



Typografie a T_EX (02)

Značkovací jazyky v T_EXu
petr@olsak.net

<http://petr.olsak.net/typotex.html>



IniT_EX versus makrobalíky

2

- IniT_EX je vybaven cca 300 primitivními příkazy a registry. Příkazy spouštějí vnitřní algoritmy T_EXu (například `\halign`, `\kern`, `\def`, `\hbox`) zatímco registry obsahují parametry vnitřních algoritmů T_EXu a je možné je měnit: (`\hsize`, `\voffset`, `\tolerance`). Příklady:

```
\hsize = 13cm  
\def \TeX {T\_kern-.1667em\_lower.5ex\_hbox{E}\_kern-.125emX}
```

- Začít přímo zpracovávat dokument jen na základě toho, co nabízí IniT_EX je nelidské a nespportovní. Proto existují makrobalíky implementované pomocí `\dump` jako formáty T_EXu.
- Specifikum T_EXu: Na vstupu může být směs příkazů, maker a textu z dokumentu, který se má vytisknout. Záleží jen na uživateli, zda udrží jistou kulturu zdrojových textů dokumentů, oddělí typografická nastavení od dobře označovaného textu dokumentu, nebo to nevhodně smíchá dohromady.
- Pravidlo: doporučuje se odělit formu od obsahu.



Plain T_EX Donalda Knutha

3

- Výchozí sada maker obsažená v přesné nebo modifikované podobě i v dalších „vyšších“ sadách maker.
- Konzervovaná D. Knuthem v roce 1989.
- Dokumentovaná v příloze knihy T_EXbook (1. díl Computers & Typesetting).
- Je možné v plain T_EXu psát dokument, ale je často třeba si k tomu dodělat plno vlastních maker (na tvorbu obsahu, na vyznačení kapitol, sekcí, na efektivnější manipulaci s fonty...)
- D. Knuth v plainT_EXu vysázel sérii Computer & Typesetting nebo The Art of Computer Programming. Nepoužil jen plain T_EX, ale také další (veřejně dostupná) makra `manmac.tex`.
- Čistý plain T_EX je použitelný jen pro anglické dokumenty.
- V plain T_EXu neexistuje jednotný přístup ke značkování dokumentů.



Jemná rozšíření plain T_EXu

- Různé jazykové mutace, např. \mathcal{C}_S plain, viz <http://petr.olsak.net/csplain.html>

Autor \mathcal{C}_S plainu zaručuje podobnou neměnnost \mathcal{C}_S plainu jako v případě plain T_EXu D. Knutha.
- Příkaz `csplain` generuje (z konzervativních důvodů) DVI, zatímco příkaz `pdfcsplain` aktivuje výstup do PDF. Obojí je nad enginem `pdftex`. Dále je k dispozici příkaz `luacsplain`, generuje PDF, používá engine `luatex` a je to dnes preferovaný způsob použití \mathcal{C}_S plainu.
- Většinou je v distribucích při použití příkazů `pdftex`, `luatex` a `xetex` dostupný plain T_EX generovaný společně s makrem `etex.src`, což zavádí všechny dostupné vzory dělení. Není to tedy čistý plain T_EX D. Knutha.
- \mathcal{C}_S plain obsahuje makrobalík OPmac.



AMS T_EX

- Michal Spivak rozšířil makra plain T_EXu (1981) o dodatečné možnosti v matematické sazbě a přidal k tomu sadu matematických fontů s mnoha značkami nepodporovanými přímo v Computer Modern.
- M. Spivak k tomu napsal manuál Joy of T_EX.
- Podporuje to AMS (American Mathematical Society)
- Dnes není AMS T_EX samostatným formátem. V plain T_EXu stačí
`\input amstex.tex`
a součástí L^AT_EXu je tzv. AMSL^AT_EX nabízející použít
`\usepackage{amsmath}`



L^AT_EX – významné časové mezníky

- Leslie Lamport dal veřejnosti svá makra kolem roku 1984. V roce 1986 publikoval v AW L^AT_EX manuál. Začalo se to široce užívat pro přípravu odborných publikací.
- L^AT_EX deklaruje základní principy značkování dokumentu (kapitoly, sekce, poznámky, odrážky, citace atd.). Výhoda: dokument je připraven v mezinárodně uznávaném značkování.
- L^AT_EX zavádí koncept hlavního stylu (*article*, *book*, *report*, *letter*, *slides*) a dodatečně zavedených balíčků *maker*.
- V roce 1990 předal L. Lamport vývoj L^AT_EXu F. Mittelbachovi. Vznik L^AT_EXu2e (1994), existuje tým kolem L^AT_EXu vyvíjející L^AT_EX3 (dosud).
- Po zveřejnění XeL^AT_EXu a LuaL^AT_EXu byla makra upravena pro tyto enginy a vznikl XeL^AT_EX a LuaL^AT_EX (příkazy *xelatex*, *lualatex*). Zde přestávají mít smysl balíčky *inputenc*, *fontenc*, *babel* a je třeba místo nich použít *fontspec* a *polyglosia*.



L^AT_EX – vývojové etapy před Unicode

- Před rokem 1994 se používaly veze ≤ 2.09 . Typické pro ně je první povinný příkaz v dokumentu `\documentstyle{...}`.
- Od roku 1994 L^AT_EX team nabízí verzi pre-L^AT_EX3, nazvanou L^AT_EX2e. Dokument může vypadat takto:

```
\documentclass[a4paper,11pt]{article} % V LaTeXu povinné
\usepackage[czech]{babel}           % nastavení jazyka
\usepackage[utf8]{inputenc}         % nastavení vstupního kódování
\usepackage[T1]{fontenc}            % nastavení výstupního kódování
\usepackage{newcent}                % nastavení rodiny fontů
\usepackage ...                     % další balíčky
\begin{document}
...
\end{document}
```

- Detekuje-li L^AT_EX2e jako první příkaz `\documentstyle`, přechází do tzv. compatibility módu a chová se jako L^AT_EX verze 2.09.
- Toto je ukázka pro 8bitový engine, zpracujete ji příkazem `pdflatex`.



L^AT_EX2e – vývojová etapa nyní

- Od roku 2004 existuje Xe^TE_X a některá volání balíčků je třeba v Xe^LA^TE_Xu (nebo později v Lua^LA^TE_Xu) nahradit:

```
\documentclass[a4paper,11pt]{article} % V LaTeXu povinné
\usepackage{polyglossia} % nastavení jazyka (náhrada za babel)
\setdefaultlanguage{czech} % české vzory dělení, atd.
\usepackage{fontspec} % nastavení fontu
\setmainfont[Ligatures=TeX,
  Extension=.otf,
  UprightFont= *-regular,
  BoldFont=*-bold,
  ItalicFont=*-italic,
  BoldItalicFont=*-bolditalic]
{texgyreschola} % nastavení rodiny fontů
\usepackage ... % další balíčky
\begin{document}
...
\end{document}
```

- Dnes se nemá smysl vracet k L^AT_EXu na 8 bitových enginech. Zkuste preferovat Xe^LA^TE_X nebo Lua^LA^TE_X, když už tíhnete k L^AT_EXu. Ukázku zde uvedenou zpracujete příkazem **xelatex** nebo **lualatex**.
- Uživatelská základna L^AT_EXu je konzervativní, k Unicode stále mnozí nepřešli a používají **pdflatex**.



Nevýhody LaTeXu

- Snaží se před uživatelem zakrýt TeX, ale to v principu nejde.
- Původně navržen jako jeden formát s konkrétní typografií. Možnosti nastavení jiné typografie pomocí balíčků věci znehledňují.
- Balíčky desítek různých autorů mohou být vzájemně sporné.
- Makra balíčků i LaTeXu samotného jsou zbytečně komplikovaná, protože často řeší, jak vytvořit jiné než primitivní TeXové rozhraní.
- Zavádí i pro jednoduché úkony jinou než primitivní syntaxi. Například místo `\hsize=10cm` se má v LaTeXu použít `\setlength{\textwidth}{10cm}`
V dokumentu je pak obojí syntaxe možná a vzniká tak často zmatek.
- LaTeXová dokumentace nerozlišuje, co je TeXové a primitivní a co LaTeXové. Například se dozvíme, že `\baselineskip` je LaTeXový parametr, ale to není pravda.
- LaTeX mění některá makra plain TeXu (`\line`) a primitivní příkazy (`\end`).
- Jádro LaTeXu2e potřebuje dodatečně načíst plno balíčků, neboť vývojově se zastavilo v roce 1992. Neumí barvy, hyperlinky...
- Dále též viz můj článek [Proč nerad používám LaTeX z roku 1997](#).



Exp13

- Další rozhraní mezi uživatelem a T_EXem snažící se zakrýt chování T_EXu a jeho syntaxi.
- Vytváří programovací jazyk implementovaný v makrech T_EXu. Je určen pro programování L^AT_EXových balíčků a jádra L^AT_EXu3.
- Velké množství L^AT_EXových balíčků je již do tohoto jazyka přepsáno.
- Ve snaze řešit chybějící separaci jmenných prostorů (to T_EX bohužel neumí) vytváří konvenci pro pojmenovávání kontrolních sekvencí, která je naprosto obskurní a silně zneřehledňuje kód.

```
\cs_new_protected:Npn \kozlovskiy_maketitle:
{
  \tagstructbegin{tag=Title}
    \prop_map_function:NN \l__kozlovskiy_data_of_title_prop
      \__kozlovskiy_tag_elt:nn
    \__kozlovskiy_original_maketitle:
  \tagstructend
}
\cs_new_eq:NN \__kozlovskiy_original_maketitle: \maketitle
\cs_gset_protected:Npn \maketitle { \kozlovskiy_maketitle: }
```



ConTeXt

- ConTeXt vytvořil v první verzi Hans Haagen v roce 1990, vyvíjí jej dosud.
- ConTeXt udělal velký skok v roce 2009, kdy byl propojen s LuaTeXem (mark IV ConTeXt). Od té doby tvoří nerozlučnou dvojici.
- Na rozdíl od L^AT_EXu nabízí všechny vlastnosti požadované v dnešní době při tvorbě dokumentů (barvy, odkazy, zařazování grafiky, bibliografie, ...) bez nutnosti načítat na každou věc nějaký balíček.
- Pro tvorbu grafických prvků je propojen s MetaPostem.
- Na rozdíl od L^AT_EXu nabízí jednotnou syntaxi pro nastavování typografických parametrů, ovšem také zakrývá T_EXové primitivní rozhraní.
- Vymezuje jiný značkovací jazyk pro dokumenty než L^AT_EX, více přizpůsobený možnostem měnit typografické parametry.
- Výrazně se odchyluje od plain T_EXu (např. při tvorbě tabulek).



OPmac

- Souromý balíček maker nad \mathcal{C}_S plainem zveřejnil autor v roce 2013.
- Je součástí \mathcal{C}_S plainu a zavolá se pomocí `\input csplain`.
- Snaží se o návrat k $\text{T}_\text{E}\text{X}$ u, tj. makra nezakývají primitivní úroveň $\text{T}_\text{E}\text{X}$ u, pouze ji doplňují.
- Umí vše, co se typicky předpokládá při přípravě dokumentu v dnešní době: barvy, odkazy, hyperlinky, zařazování grafiky, transformace, bibliografii, rejstříky, fontový systém, ... Rejstříky se řadí a bibliografie vybírá jen pomocí maker OPmac bez externího programu.
- Běhá na všech enginech se základním makrem plain $\text{T}_\text{E}\text{X}$.
- Makra jsou jednoduchá a přímočará (v ukázce je srovnání v počtu řádku kódu $\text{LaT}_\text{E}\text{X}$ ových balíčků a OPmac). Důvod je ten, že se makra nesnaží zakrýt $\text{T}_\text{E}\text{X}$ ani vytvořit nad ním novou syntaxi.
- Chcete-li modifikovat typografii nebo jiné chování dokumentu, neděláte to pomocí stovek nových parametrů, ale tak, že si zkopírujete makro OPmac do svého makro souboru a upravíte si je podle potřeby.
- Viz např. plakátek OPmac.



OpTeX

- První verzi autor zveřejnil v lednu v roce 2020.
- Spojuje LuaTeX, plainTeX a OPmac.
- Viz <http://petr.olsak.net/optex/>.
- Využívá síly maker OPmac, opouští komplikované klíčky dříve nutné kvůli existenci různých kódování a různých enginů. Podporuje jediné LuaTeX a Unicode.
- Makra zůstávají jako v PlainTeXu jednoduchá.
- Hlavní nápor na vývoj očekává autor v létě roku 2020. Už nyní existuje v TeXlive alpha verze pro testování.



CTUstyle

- Šablona pro přípravu závěrečných studentských prací na ČVUT.
- Autor ji vytvořil na základě námětu vysloveném v diskusním fóru studentů ČVUT v roce 2013.
- Šablona se opírá o plain $\text{T}_\text{E}\text{X}$ a OPmac.
- Viz <http://petr.olsak.net/ctustyle.html>.
- V roce 2015 byla Tomášem Hejdou vytvořena šablona CTUthesis postavená na $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ u a přebírající kompletně typografický návrh CTUstyle. Stalo se tak na popud $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ ových uživatelů, kteří nechtěli používat plain $\text{T}_\text{E}\text{X}$ a OPmac.
- V únoru 2018 byla vytvořena verze 2 CTUstyle, implementující doporučený vizuální styl na ČVUT založený na písmu Technika.
- $\text{L}^{\text{A}}\text{T}_\text{E}\text{X}$ ová obdoba CTUstyle2 dosud neexistuje.
- Tyto slídy, na které zrovna hledíte, jsou vytvořeny v CTUstyle2.