

UŽITÍ T_EXU A LUA PŘI SAZBĚ KNIHY PRŮVODCE SVĚTEM ARDUINA

PAVEL STRÍŽ (CZ)

Abstrakt. Během sazby knihy *Průvodce světem Arduina* (viz <http://www.striz.cz/90arduino.php>) od Zbyška Vody a HWKitchen (Oldřich Horáček) jsem narazil na zajímavý problém. Během dokončení knihy jsem potřeboval extrahovat zdrojové kódy, zásáhnout do nich (vyházet diakritické znaky, nabízela se možnost přidání jednotného záhlaví dle typu souboru) a nadělat z nich individuální textové soubory, které v názvu souboru obsahují číslo strany knihy. Tyto soubory se pak šíří mimo tištěnou knihu. Navíc jsem potřeboval ošetřit situaci, kdy začíná na jedné straně knihy víc zdrojových kódů, tedy to chtělo nějaký čítač zdrojových kódů nebo číslo strany s dodatečným čítačem. Řešení jsem potřeboval univerzální, aby se při sazebním zásahu v knize všechny zdrojové kódy znovu extrahovaly a byly vždy aktuální. Představím své řešení a knihu nechám kolovat mezi účastníky konference s možností si ji na místě zakoupit.

Klíčová slova. Arduino, T_EX Live, T_EX, Lua, listings.

USING T_EX AND LUA IN TYPESETTING BOOK THE GUIDE IN THE WORLD OF ARDUINO

Abstract. The article introduces problems which the typesetter encountered while typesetting book *The Guide in the World of Arduino* by Zbyšek Voda and HWKitchen (Oldřich Horáček) <http://www.striz.cz/90arduino.php>. The author presents his solutions, tips and tricks. The most interesting problem was an extraction of the source codes mentioned in the book. The filenames needed to include page number where source code starts. A unique counter was necessary because source codes can start on the same page. This was a fine place where the author used Lua. My brother and I have finished book under tremendous time pressure (just before Christmas 2015), but we accomplished that on time. Book was available for seeing and buying at the conference venue. There was 200 copies in the first printing and 1000 copies in the second printing.

Keywords. Arduino, T_EX Live, T_EX, Lua, listings.

T_EXista kafrá do Open Hardware místo Úvodu

V červenci 2015 jsem se poprvé dozvěděl o Arduinu a neměl jsem s ním zkušenosti ani s Raspberry Pi. Měl jsem v hlavě divadelní projekt (viz samostatný článek v tomto sborníku) a velkou chuť si to „něco“ připravit. Proto jsem začal aktivně studovat, nakupovat součástky a experimentovat.

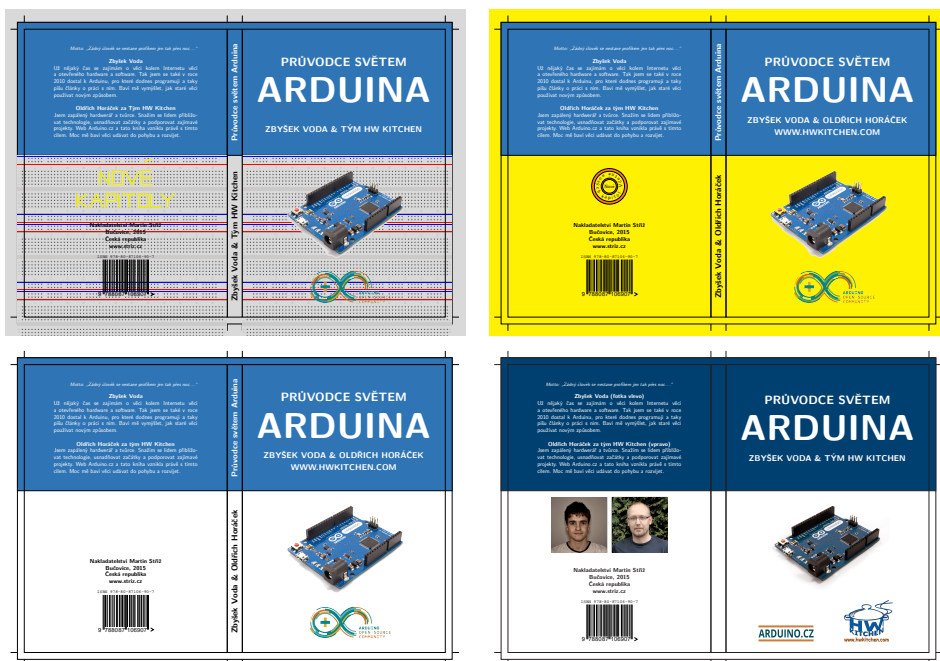
Hlavně první měsíce testů byly intenzivní a dostal jsem se do kontaktu s Oldřichem Horáčkem z HWKitchen. Kde mohl, poradil (jak začít, jaká relátka zvažovat ap.). Prozradil jsem na sebe, že jsem typograf a bratr má nakladatelství. On zas prozradil, že by rád dokončil papírovou knihu, kterou vydávali jako seriál na webových stránkách a jako PDF pro zájemce ke stažení zadarmo.

To však vyžadovalo sazební zásahy, čištění a sjednocení zdrojových kódů (více autorů, články psané v různých obdobích). A najednou byl projekt na mém ponku.

1. Sazba mimo blok knihy: obálka

Ačkoliv jsem byl pod zajímavým časovým tlakem, zkusil jsem si několik verzí obálky. Představuji dvě ukázky nevyužitých nápadů a barevnou variantu přesazenou a dokončenou dle návrhu autorů (představuji rozkresy, se kterými se čtenář knihy normálně nesetká). Na tento úkol téměř výhradně používám TikZ.

Bylo potřeba převést barevné obrázky do stupňů šedi a zajistit, že jsou v barevném prostoru CMYK. Užil jsem GraphicsMagick. Na generování několika variant jsem použil proměnné (příkaz echo) a několik běhů $\text{T}_{\text{E}}\text{X}$ u (načítání generovaného souboru programem bash přes $\backslash\text{input}$).



Obrázek 1. Dva nápady: užít Fritzing a jinou barevnost s razítkem, ve spodním řádku je původní rozkres dle návrhu autorů (vlevo) a dokončená obálka po dvou či třech iteracích

2. Zásahy před sazbou knihy: čištění

Letos se na konferenci zadařilo pokračovat v představování LuaT_EXu (Pavel Tišnovský, včetně dvou workshopů), resp. ConT_EXtu (Tomáš Hála, zvaná přednáška). Máme z toho velkou radost. Velice efektivní je používat Lua přímo za běhu T_EXu, nemusíte věci řešit před či po běhu T_EXu, je to jiný svět. V dřívějším zaměstnání jsem si dokonce ověřoval věci z webu za běhu T_EXu (`\write18 a wget`). Poněvadž jsem zdrojové kódy „vracel“ zpátky autorům, kteří umí s pdfL^AT_EXem, tak jsem Lua používal hlavně mimo T_EX.

Masivně používám Lua na čištění a zanášení korektur u datových, textových, obecně T_EXových souborů. O tom snad víc v budoucnu. Používám na to regulární výrazy (příkaz Lua `unicode.utf8.gsub`) s tahem dostudovat knihovnu LPeg.

Představím dva momenty z čištění před T_EXováním. Ve zdrojových kódech (prostředí `lstlisting`) jsme se domluvili, že nebude diakritika, někde byla, někde ne. Nepodařilo se mi znovu dohledat univerzální řešení pro všechny diakritické znaky (Unicode, asi to bylo v Perlu), udělal jsem to pro české diakritické znaky.

Druhým momentem bylo aplikování nedělitelných mezer za jednoznakové předložky (já s oblibou nastavuji i spojky). Na to se hojně používá program `vlna` od Petra Olšáka, ale musel jsem zajistit, aby se tentokrát do zdrojových kódů nezasáhlo. To je T_EXové prostředí, nikoliv však `verbatim`, které by program `vlna` ještě uměl vynechat (parametr `-n`).

2.1. Miniukázky

Začněme jednoduchou ukázkou, kdy chceme desetinné tečky převést na desetinné čárky. Princip je jednoduchý, jak však Lua není nejužívanějším jazykem, je dobré na to poukázat. Procento a tečka tvoří operátor (tečka doslova), jinak tečka znamená libovolný znak.

U druhé ukázky využijeme anonymní funkci a znaky otočíme, co byla tečka bude čárka a obráceně. V parametru `s` (používám to jako zkratku slova String) máme uložený nalezený znak a na jeho místo vrátíme něco nového.

U třetí ukázky pracujeme s vícebajtovou sekvencí (diakritické znaky), to už s příkazem `string.gsub` nevystačíme. Můžeme pracovat s vestavěným příkazem `unicode.utf8.gsub`. Ukažme si to na počtu znaků.

```
textA=[[123.45 24.78 81.36]]
textA=string.gsub(textA, "%.", ",")
print(textA) -- 123,45 24,78 81,36

textB=[[47,54 61.7 4,5 9.8 10,4]]
textB=string.gsub(textB, "[%.,]", function(s)
    if s=="." then return "," else return "." end
end)
print(textB)-- 47.54 61,7 4.5 9,8 10.4
```

```

textC=[[Pavel Stříž šel na oběd.]]
print(string.len(textC))      -- 29, chybné (je to počet bajtů)
print(unicode.utf8.len(textC)) -- 24, správné (je to počet znaků)

```

Pokud máme následující řádky uloženy pod `zamen.lua`, spustíme si (autor z T_EX Live 2016): `texlua zamen.lua`.

Podívejme se na dvě náročnější ukázky z čištění bloku knihy.

2.2. Náhrada diakritických znaků ve zdrojových kódech

Náhradu diakritických znaků si dáme do cyklu. Už musíme jen vyhledat bloky zdrojových kódů, v knize se používal balíček `listings` a prostředí `lstlisting`. Ponevadž není pravděpodobné, že by se prostředí vnořovala, lze použít sekvenci „.-“: najdi nejbližší.

Pro zájemce kolem úpravy textů hledejte více ukázek např. v balíčku `chickenize`.

```

-- Zadej texty.
obsah=[[Text předem.
\begin{lstlisting}
Text uprostřed kódu.
\end{lstlisting}
Text za zdrojovým kódem.]]
celkem=0 -- Počty měněných znaků.
diafrom, diato=
  "áéíóúýčďěňřšťžůÁÉÍÓÚÝČĎĚŇŘŠŤŽŮ",
  "aeiouycdenrstzuaEIOUYCDENRSTZU"

-- Zpracuj texty.
obsah=unicode.utf8.gsub(obsah,
"\begin{lstlisting}.-\end{lstlisting}",
function(s)
for poradi=1,unicode.utf8.len(diafrom) do
  znakfrom=unicode.utf8.sub(diafrom,poradi,poradi)
  znakto=unicode.utf8.sub(diato,poradi,poradi)
  s,kolik=unicode.utf8.gsub(s,znakfrom,znakto)
  if kolik>0 then celkem=celkem+kolik end
end -- for
return s
end -- function)

-- Vypiš výsledek.
print("Celkem měněných diakritických znaků: "..celkem)
-- Dostáváme: Celkem měněných diakritických znaků: 2

```

2.3. Rychlá vlna přes Lua

Zajímavější situace je, když chceme inverzi, tedy mimo zdrojové kódy. Nevýhoda operátoru %b je, že pracuje jen s jedním znakem. Místo, abych přešel na LPeg, tak jsem si začátek prostředí nahradil za znak \002 a konec prostředí za znak \003. Po úpravách si nastavím zpět. Dává mi to možnost párového vyhledávání i cyklu přes různá prostředí. U inverze už stačí si přidat trojku na začátek a dvojku na konec řetězce. Vyhledávat \002 až \003 je prostředí, \003 až \002 jsou bloky mimo prostředí. Podobný způsob se dá využít na komplikovanější vyhledávání: uvnitř a mimo matematiku ap.

Za pozornost u tohoto problému stojí nejen program vlna (Petr Olšák), ale i encxvlna (Zdeněk Wagner, Petr Olšák) a luavlna (Michal Hoftich).

```
-- Zadej upravovaný text
obsah=[[Text v přední části s~i bez nedělitelné mezery.
\begin{lstlisting}
Zde jsem v zóně.
Nic neměnit!
\end{lstlisting}
Text pokračuje v této části.]]

-- Tam
obsah=unicode.utf8.gsub(obsah, "\\begin{lstlisting}", "\002")
obsah=unicode.utf8.gsub(obsah, "\\end{lstlisting}", "\003")
obsah="\003"..obsah.."\002"

-- Zásahy
obsah=unicode.utf8.gsub(obsah, "%b\003\002",
function(s)
s=unicode.utf8.gsub(s,"([%s~][KkSsVvZzOoUuAaIi])%s","%1~")
-- Zde může být řada dalších zásahů a čištění.
return s
end)

-- Zpět
obsah=unicode.utf8.sub(obsah, 2, -2) -- Umaž pomocný první a
poslední znak.
obsah=unicode.utf8.gsub(obsah, "\002", "\\begin{lstlisting}")
obsah=unicode.utf8.gsub(obsah, "\003", "\\end{lstlisting}")
-- Vypiš výsledek
print(obsah)
--[[Text v~přední části s~i~bez nedělitelné mezery.
\begin{lstlisting}
```

```

Zde jsem v zóně.
Nic neměnit!
\end{lstlisting}
Text pokračuje v~této části.--]]

```

3. Zásahy po sazbě knihy: extrakce zdrojových kódů

Existuje několik T_EXových balíčků a pomůcek za běhu Lua. T_EXové balíčky jsem nevyužil, protože jsem potřeboval do souborů zasahovat a do poslední chvíle nebylo jisté, jestli se bude přidávat komentář, tzv. hlavička souboru (v Arduinu `/* ... */`), případně kontrolní znaky či velikost souboru na konci zdrojového kódu. Lua za běhu T_EXu nešlo použít, neb jsem chtěl zůstat v pdf^LA_TE_Xu. Kdyby nastal problém či si autoři potřebovali zasáhnout, měli by možnost.

Postup byl takový, že T_EX si u každého zdrojového kódu zapisoval do `.aux` souboru informace jako u každého jiného křížového odkazu. Využil jsem vlastní značku (`lst-pajovo`), ale šel by užít extra soubor.

Po dokončení T_EXování si Lua vytáhl informace z tohoto `.aux` souboru, vytáhl si zdrojové kódy z T_EXového souboru (s bratrem obvykle pracujeme s jedním T_EXovým souborem, usnadňuje to vyhledávání a globální náhrady v celé knize), extrahoval je do souboru, který se jistým klíčem pojmenoval.

V preambuli dokumentu za `\usepackage{listings}` jsem dopsal:

```

\let\oldbeginlstlisting=\lstlisting
\newcount\pajovo
\pajovo=0
\def\lstlisting{%
  \global\advance\pajovo by 1%
  \label{lst-pajovo-\the\pajovo}%
  \oldbeginlstlisting}

```

V `.aux` souboru jsem snadno dohledával informace typu:

```

\newlabel{lst-pajovo-1}{6.4}{24}{Programovací jazyk}{section.6.4}{}
\newlabel{lst-pajovo-2}{6.4}{24}{Programovací
jazyk}{lstnumber.-1.9}{}
\newlabel{lst-pajovo-3}{6.4}{25}{Programovací
jazyk}{lstnumber.-2.3}{}
[... zkráceno ...]

```

Řídící soubor `vytahni.lua` si vytáhl potřebné a připravil extrakci zdrojových kódů s příslušnými názvy obsahující číslo strany knihy, kde zdrojový kód začíná.

Za pozornost stojí volání si příkazu `unicode.utf8.gsub` v anonymní funkci uvnitř `unicode.utf8.gsub` (hledám část ve vyhledané části). Také si výsledek neukládám do proměnné `tex`, nepotřeboval jsem jej.

```
-- Vstupní parametry
zdrojaux="preamble.aux"
zdrojtex="arduino-blok.tex"
adresar="!zdrojaky/"

-- Otevření souborů a načtení obsahů
soubor=io.open(zdrojaux,"r")
aux=soubor:read("*all")
soubor:close()

soubor=io.open(zdrojtex,"r")
tex=soubor:read("*all")
soubor:close()

-- Načtení údajů a zpracování
jadro={}
unicode.utf8.gsub(aux, "\\newlabel{lst%-pajovo%-(%d+)}(%b{}}",
function(citac,strana)
strana=unicode.utf8.sub(strana,2,-2)
strana=unicode.utf8.gsub(strana,"%b{}}(%b{}}).+", "%1")
strana=unicode.utf8.sub(strana,2,-2)
print(citac,strana)
jadro[tonumber(citac)]=strana
end)

-- Úpravy a zápis zdrojových kódů do souborů
citac=0
unicode.utf8.gsub(tex, "\\begin{lstlisting}(.)\\end{lstlisting}",
function(s)
s=unicode.utf8.gsub(s, "^%s-%b[[]", "") -- vol. parametr pryč
s=unicode.utf8.gsub(s, "^%s+", "") -- mezery začátek řádků
s=unicode.utf8.gsub(s, "%s+$", "") -- mezery konec řádků
citac=citac+1
fstrana=unicode.utf8.format("%03d",jadro[citac])
fcitac=unicode.utf8.format("%03d",citac)
print("Zpracovávám kód číslo "..citac.."...")

kam=io.open(adresar.."strana-"..fstrana.."poradi-"..fcitac.."txt",
"w")
kam:write(s)
kam:close()
end)
```

U této konkrétní knihy pak v předpřipraveném adresáři vznikly TXT soubory s přibližně těmito názvy:

```
strana-024-poradi-001.txt
strana-024-poradi-002.txt
strana-025-poradi-003.txt
[... zkráceno ...]
```

Promazat si adresář jsem dělal na úrovni operačního systému v dávkovém souboru či se nabízí `Makefile`.

Název obsahuje stranu v knize (můžete si sami ověřit, až budete knihu držet v ruce) a celkové pořadí kódu v knize. Poznámky v úvodu či na závěr kódu se nedávaly. Poněvadž se kódy dají vykopírovat z PDF, tak se autoři rozhodli tyto soubory zatím nevyužít. Připravené to však je, dodávám s humorem.

4. Kdy mi práce končí: předávka knihy

Má práce skončila, když jsem předal vše (upravené a ořezané obrázky, zdrojové kódy obálky a bloku, vyextrahované zdrojové kódy) a soubor přijala tiskárna. Použil jsem program `rar`. Vyžádali si jen jedno vylepšení u obálky, a to rozšíření okrajů, aby neměli problém s tiskem na spád (do okrajů). To už byla typografická operativa do hodiny.

Byla to fuška, na konferenci jsem však mohl knihu představit s úsměvem na tváři. První tisk byl u velké tiskárny nastaven na 200 kusů, druhý na 1000 kusů.

Několik myšlenek místo Závěru

Musím uznat, že tento knižní projekt znamenal větší část mých příjmů za rok 2015. Proč ne? Ovšem utratil jsem je dokázal téměř obratem za elektrosoučástky a příslušenství k Arduino a Raspberry Pi.

V domácích úkolech mám už roky dostudovat Lua knihovnu `LPeg`, která je ještě obecnější a mnohem silnější na regulární výrazy, ale u produkce jsem s příkazem `unicode.utf8.gsub` (plus jejich případným vnořováním) zatím vystačil.

Tak si s úsměvem říkám, jestli by to už nechtělo zase něco nového, stále se vyvíjející se `LuaTeX` je nuda, no ne? Že by `LATEX3` či `ConTEXt`?

Kontaktní adresa

Ing. Pavel Stríž, Ph.D., Nakladatelství Martin Stríž, U Škol 940, Bučovice, okres Vyškov, CZ-685 01, Česká republika,

E-mailová adresa: `pavel@striz.cz`