky, stylové jazyky, ϵ -T_EX, pdfT_EX, LuaT_EX, LuaMetaT_EX, L^AT_EX 2 _{ϵ} , L^AT_EX3, expl3, XML, DocBook, TEI, XHTML, XSLT, CSS, CSL, ConT_EXt, HTML, markdown, YAML, Pandoc, TikZ, BibL^AT_EX, BibL^AT_EXML, LyLuaT_EX

Kreativnímu jedinci je T_EX pozvánkou k tomu, aby se na chvíli stal spisovatelem, grafikem, ilustrátorem, sazečem a vývojářem v jedné osobě. Rozsáhlým dokumentům však prospívá, když obsah, stylopisy a programová výbava vznikají do jisté míry nezávisle.¹ To umožňuje dělbou práce mezi doménové experty, aniž by neúnosně rostly náklady na vzájemnou koordinaci. Kreativec, který dokáže zastoupit několik rolí, se může v případě potřeby omezit jen na jednu z nich a věnovat jí svou plnou a ničím nerušenou pozornost.

T_EXový dokument můžeme rozdělit na obsah, stylopisy a programovou výbavu jednoduše tak, že vytvoříme tři samostatné soubory:

- text dokumentu označovaný T_EXovými makry
- stylopis s nastavením délkových registrů, písem, výstupní rutiny, apod.
- program s definicí značkovacích maker

Při dělbě práce ale můžeme narazit na to, že doménoví experti nejsou schopni nebo ochotni používat při své práci T_EX, což má několik důvodů:

¹Čtenář může namítnout, že rozdělení dokumentu na nezávislé části je fikce: Jsou obrázky v textu doménou spisovatele, nebo ilustrátora? Může spisovatel měnit písmo a barvu textu či způsob číslování seznamů, nebo jde o úkol grafika? Kde jsou hranice mezi rolmi grafika, sazeče a vývojáře?

Tyto námítky jsou na místě především u akcidenčních a nízkonákladových dokumentů, jako jsou plakáty, básnické sbírky nebo jídelní a nápojové lístky. U takových dokumentů může být vhodnější zvolit takový přístup, při kterém jsou role volnější a mezi účastníky procesu přípravy dokumentu probíhá soustavná komunikace. Autor článku má zkušenosti převážně s přípravou odborných a technických dokumentů, u kterých jsou jednotvárnost a kompozicionalita úmyslnými prvky stylu. Takové dokumenty lze snadno rozdělit na dílčí části bez újmy na celku.

- Moderní značkovací jazyky jako markdown minimalizují poměr značek vůči textu pro snadný zápis a zvýšenou čitelnost, zatímco $\text{T}_{\text{E}}\text{X}$ bývá při větším množství značek upovídaný a nepřehledný.²
- Moderní stylovací jazyky jako CSS jsou deklarativní a nepožadují programátorské dovednosti, zatímco $\text{T}_{\text{E}}\text{X}$ je imperativní programovací jazyk.
- Moderní programovací jazyky jako Python nabízí vysokoúrovňové abstrakce a bohaté základní knihovny vestavěných funkcí, zatímco $\text{T}_{\text{E}}\text{X}$ nabízí pouze primitivní datové typy a operace nad nimi.

Nicméně existuje mnoho vysokoúrovňových jazyků pro $\text{T}_{\text{E}}\text{X}$, na které se výše uvedená omezení nevztahují a které můžeme použít pro přípravu obsahu, stylolistů a programové výbavy namísto nízkoúrovňového $\text{T}_{\text{E}}\text{X}$ u.

V sekci 1 shrnuji základní pojmy související s $\text{T}_{\text{E}}\text{X}$ em, jako jsou stroje, makrobalíky a formáty. V sekcích 2 až 5 nabízím přehled programovacích, značkovacích, stylových a dalších doménově specifických vysokoúrovňových jazyků pro $\text{T}_{\text{E}}\text{X}$. V sekci 6 shrnuji poznatky z tohoto článku. V sekci 7 se zamýšlím nad dalším směřováním $\text{T}_{\text{E}}\text{X}$ u a vysokoúrovňových jazyků jako takových.

1. Přehled základních pojmů

$\text{T}_{\text{E}}\text{X}$ je nízkoúrovňový programovací jazyk pro digitální sazbu [2]. Referenční implementací $\text{T}_{\text{E}}\text{X}$ u je *stroj* $\text{T}_{\text{E}}\text{X}90$ od Donalda Knutha [3]. Moderní $\text{T}_{\text{E}}\text{X}$ ové stroje jako $\varepsilon\text{-T}_{\text{E}}\text{X}$ [4], $\text{pdfT}_{\text{E}}\text{X}$ [5] a $\text{LuaT}_{\text{E}}\text{X}$ [6] rozšiřují $\text{T}_{\text{E}}\text{X}90$ o dodatečně primitivní příkazy, které zvyšují vývojářský komfort.

Makrobalíky jako plain [7], $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ [8] a $\text{ConT}_{\text{E}}\text{Xt}$ [9] staví z primitivních příkazů $\text{T}_{\text{E}}\text{X}$ ových strojů vysokoúrovňové značkovací a programovací jazyky pro spisovatele a vývojáře. *Formáty* odpovídají kombinaci stroje a makrobalíku, např. $\text{LuaT}_{\text{E}}\text{X} + \text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} = \text{LuaL}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ³ a $\text{LuaT}_{\text{E}}\text{X} + \text{ConT}_{\text{E}}\text{Xt} = \text{ConT}_{\text{E}}\text{Xt MkIV}$. Pomocí formátů připravuje sazeč dokumenty z příkazové řádky s příkazy jako `lualatex` a `context`.

2. Programovací jazyky pro vývojáře

V této sekci popisují, jaké možnosti nabízí moderní $\text{T}_{\text{E}}\text{X}$ ové stroje a makrobalíky vývojářům. Veškeré ukázky v této sekci vypíší text $1 + 2 = 3$.

²Čitelnost textu označovaného v $\text{T}_{\text{E}}\text{X}$ u se zlepší, pokud značky sestavíme z interpunkčních znamének. Pokud se však pro usnadnění zápisu omezíme na znakovou sadu ASCII, skončíme s pouhými 32 znaky, z nichž 10 už je využíváno $\text{T}_{\text{E}}\text{X}$ em. Při větším množství značek by proto jednotlivá interpunkční znaménka odpovídala několika různým značkám. Vyřešení nejednoznačností by vyžadovalo buďto složitou lexikální a syntaktickou analýzu textu, pro kterou $\text{T}_{\text{E}}\text{X}$ nemá vhodné primitivní datové typy ani příkazy, nebo bychom museli interpunkci ve značkách použít způsobem, který usnadňuje analýzu na úkor čitelnosti. Proto se při větším množství značek typicky místo interpunkce používají upovídaná $\text{T}_{\text{E}}\text{X}$ ová makra.

³Většina $\text{T}_{\text{E}}\text{X}$ ových formátů poskytuje vlastní značkovací jazyky. Formát $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X} 2_{\varepsilon}$ je však zdaleka nejznámější a také nejdůslednější v odstínění spisovatele od programování a stylování.

Stroj ε -TeX a jeho následovníci jako pdfTeX a LuaTeX nabízí primitivní příkaz `\numexpr`, který vyhodnocuje celočíselné aritmetické výrazy:

```
$ 1 + 2 = \numexpr 1 + 2 \relax $
```

Příkaz `\numexpr` zvyšuje komfort oproti primitivním příkazům TeXu 90:

```
$ 1 + 2 = \newcount\x \x=1 \advance\x by 2 \the\x $
```

Pro další primitivní typy má ε -TeX příkazy `\dimexpr`, `\glueexpr` a `\muglueexpr`.

Stroje LuaTeX a LuaMetaTeX [10] nabízí primitivní příkaz `\directlua`, který umožňuje zadávat a spouštět programy v jazyce Lua:

```
$ 1 + 2 = \directlua { tex.print(1 + 2) } $
```

Kromě základní knihovny jazyka Lua mohou vývojáři interagovat s TeXovým strojem a instalací TeXu [6, kapitoly 5–10; 10, kapitoly 4–10] a využívat rozšiřující softwarové knihovny pro práci se soubory a zpracování textu [6, sekce 4.3].

Stroje pdfTeX a LuaTeX rozšiřují primitivní příkaz `\input` o variantu, která spouští libovolné programy pomocí příkazové řádky operačního systému:

```
$ 1 + 2 = \input|" echo 1 + 2 | bc " $
```

Rozšířená varianta příkazu `\input` umožňuje integrovat TeXový kód s širším ekosystémem programové výbavy mimo instalaci TeXu.⁴

Součástí experimentálního formátu L^ATeX3 je makrobalík *expl3-generic*, který poskytuje vysokoúrovňový programovací jazyk `expl3` [11; 12; 13]:

```
\input expl3-generic
\ExplSyntaxOn
$ 1 + 2 = \int_eval:n { 1 + 2 } $
\ExplSyntaxOff
```

`Expl3` nabízí vysokoúrovňové datové typy pro obecné programování (seznamy a hašové tabulky) a typografické programování (rakve⁵ a barvy) a bohatou základní knihovnu funkcí pro řízení toku programu a TeXové expanze, celočíselnou i reálnou aritmetiku a zpracování TeXových tokenů a unicodového textu. Jazyk `expl3` využívá primitivní příkazy strojů TeX90, ε -TeX a pdfTeX; funguje ale i s novějšími stroji, jako jsou XeTeX, LuaTeX a LuaMetaTeX [12, sekce 9].

⁴Jednou z nevýhod rozšířené varianty příkazu `\input` je vazba na konkrétní příkazovou řádku a programovou výbavu, což snižuje přenositelnost dokumentů. V našem příkladu se jedná o příkazovou řádku Bourne shell z UNIXU V7 (také `sh`), případně o zpětně kompatibilní nadstavby jako `bash`, `dash` a `ksh`, a o unixovou kalkulačku `Bench calculator` (`bc`). Většina Linuxových distribucí ale používá příkazovou řádku kompatibilní s `sh` a zahrnuje `bc` v základní programové výbavě; náš příklad na nich tedy můžete bez úpravy vysázet.

Další nevýhodou rozšířené varianty příkazu `\input` je skutečnost, že představuje bezpečnostní riziko a uživatel ji proto musí explicitně povolit parametrem `-shell-escape`. Například pro vysázení našeho příkladu musíme použít příkaz `pdftex -shell-escape <jméno dokumentu>`.

⁵Rakev (z anglického *coffin*) sestává z TeXového boxu, informací o jeho tvaru a rozměrech a množiny vertikálních a horizontálních přímk. Průsečíky přímk tvoří úchyty rakve, které slouží k vzájemnému pozicování rakví na stránce.

Vývojáře zvyklé na kategorie znaků v konvenčních formátech jako plain \TeX , \LaTeX a Con \TeX t mohou u explu překvapit podtržítka a dvojtečky s kategorií písmene a bílé znaky ($_$, $_$) s kategorií ignorovaného znaku. Studie však ukazují, že oddělování slov v identifikátorech pomocí `_podtržitek` je čitelnější než oddělování slov pomocí Kapitálek [14], mezery jsou častým zdrojem chyb při programování v \TeX u a dvojtečky v explu slouží pro oddělení názvu příkazu (`\int_eval`) od jeho typové signatury (`:n`).⁶ Kategorie znaků v explu jsou tedy účelné.

3. Značkovací jazyky pro spisovatele

V této sekci se podíváme na některé existující značkovací jazyky a možnosti jejich využití v \TeX u.

Je překvapivě obtížné přesvědčit o tom uživatele, ale nedostatky \LaTeX u pro koncentraci, psaní a myšlení si v ničem nezadají s nedostatky Wordu. Je to z toho prostého důvodu, že \LaTeX dává spisovateli do rukou příliš velkou moc: Vždy existuje další makrobalík, který můžeme zavést v preambuli, stejně jako vždy existuje další rozbalovací nabídka ve Wordu. — Thompson [15, v překladu autora]

Formát $\LaTeX 2_{\epsilon}$ poskytuje jednoduchý značkovací jazyk pro přípravu dokumentů, který je možné rozšiřovat pomocí makrobalíků:⁷

```
\documentclass{book}
\usepackage[czech]{babel}
\title{Ukázkový dokument v~\LaTeX u}
\author{Vít Novotný}
\date{14. května 2022}
\begin{document}
\maketitle
\chapter{Kapitola}
Ahoj, \LaTeX u!
\end{document}
```

⁶Typové signatury umožňují definovat příkazy s jiným typem argumentu, než s jakým příkazy voláme. Můžeme např. zadefinovat příkaz `\pozdrav:n`, který bude přijímat jeden argument s \TeX ovými tokeny:

```
\cs_new:Nn \pozdrav:n { Ahoj,~#1! }
```

Při volání příkazu pro nás ale může být snazší použít jako argument např. jméno proměnné:

```
\cs_generate_variant:Nn \pozdrav:n { v }
\tl_new:N \g_jmeno_tl
\tl_set:Nn \g_jmeno_tl { světe }
\pozdrav:v { g_jmeno_tl } % Expanduje na \pozdrav:n{světe} a poté na Ahoj, světe!
```

Díky typovým signaturám může expl3 automaticky přetypovat argumenty, což zvyšuje komfort.

⁷Většina \TeX ových formátů poskytuje vlastní značkovací jazyky. Formát $\LaTeX 2_{\epsilon}$ je však zdaleka nejznámější a také nejdůslednější v odstínění spisovatele od programování a stylování.

Pokud nejsme spokojeni s automatickým výstupem vysokoúrovňových značek jako `\chapter`, můžeme místo nich použít stylovací příkazy L^AT_EXu jako `\textbf` a primitivní příkazy T_EXového stroje jako `\vskip`. Toto může být vhodné pro řešení konkrétních typografických nedostatků, které nelze řešit systémově. Nadměrné užívání nízkourovňových příkazů ale narušuje dělbů práce a vede k nejednotnému vzhledu koncového dokumentu.

Mimo svět T_EXu jsou oblíbené značkovací jazyky založené na meta-jazyku XML. Můžeme si navrhnout buďto svůj vlastní XML jazyk [16] nebo využít existující jazyk jako DocBook, TEI, nebo XHTML (vizte níže):

```
<?xml version="1.0" encoding="utf-8"?>
<html xml:lang="cs" xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Ukázkový dokument v XHTML</title>
    <meta name="author" content="Vít Novotný" />
  </head>
  <body>
    <h1>Kapitola</h1>
    <p>Ahoj, XHTML!</p>
  </body>
</html>
```

Pro zpracování tohoto dokumentu T_EXem požádáme o pomoc vývojáře. Ten buďto využije vestavěnou podporu XML v pokročilých T_EXových formátech jako ConT_EXt [17; 18] nebo připraví program v jazyce XSLT:

```
<?xml version="1.0" encoding="utf-8"?>
<stylesheet xmlns="http://www.w3.org/1999/XSL/Transform"
  xmlns:xhtml="http://www.w3.org/1999/xhtml"
  version="1.0">
  <output method="text" />

  <!-- Následující část zpracovává prvek <head> -->
  <template match="xhtml:head">
    \documentclass{book}
    \usepackage[czech]{babel}
    <!-- Zpracuj zanořené prvky <title> a <meta> -->
    <apply-templates select="*" />
  </template>

  <template match="xhtml:title">
    \title{<value-of select="text()" />}
  </template>
```

```

<template match="xhtml:meta">
  <choose>
    <when test="@name = 'author'">
      \author{<value-of select="@content" />}
    </when>
  </choose>
</template>

<!-- Následující část zpracovává prvek <body> -->
<template match="/xhtml:html/xhtml:body">
  \begin{document}
  \maketitle
  <!-- Zpracuj zanořené prvky <h1> a <p> -->
  <apply-templates select="*" />
  \end{document}
</template>

<template match="xhtml:h1">
  \chapter{<value-of select="text()" />}
</template>

<template match="xhtml:p">
  <value-of select="text()" /> \par
</template>
</stylesheet>

```

Příkazem `xsltproc` *<jméno XSLT programu>* *<jméno XML dokumentu>* ze softwarové knihovny `libxslt` převedeme náš XML dokument na \LaTeX ový dokument:

```

\documentclass{book}
\usepackage[czech]{babel}
\title{Ukázkový dokument v~XHTML}
\author{Vít Novotný}
\begin{document}
\maketitle
\chapter{Kapitola}
Ahoj, XHTML! \par
\end{document}

```

XML jazyky mají dobrou podporu v softwarových knihovnách, což usnadňuje další zpracování dokumentů. Na rozdíl od \LaTeX u nemůže spisovatel v XML jazycích snadno řešit konkrétní typografické nedostatky, což ztěžuje přípravu akcidenčních a nízkonákladových dokumentů. Vzhledem k vysokému poměru značek vůči textu je pro spisovatele výhodné použít specializovaný textový editor pro snadný zápis.

V markdownu je spisovatel vždy konfrontován pouze s jednou otázkou, a je to ta správná otázka: Jak by měla znít následující věta?

— Thompson [15, v překladu autora]

Odlehčené značkovací jazyky jako YAMLoň⁸ [19] a markdown [20] minimalizují poměr značek vůči textu pro snadný zápis a zvýšenou čitelnost:

title: Ukázkový dokument v YAMLoni a markdownu

author: Vít Novotný

date: 2022-05-14

lang: cs

Kapitola

Ahoj, YAMLoni a **markdowne**!

Pro zpracování tohoto dokumentu TEX em můžeme použít např. konverzní nástroj Pandoc [21], makrobalík *markdown* [22] nebo oba zároveň [23; 24, sekce 2.3].

Markdown je jednoduchý jazyk, kterému chybí značky pro složitější a méně časté prvky jako tabulky, poznámky a citace. Pandoc i *markdown* proto nabízí rozšiřující značky⁹ a umožňují uživateli vytvářet značky vlastní [21, sekce Filters; 22, sekce 2.1.2] nebo připojovat k prvkům doplňující *atributy*.¹⁰

Ahoj, rozšiřující značky `[^1]` a `[atributy]{.vysazej-mne-tucne}`.

`[^1]`: Ahoj, já jsem rozšiřující značka pro poznámky.

Jazyk markdown původně vznikl jako preprocessor jazyka HTML a spisovatel v něm proto může využívat také HTML značky.¹¹

Ahoj také `<abbr title="hatmatilko">HTML</abbr>`.

Pro sazbu matematiky, další rozšíření repertoáru značek a řešení konkrétních typografických nedostatků může spisovatel v markdownu použít i primitivní příkazy TEX ového stroje a příkazy TEX ových formátů:¹²

Ahoj i vám, `\LaTeX u`{=latex}` a `\text{matematiko}`\$.

Nadužívání rozšiřujících značek a atributů snižuje čitelnost vstupních dokumentů, ztěžuje jejich další zpracování, narušuje dělbů práce a vede k nejednotnému vzhledu koncových dokumentů. Užívejte je proto s mírou.

⁸YAMLoň podle vzoru jabloň

⁹V Pandocu a *markdownu* je třeba zápis tabulek, poznámek a citací povolit volbami `pipe_tables`, `footnotes` a `citations`.

¹⁰V Pandocu a *markdownu* je třeba zápis atributů povolit volbami `header_attributes`, `fenced_code_attributes`, `inline_code_attributes` a `link_attributes`.

¹¹V *markdownu* je třeba zápis HTML značek povolit volbou `html`.

¹²V Pandocu je třeba zápis matematiky a TEX ových značek povolit volbami `tex_math_dollars` a `raw_attribute`. V *markdownu* je třeba povolit volbu `hybrid`.

Různé značkovací jazyky mají různé výhody a může být vhodné je kombinovat. Markdown a YAMLoň mohou sloužit jako vstupní jazyky pro spisovatele. Pomocí Pandocu můžeme vstupní dokumenty převést do XML mezijazyka pro archivaci a další zpracování. Z XML mezijazyka získáme dokument v L^AT_EXu nebo jiném T_EXovém formátu, který může sloužit jako koncový jazyk pro sazeče.

4. Stylové jazyky pro grafiky

V T_EXu nelze využívat vysokoúrovňové deklarativní stylové jazyky jako CSS a grafici jsou proto odkázáni na programování v T_EXu. V této sekci stručně referuji o tom, jaké možnosti usnadnění nabízí grafikům formáty L^AT_EX_{2 ϵ} a L^AT_EX₃.

Formát L^AT_EX_{2 ϵ} poskytuje vysokoúrovňové příkazy pro vytváření pojmenovaných stylů stránek (`\newpagestyle`), změny obsahu záhlaví a zápatí (`\markboth`) a nastavení rodiny, řezu a velikost písma nezávisle na sobě a nezávisle na konkrétním písmu (`\textbf`) [25]. L^AT_EX_{2 ϵ} dále poskytuje délkové registry pro nastavování rozměrů sazebního zrcadla (`\textheight`) a vzhledu některých L^AT_EXových prvků jako seznamy (`\itemsep`). Rozšiřující makrobalíky L^AT_EXu jako *enumitem*, *geometry* a *fancyhdr* umožňují grafikovi měnit další aspekty vzhledu koncového dokumentu bez programování.

Součástí formátu L^AT_EX₃ je makrobalík *xtemplate* [26; 27], který sazeči, grafikovi a vývojáři pomáhá společně připravovat stylopisy. Nejprve sazeč definuje *typy* prvků dokumentu, např. sekce:

```
\usepackage{xtemplate}
\ExplSyntaxOn
\DeclareObjectType { sekce } { 1 } % Sekce má 1 argument: název
```

Grafik pro každý typ vytvoří *šablonu*. Šablona zadává vysokoúrovňové grafické parametry, kterými se od sebe liší různé úrovně sekcí:

```
\DeclareTemplateInterface { sekce } { moje-šablona } { 1 }
{
  mezera-nad : skip,
  mezera-pod : skip,
  písmo : tokenlist,
  zarovnání : choice { doleva, doprostřed, doprava } = doleva,
}
```

Vývojář implementuje parametry šablony pomocí vysokoúrovňového jazyka `expl3`, stylovacích příkazů L^AT_EXu a primitivních příkazů T_EXového stroje:

```
\skip_new:N \l_mezera_nad_skip
\skip_new:N \l_mezera_pod_skip
\l_new:N \l_pismo_tl
\l_new:N \l_zarovnani_tl
```

```

\DeclareTemplateCode { sekce } { moje-šablona } { 1 }
{ % Hodnoty parametrů ukládáme do lokálních proměnných
  mezeza-nad = \l_mezeza_nad_skip ,
  mezeza-pod = \l_mezeza_pod_skip ,
  písmo = \l_pismo_tl ,
  zarovnaní = {
    doleva = \cs_set_eq:NN \l_zarovnani_tl \raggedright ,
    doprava = \cs_set_eq:NN \l_zarovnani_tl \raggedleft ,
    doprostřed = \cs_set_eq:NN \l_zarovnani_tl \centering ,
  },
}
{ % Při spuštění šablony vysázíme nadpis sekce
\AssignTemplateKeys
\par \skip_vertical:N \l_mezeza_nad_skip
\group_begin:
  \l_zarovnani_tl
  \l_pismo_tl
  #1
  \par \skip_vertical:N \l_mezeza_pod_skip
\group_end:
}

```

Grafik navrhne *instance* šablony s konkrétními hodnotami parametrů pro různé úrovně sekcí jako kapitoly:

```

\DeclareInstance { sekce } { kapitola } { moje-šablona }
{
  mezeza-nad = 10pt,
  mezeza-pod = 12pt,
  písmo = { \Large \bfseries } ,
}
\ExplSyntaxOff

```

Když pak spisovatel v dokumentu zadá nadpis sekce, spustí se příslušná instance:

```

\UseInstance{sekce}{kapitola}{Název její kapitoly}

```

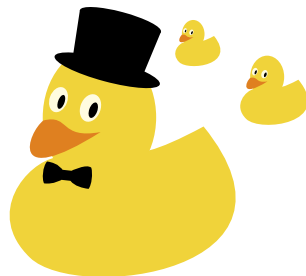
Při použití makrobalíku *xtemplate* může být grafik součástí procesu přípravy dokumentu a průběžně upravovat instance, aniž by musel programovat. V budoucnu má být součástí formátu L^AT_EX3 systém LDB [28; 29], který grafikům oproti makrobalíku *xtemplate* umožní postihnout vzájemné interakce mezi prvky, jako např. seznam bezprostředně po nadpisu (v notaci LDB `!head<list`), zanořené seznamy (`<list*<list`) a druhý popis obrázku (`<float*<caption>*<caption`).

5. Doménově specifické jazyky pro experty

Kromě programovacích, značkovacích a stylových jazyků existuje mnoho dalších vysokoúrovňových jazyků pro doménové experty, jako jsou knihovníci, ilustrátoři a hudebníci. V této sekci uvádím přehledový výčet několika takových jazyků.

Makrobalík *TikZ* [30] poskytuje jazyk pro přípravu ilustrací:

```
\usemodule[tikzducks]
\usecolors[xwi]
\starttext
\starttikzpicture
\duck[tophat, bowtie]
\duck[xshift=60pt, yshift=40pt, scale=0.3]
\duck[xshift=44pt, yshift=55pt, scale=0.2]
\stoptikzpicture
\stoptext
```



Jazyk TikZu je nezávislý na použitém T_EXovém formátu (zde ConT_EXt MkIV) a lze ho rozšiřovat pomocí makrobalíků (zde *tikzducks* [31]).

Makrobalík *BibL_AT_EX* [32] poskytuje jazyk pro přípravu bibliografie:

```
@online{lehnman2022biblalex,
  title = {The Bib\LaTeX{} Package},
  subtitle = {Programmable Bibliographies and Citations},
  author = {Philip Kime and Moritz Wemheuer and Philipp Lehman},
  date = {2022-07-12},
  url = {https://ctan.org/pkg/biblalex},
  urldate = {2022-10-06},
}
```

Jazyk BibL_AT_EXu je přesně definovaný [32, sekce 2] a bibliografické záznamy lze validovat a převést do XML jazyka BibL_AT_EXML pro další zpracování [33, sekce 3.4]. BibL_AT_EX lze rozšiřovat pomocí makrobalíků s citačními styly [34]. Chybí podpora stylpisů ve standardním deklarativním stylovém jazyce CSL.

Makrobalík *LyLuaT_EX* [35] poskytuje jazyk pro notový zápis:

```
\score {
  \relative c' {
    \time 4/4
    \clef treble
    c4 d8 e f r g a \bar "|."
  }
}
```



LyLuaT_EX potřebuje T_EXový formát LuaL_AT_EX a instalovaný program LilyPond.

6. Závěr

$\text{T}_{\text{E}}\text{X}$ je strojový kód světa digitální sazby, který od spisovatelů a grafických návrhářů vyžaduje netriviální programátorské dovednosti a programátorům poskytuje minimum vysokoúrovňových abstrakcí. V článku jsme si představili značkovací, programovací a stylové jazyky pro $\text{T}_{\text{E}}\text{X}$, které umožňují dělbou práce mezi vývojáře, spisovatele a grafické návrháře a které usnadňují proces přípravy elektronických dokumentů.

Článek jsem připravil ve vysokoúrovňovém značkovacím jazyku markdown pro spisovatele a v doménově specifickém jazyce $\text{BibL}_{\text{A}}\text{T}_{\text{E}}\text{X}$ pro knihovníky. Zdrojový kód článku je dostupný online [36].

7. Výhled do budoucnosti

Veškeré jazyky, které jsme si představili v článku, jsou uměle navržené tak, aby byly syntakticky jednoznačné a snadno strojově čitelné. Tím se zcela liší od přirozeného jazyka, který je víceznačný a jeho strojové zpracování bylo svatým grálem informatiky od jejího vzniku. Umělé jazyky jsou překážkou pro doménové experty, kteří ovládají svůj mateřský jazyk, ale nejsou vystudovaní informatici.

V posledních pěti letech došlo na poli umělé inteligence k významným posunům [37; 38], které zvýšily strojovou čitelnost přirozeného jazyka a umožňují automaticky generovat kód programu na základě pokynů zadaných v přirozeném jazyce [39]. Budoucím vysokoúrovňovým značkovacím,¹³ programovacím i stylovým jazykem proto může být přirozený jazyk.¹⁴

Odkazy

1. NOVOTNÝ, Vít. *Vysokoúrovňové jazyky pro $\text{T}_{\text{E}}\text{X}$* . 2022. Dostupné také z: <https://www.cstug.cz/informace/zpravy/2022-04-01-valna-hromada-2022/>.
2. KNUTH, Donald E. *The $\text{T}_{\text{E}}\text{X}$ book*. Sv. A. Reading, MA: Addison-Wesley, 1984. Computers & Typesetting. V současnosti jsou dostupné 35. výtisk (měkká vazba, 2017) a 23. výtisk (pevná vazba, 2021).

¹³Dnešní jazykové modely dokážou v písemném projevu opravit překlepy [40] a doplnit chybějící interpunkci [41]. O značkovacích jazycích můžeme uvažovat jako o rozšířené množině interpunkčních znamének, která nám umožňují text dále členit. Na první pohled by se tedy zdálo, že doplňování chybějících značek je typově stejný problém jako doplňování chybějící interpunkce a k jeho řešení je možné použít stejné techniky. Tuto hypotézu by bylo vhodné ověřit na kolekcích označovaných textů, jako jsou webové stránky označované v jazyce HTML.

¹⁴Posuny v umělé inteligenci se týkají i dalších modalit, jako je např. obraz [42]. To grafikovi umožní doplnit textový popis o nákresey, neboť obrázek vydá za tisíc slov. Tuto primární dokumentaci můžeme kdykoliv strojově přeložit na stylpis v jazyce nižší úrovně, jako je CSS.

3. KNUTH, Donald E. *T_EX: The Program*. Sv. B. Reading, MA: Addison-Wesley, 1986. Computers & Typesetting. Od pátého výtisku (1994) xvi + 600 stran. V současnosti je dostupný 11. výtisk (pevná vazba, 2021).
4. THE $\mathcal{N}\mathcal{T}\mathcal{S}$ TEAM; BREITENLOHNER, Peter. *ϵ -T_EX: An extended version of T_EX, from the $\mathcal{N}\mathcal{T}\mathcal{S}$ project* [online]. CTAN, 1998-02 [vid. 2022-09-26]. Dostupné z: <https://ctan.org/pkg/etex>. Verze 2.
5. THÀNH, Hàn Thê et al. *pdfT_EX: A T_EX extension for direct creation of PDF* [online]. CTAN, 2022-03-01 [vid. 2022-09-26]. Dostupné z: <https://ctan.org/pkg/pdfTeX>. Rev. 875.
6. L^AT_EX DEVELOPMENT TEAM. *LuaT_EX Reference Manual* [online]. CTAN, 2022-02-28 [vid. 2022-09-26]. Dostupné z: <https://ctan.org/pkg/luatex>. Verze 1.15.
7. KNUTH, Donald E. *Plain: The Plain T_EX format* [online]. CTAN, 2021 [vid. 2022-09-26]. Dostupné z: <https://ctan.org/pkg/plain>. Verze 3.141592653.
8. LAMPORT, Leslie. *L^AT_EX: A Document Preparation System*. 2. vyd. Addison-Wesley, 1994. ISBN 978-0201529838.
9. HAGEN, Hans. *ConT_EXt: the manual* [online]. PRAGMA ADE, 2001 [vid. 2022-09-26]. Dostupné z: <http://pragma-ade.com/general/manuals/cont-eni.pdf>.
10. LUAMETAT_EX DEVELOPMENT TEAM. *LuaMetaT_EX Reference Manual* [online]. PRAGMA ADE, 2022-08-03 [vid. 2022-10-04]. Dostupné z: <http://pragma-ade.nl/general/manuals/luametatex.pdf>. Verze 2.09.59.
11. THE L^AT_EX PROJECT TEAM. *The L^AT_EX3 kernel: style guide for code authors* [online]. CTAN, 2022-09-28 [vid. 2022-10-04]. Dostupné z: <https://ctan.org/pkg/l3kernel>.
12. THE L^AT_EX PROJECT TEAM. *The expl3 package and L^AT_EX3 programming* [online]. CTAN, 2022-09-28 [vid. 2022-10-04]. Dostupné z: <https://ctan.org/pkg/l3kernel>.
13. THE L^AT_EX PROJECT TEAM. *The L^AT_EX3 interfaces* [online]. CTAN, 2022-09-28 [vid. 2022-10-04]. Dostupné z: <https://ctan.org/pkg/l3kernel>.
14. SHARIF, Bonita; MALETIC, Jonathan I. An Eye Tracking Study on camelCase and under_score Identifier Styles. In: *18th International Conference on Program Comprehension*. 2010, s. 196–205. Dostupné z DOI: 10.1109/ICPC.2010.41.
15. THOMPSON, Michael. *pandoc-discuss*. Re: Error in “cabal install pandoc” [online]. Google Groups [vid. 2022-10-04]. Dostupné z: <https://groups.google.com/g/pandoc-discuss/c/tKB4E7y6H2E/m/OiieKAuWs14J>.
16. WAGNER, Zdeněk. *Kombinace XML a T_EXu při sazbě divadelní hry*. 2017. Dostupné také z: <https://www.cstug.cz/informace/zpravy/2017-11-15-valnahromada-2017/>.
17. XML [online]. ConT_EXt Garden, 2022-07-01 [vid. 2022-10-04]. Dostupné z: <https://wiki.contextgarden.net/XML>.
18. MAIER, Denis. *Typesetting XML with ConT_EXt*. 2019. Dostupné také z: <https://youtu.be/TEZJ9uZmoJY>.

19. BEN-KIKI, Oren; EVANS, Clark; NET, Ingy döt. *YAML Ain't Markup Language* [online]. 2021-10-01. [vid. 2022-10-05]. Dostupné z: <https://yaml.org/spec/1.2.2/>. Verze 1.2, Revize 1.2.2.
20. GRUBER, John. *Markdown* [online]. Daring Fireball, 2004 [vid. 2022-10-05]. Dostupné z: <https://daringfireball.net/projects/markdown/>.
21. MACFARLANE, John. *Pandoc: a universal document converter* [online]. 2022. [vid. 2022-10-05]. Dostupné z: <https://pandoc.org/>.
22. NOVOTNÝ, Vít. *A Markdown Interpreter for T_EX* [online]. CTAN, 2022-10-03 [vid. 2022-10-05]. Dostupné z: <https://ctan.org/pkg/markdown>. Verze 2.17.1.
23. REHÁK, Dominik. Priama sadzba dokumentov rôznych formátov v TeXu pomocou nástroja Pandoc. *Zpravodaj ČSTUGu*. 2021, roč. 31, č. 1–4, s. 83–92. Dostupné z DOI: 10.5300/2021-1-4/83.
24. NOVOTNÝ, Vít et al. Markdown 2.15.0: What's New? *TUGboat*. 2022, roč. 43, č. 1, s. 10–15. Dostupné z DOI: 10.47397/tb/43-1/tb133novotny-markdown.
25. THE L^AT_EX PROJECT TEAM. *L^AT_EX 2_ε font selection* [online]. CTAN, 2021-12 [vid. 2022-10-05]. Dostupné z: <https://ctan.org/pkg/fntguide>.
26. THE L^AT_EX PROJECT TEAM. *The xtemplate package: Prototype document functions* [online]. CTAN, 2022-06-22 [vid. 2022-10-05]. Dostupné z: <https://ctan.org/pkg/xtemplate>.
27. NIEDERBERGER, Clemens. The xtemplate package: An example. *TUGboat*. 2012, roč. 33, č. 3, s. 272–275. Dostupné také z: <https://tug.org/TUGboat/tb33-3/tb105niederberger.pdf>.
28. MITTELBACH, Frank. *L^AT_EX 3 architecture and current work in progress*. 2011. Dostupné také z: <https://youtu.be/-lr6KEPGLDs>.
29. MITTELBACH, Frank. *Using L^AT_EX 3's xtemplate* [online]. Stack Exchange, 2013-06-06 [vid. 2021-12-06]. Dostupné z: <https://tex.stackexchange.com/a/118015/70941>.
30. TANTAU, Till. *The TikZ and PGF Packages: Manual for version 3.1.9a* [online]. CTAN, 2021 [vid. 2022-10-06]. Dostupné z: <https://ctan.org/pkg/pgf>.
31. CARTER, Sam. *The TikZducks package: using ducks in TikZ* [online]. CTAN, 2020 [vid. 2022-10-06]. Dostupné z: <https://ctan.org/pkg/tikzducks>. Verze 1.5.
32. KIME, Philip; WEMHEUER, Moritz; LEHMAN, Philipp. *The BibL^AT_EX Package: Programmable Bibliographies and Citations* [online]. CTAN, 2022-07-12 [vid. 2022-10-06]. Dostupné z: <https://ctan.org/pkg/biblatex>. Verze 3.18b.
33. NOVOTNÝ, Vít. Příprava Zpravodaje ČSTUG. *Zpravodaj ČSTUGu*. 2018, roč. 28, č. 1–4, s. 1–10. Dostupné z DOI: 10.5300/2018-1-4/1.
34. LUPTÁK, Dávid. Sadzba bibliografie podľa normy ISO 690 v systéme L^AT_EX. *Zpravodaj ČSTUGu*. 2016, roč. 26, č. 1–4, s. 106–120. Dostupné z DOI: 10.5300/2016-1-4/106.
35. PERON, Fr. Jacques; LISKA, Urs; SPRINGUEL, Br. Samuel. *lyLuaT_EX: Programmable Bibliographies and Citations* [online]. CTAN, 2019-05-27 [vid. 2022-10-06]. Dostupné z: <https://ctan.org/pkg/lyluatex>. Verze 1.0f.
36. NOVOTNÝ, Vít. *Vysokoúrovňové jazyky pro T_EX* [online]. GitHub [vid. 2022-10-16]. Dostupné z: <http://github.com/witiko/high-level-languages-for-tex>.

37. DEVLIN, Jacob et al. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: *NAACL 2019*. 2018. Dostupné také z: <https://arxiv.org/abs/1810.04805v2>.
38. BROWN, Tom et al. Language Models are Few-Shot Learners. In: LAROCHELLE, H. et al. (ed.). *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2020, sv. 33, s. 1877–1901. Dostupné také z: <https://arxiv.org/abs/2005.14165v4>.
39. PAPERS WITH CODE CONTRIBUTORS. *Code Generation* [online]. Ed. STOJNIC, Robert et al. Papers with Code [vid. 2022-09-29]. Dostupné z: <https://paperswithcode.com/task/code-generation>.
40. ZHOU, Yingbo; PORWAL, Utkarsh; KONOW, Roberto. Spelling Correction as a Foreign Language. In: DEGENHARDT, Jon et al. (ed.). *eCOM 2019: The SIGIR 2019 Workshop on eCommerce* [online]. Paris, France, 2019 [vid. 2022-09-29]. CEUR Workshop Proceedings, č. 2410. ISSN 1613-0073. Dostupné z: <http://ceur-ws.org/Vol-2410/paper28.pdf>.
41. NAGY, Attila; BIAL, Bence; ÁCS, Judit. *Automatic punctuation restoration with BERT models* [online]. Cornell University, 2021 [vid. 2022-09-29]. Dostupné z: <https://arxiv.org/abs/2101.07343v1>.
42. LU, Jiasen et al. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. In: WALLACH, H. et al. (ed.). *NeurIPS 2019*. Curran Associates, Inc., 2019, sv. 32. Dostupné také z: <https://arxiv.org/abs/1908.02265v1>.

Summary: High-Level Languages for \TeX

\TeX is the assembly language of digital typesetting, which requires advanced programming skills from authors and designers, and which provides few high-level abstractions to programmers. In this article, I introduce selected markup, programming, and style-sheet languages for \TeX , which enable the division of labor between authors, programmers, and designers, and which simplify the process of electronic document preparation. The article transcribes my invited talk at the general assembly of \LaTeX on May 14, 2022 [1].

Keywords: high-level languages, programming languages, markup languages, style-sheet languages, $\varepsilon\text{-}\text{\TeX}$, $\text{\pdf}\text{\TeX}$, $\text{\Lua}\text{\TeX}$, $\text{\LuaMeta}\text{\TeX}$, $\text{\La}\text{\TeX} 2\varepsilon$, $\text{\La}\text{\TeX} 3$, \expl3 , XML, DocBook, TEI, XHTML, XSLT, CSS, CSL, $\text{\Con}\text{\TeX}t$, HTML, markdown, YAML, Pandoc, \TikZ , $\text{\Bib}\text{\La}\text{\TeX}$, $\text{\Bib}\text{\La}\text{\TeX}XML$, $\text{\Ly}\text{\Lua}\text{\TeX}$

Vít Novotný, witiko@mail.muni.cz