

**Fakulta riadenia a informatiky Žilinskej Univerzity
Spoločnosť pre otvorené informačné technológie
klub AMAVET 962
združenie CSTUG**

**OTVORENÝ SOFTVÉR VO VZDELÁVANÍ,
VÝSKUME A V IT RIEŠENIACH**



**Zborník príspevkov medzinárodných
konferencií
OSSConf 2018 a OSSConf 2019**

**2. – 4. júla 2018 Žilina, Slovensko
2. – 4. júla 2019 Žilina, Slovensko**

Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach
2.–4. júla 2018 a 2.–4. júla 2019

Vedecký garant konferencie:

prof. Ing. Karol Matiaško, PhD., Žilinská univerzita, Žilina

Vedecký a programový výbor:

prof. Oleg Černojarov, DrSc., Moscow Power Engineering Institute (RU) – **predseda**

doc. RNDr. Štefan Peško, CSc., Žilinská univerzita, SOIT, Žilina

doc. Ing. Jiří Rybička, Dr., MZLU Brno (CZ)

RNDr. Rudolf Blaško, PhD., Žilinská univerzita, SOIT, Žilina

doc. Ing. Miroslav Kvaššay, PhD., Žilinská univerzita, Žilina

RNDr. Aleš Kozubík, PhD., Žilinská univerzita, SOIT, Žilina

Dr inż. Remigiusz Olejnik, Zachodniopomorski Uniwersytet Technologiczny w Szczecinie (PL)

Ing. Pavel Stříž, Ph.D., Nakladatelství Martin Stříž, Bučovice (CZ)

Ing. Tomáš Majer, PhD., Žilinská univerzita, SOIT, Žilina

Organizačný výbor:

Aleš Kozubík, Žilinská univerzita, SOIT, Žilina – **predseda**

Rudolf Blaško, Žilinská univerzita, SOIT, Žilina

Tomáš Majer, Žilinská univerzita, SOIT, Žilina

Miloslav Ofúkaný, GeoCommunity, SOIT, Bratislava

Pavel Stříž, Nakladatelství Martin Stříž, Bučovice

Peter Štrba, Gymnázium Mikuláša Galandu, SOIT, Turčianske Teplice

Vydavateľ: Žilinská univerzita v Žiline

ISBN 978-80-554-1627-4

Copyright © 2018, 2019 autori príspevkov

Ktokoľvek má dovoľenie vyhotoviť alebo distribuovať doslovný opis tohoto dokumentu alebo jeho časti akýmkoľvek médiom za predpokladu, že bude zachované oznámenie o copyrighte a o tom, že distribútor príjemcovi poskytuje povolenie na ďalšie šírenie, a to v rovnakej podobe, akú má toto oznámenie.

**Fakulta riadenia a informatiky Žilinskej Univerzity
Spoločnosť pre otvorené informačné technológie
klub AMAVET 962
združenie CSTUG**

**OTVORENÝ SOFTVÉR VO VZDELÁVANÍ,
VÝSKUME A V IT RIEŠENIACH**



**Zborník príspevkov medzinárodných konferencií
OSSConf 2018 a OSSConf 2019**

2. – 4. júla 2018 Žilina, Slovensko

2. – 4. júla 2019 Žilina, Slovensko

Recenzenti:

Blaško Rudolf, RNDr., PhD., Borčinová Zuzana, RNDr., PhD., Kaukič Michal, Mgr., CSc., Kovalík Štefan, RNDr., PhD., Kozubík Aleš, RNDr., PhD., Kvaššay Miroslav, doc. Ing., PhD., Majer Tomáš, Ing., PhD., Peško Štefan, doc. RNDr., PhD., Rybička Jiří, doc. Ing., Dr., Varša Peter, RNDr. PhD., Stříž Pavel, Ing., Ph.D.,

Editori:

Rudolf Blaško, Aleš Kozubík

Všetky práce, uverejnené v zborníku, boli posúdené dvomi nezávislými recenzentmi.

Za jazykovú úroveň zodpovedajú autori príspevkov.

Obsah

Úvod	5
Rudolf Blaško Asymptote a L ^A T _E X	7
Michal Chovanec Programovanie v OS Suzuha	15
Aleš Kozubík Automatizácia tvorby tabuliek z dát uložených vo formáte .csv	21
Jiří Rybička Výsledky výuky zpracování textů	27
Rudolf Blaško Použitie rôznych písiem v L ^A T _E X-u I Dynamická, prechodová a statická antikva	33
Rudolf Blaško Použitie rôznych písiem v L ^A T _E X-u II Lineárne, kaligrafické, voľne písané a lomené písma	41
Zdena Dobešová, Michal Kučera Jak se změnil QGIS Processing Modeler ve verzi 3	51
Aleš Kozubík Sadzba aktuárskych symbolov v systéme L ^A T _E X	57
Aleš Kozubík Sadzba tabuliek z tabuľkových procesorov v systéme L ^A T _E X	63
Matej Madeja, Miroslav Biñas, Jaroslav Porubän Automatizácia masovej správy počítačových učební pomocou Ansible	71
Jiří Rybička Zpracování textů a závěrečné práce	79
Abstrakty nerecenzovaných ukážok a prezentácií 2018	89
Abstrakty nerecenzovaných ukážok a prezentácií 2019	91

Vážení čitatelia, priatelia otvoreného softvéru a otvorených technológií,

dostáva sa vám do rúk zborník, ktorý je výstupom samostatnej konferencie, venovanej slobodnému a otvorenému softvéru a jeho využitiu vo vzdelávaní, vede a ostatných oblastiach, ktoré si vyžadujú IT riešenia. S radosťou môžeme konštatovať, že naše konferencie nebadane prehupli už do svojej druhej dekády, čo potvrdzuje životaschopnosť otvorených nástrojov.

Pokúsme sa krátko zrekapitulovať prvú desaťročnicu nášho podujatia. Tá sa z pôvodne jednej zo sekcií podujatia Aplimat vyvinula do podoby samostatnej konferencie. Od svojho prvého ročníka predstavuje predovšetkým neformálne priateľské stretnutie priaznivcov otvoreného softvéru a otvorených technológií z rôznych oblastí života. Gro účastníkov prirodzene tvoria účastníci z oblasti vzdelávania, no postupne si na naše stretnutia našli aj ľudia z oblasti vývoja IT či hospodárskej praxe a verejnej sféry. V snahe priblížiť rozvoj otvorených technológií aj do regiónov, postupne sa od jednej centralizovanej akcie vyčlenili ďalšie podujatia a to predovšetkým vo forme OSS víkendov. Neprehliaditeľným výsledkom boli aj úspešne realizované projekty KEGA zamerané na využívanie otvoreného softvéru vo vzdelávaní a medzinárodný projekt v rámci programu ERASMUS+, ktorého náplňou je príprava spoločných predmetov s podporou open source, určených pre študijné programy troch partnerských univerzít v troch krajinách.

Nie je možné prehliadnuť, že počas tohto možno krátkeho, možno dlhého desaťročia sme zaznamenali aj významný pokrok v šírení a vývoji otvoreného softvéru. Možno aj vďaka nám sa už otvorený softvér dostal do povedomia širšej užívateľskej spoločnosti a tak sa už nikto nepozastavuje nad tým, ak povie, že „fíčite na Linuxe“. Dokonca sa nájdú aj informovanejší, ktorí sa spýtajú na distribúciu. Je málo pravdepodobné, že stretnete užívateľa, ktorý by nepoznal Firefox, bežne používame smartfóny a tablety s Androidom a ani záverečné práce sádzané v TEX-u už nie sú výnimočné. Žiaľ, len v oblasti verejnej a štátnej správy stále zaostávame za svetovým trendom a výskytu OSS vo verejnom živote sú stále skôr vzácnou raritou než bežným javom. Ale aj malé krôčiky v tejto oblasti sú povzbudivé.

Ani po desaťročí nemeníme tradičné, „zabehnuté“ sekcie. Jednou z nich je sekcia „ \LaTeX a jeho priatelia“, ktorá mala svoje stabilné miesto na všetkých doterajších ročníkoch a dlhodobo patrí medzi príspevkovo najbohatšie sekcie. Za svoju životaschopnosť vďačí najmä nesmiernej aktivite a úsiliu jej hlavného organizátora, neúnavného Pavla Stříža. Aj vďaka nemu sa jedným z partnerov konferencie stalo československé združenie užívateľov TeX-u CSTUG.

Druhou takouto pevne zakotvenou sekciou je „Open GIS“, ktorá taktiež doteraz nechýbala ani na jednom ročníku OSSConf. Je to predovšetkým zásluha zapáleného organizátora Mila Ofúkaného, ktorý je zástupcom jedného zo spoluorganizátorov konferencie – klubu AMAVET č.962. Neviem, či je dielom náhody, že táto sekcia vždy vyplnila záverečný deň našich rokovaní, no v každom prípade sa schádzanie jej účastníkov stalo symbolom úľavy pre nás organizátorov, že ďalší ročník skončí úspešne. Dobrým počínom zo starnej jej organizátora bolo aj zaradenie akcie MAPATHON do programu konferencie. Táto si našla ihneď početnú skupinu priznivcov a dokonca viedla aj k viacnásobnému opakovaniu tohto podujatia počas roka aj mimo programu našej konferencie.

Tretou z tradičných sekcií je „OSS vo vede a vzdelávaní“. Túto sekciu je možné označiť za obsahovo najrozmanitejšiu. Vďaka svojej širokospektrálnosti nemala nikdy núdzu o hodnotné príspevky. Po odchode jedného z jej ústredných protagonistov a zakladateľov konferencie, Michala Kaukiča, však nemá jednoznačného lídra tak ako dve vyššie uvedené. V ostatných ročníkoch sa v tejto sekcii, ako zástupca SOIT, angažuje aj Martin Šechný. Verím, že má potenciál stať sa do budúca garantom tejto sekcie.

Rád konštatujem, že v našom programe si našla svoje miesto aj sekcia venovaná otvorenému otvorenému hardvéru. Ukazuje sa, že aj táto oblasť otvorených technológií je dostatočne životaschopnou. Iniciátorom zaradenia tejto sekcie bol Remigiusz Olejnik z poľského Štetína. Atraktivita tejto sekcie je daná najmä skutočnosťou, že väčšina vystúpení je spojená s ukážkou „niečoho hmatateľného“, čo sa aj pohybuje alebo niečo viditeľného robí. Preto sme aj tradičné sprievodné podujatie – seminár pre učiteľov (nielen) informatiky venovali oblasti hardvéru a to jednak so zameraním na atraktívnu a aktuálnu oblasť 3D tlače a tiež na problematiku strojového učenia – deep machine learning.

My organizátori veríme, že každý účastník si medzi uvedenými oblasťami nájde svoje záujmy ako aj dostatok nových podnetov a inšpirácií. Rovnako radi prijmem aj Vaše podnety a nápady o tom, čo by ste radi videli a počuli na našom ďalšom ročníku. Mimoriadne vítané sú ponuky plenárnych prednášok od všetkých, kto môžu našej open source komunite niečo ponúknuť a predviesť.

Záverom by som sa chcel osobitne poďakovať vedeniu Fakulty Riadenia a Informatiky Žilinskej univerzity zastúpenému osobou jej dekana pána doc. Ing. Emila Kršáka, PhD., za dlhoročnú podporu našej konferencie a bezodplatné poskytnutie konferenčných miestností a laboratórií v priestoroch fakulty.

Na úplný záver chcem využiť skutočnosť, že náš zborník vychádza ako postkonferenčný a informovať Vás o pripravovanej letnej škole otvoreného softvéru, ktorú pripravujeme v rámci projektu ERASMUS+ – strategické partnerstvo v týždni od 6. do 10. júla 2020. Zároveň Vás touto cestou rád informujem, že ďalší ročník OSSConf sme naplánovali v termíne 1. – 3. júla 2020.

Za organizačný výbor OSSConf 2018 a OSSConf 2019

Aleš Kozubík
predseda

ASYMPTOTE A \LaTeX

RUDOLF BLAŠKO (SK)

Abstrakt. \TeX , resp. \LaTeX patria medzi najlepšie DTP programy na svete a v prípade odborných textov (matematických, fyzikálnych, chemických ap.) existuje len veľmi málo programov (aj to iba komerčných), ktoré by im mohli konkurovať. Kancelárske programy Office nemôžu \TeX -u konkurovať ani v prípade hladkej sadzby. Aj nakresliť graf matematickej funkcie nie je problém. Existuje mnoho programov, ktoré dokážu grafy pomerne pekne vykresliť. Problém nastáva s exportom a následným použitím v inom dokumente. Problém býva hlavne so sprievodným textom, t. j. použitým písmom. \LaTeX má k dispozícii už niekoľko rokov veľmi silný prostriedok – balíček `TikZ`. Skúsený pisateľ dokáže pomocou balíčka `TikZ` nakresliť prakticky všetko od jednoduchých obrázkov, cez grafy funkcií až po zložité schémy. Iná možnosť pri písaní dokumentov v \TeX -u, resp. \LaTeX -u je použitie programu `Asymptote`, ktorý je súčasťou distribúcie \TeX Live. Nevýhodou je nutnosť viacnásobného prekladu dokumentu (vrátane `Asymptote`). Nepopierateľná pridaná hodnota (pri použití `TikZ`-u aj `Asymptote`) ostáva v možnosti použitia všetkých dostupných prostriedkov \LaTeX -u. V spojení s balíčkom `Animate` môžeme grafy a obrázky vytvorené pomocou `TikZ`-u a `Asymptote` jednoducho animovať.

Kľúčové slová. \TeX , \LaTeX , balíček `TikZ`, `Asymptote`, balíček `Animate`, reálna funkcia, 3D funkcia.

ASYMPTOTE AND \LaTeX

Abstract. \TeX , respectively, \LaTeX belong to the best desktop publishing programs in the world and what is concerning the scientific texts (mathematical, physical, chemical, etc.), there are very few programs (even commercial ones) that could compete with them. The “Office type” programs cannot compete with \TeX even with smooth text typeset. Even drawing a graph of a mathematical function is not a problem. There are many programs that can render graphs quite nicely. The problem arises with the export and subsequent use in another document. The problem is especially with the accompanying text, i.e. font used. \LaTeX contains a very strong tool – the `TikZ` package – for several years now. The experienced writer can draw almost everything from simple pictures, function graphs to complex diagrams using the `TikZ` package. Another option when writing documents in \TeX , respectively, \LaTeX is the `Asymptote` program included with \TeX Live. The disadvantage is the necessity of multiple translations of the document (if it includes `Asymptote`). The undeniable added value (using both `TikZ` and `Asymptote`) remains in the use of all available \LaTeX tools. In conjunction with the `Animate` package, you can easily animate graphs and images.

Keywords. \TeX , \LaTeX , package `TikZ`, `Asymptote`, package `Animate`, real function, 3D function.

Úvod

\TeX , resp. \LaTeX patria medzi najlepšie DTP programy na svete a v prípade odborných textov (matematických, fyzikálnych, chemických, technických ap.), kde sa vyžadujú špeciálne symboly alebo dokonca zložitejšie vzťahy, existuje len veľmi málo programov (aj to iba komerčných), ktoré by im mohli konkurovať. V uvedených odborných kruhoch sú samozrejme aj najčastejšie používané. Ale to sa žiaľ, netýka Slovenska a Česka. Aj keď sú \TeX a \LaTeX zadarmo (Open Source), mnoho autorov uprednostní komerčný MS Office, v lepšom prípade Libre Office. Tieto programy sú výborné hlavne pre sekretárky, prípadne pre hladkú sadzbu. Pre vedecké a odborné texty (matematické, technické, chemické ap.) sú nevhodné a nepraktické. Ak chce niekto písať matematické vzorce a rovnice pomocou týchto kancelárskych programov, dokáže to, ale samozrejme s neporovnateľne väčším úsilím a neporovnateľne nižšou kvalitou.

Ak píšeme odborný text, mnohokrát potrebujeme svoje výsledky ilustrovať grafmi nejakých matematických funkcií. Sú to nielen funkcie jednorozmerné (ich grafy sú 2D), ale aj viacrozmerné s grafmi zobrazenými v 3D priestore. Nakresliť graf takejto matematickej funkcie nie je problém. Existuje mnoho programov (napr. komerčné programy Wolfram Mathematica, Maple, Matlab, Open Source Maxima, resp. wxMaxima, prípadne on line wolframalpha), ktoré dokážu grafy pomerne pekne vykresliť. Problém nastáva s exportom a následným použitím v inom dokumente. Statické grafy sa exportujú pomerne jednoducho, oveľa ťažšie je exportovať animované grafy. V prípade 2D grafov môžeme v uvedenom programe vytvoriť animovaný súbor vo formáte `.gif` a následne ho importovať do \TeX -ovského dokumentu. Oveľa výhodnejšie je v tomto prípade vytvoriť pre každý relevantný záber jeden grafický súbor (najlepšie vo formáte `.pdf`, `.png` alebo `.jpg`) a tie následne spojiť do jediného animovaného obrázka pomocou balíčka `animate`. Takto vytvorené obrázky môžeme pomocou jednoduchého menu ovládať (meniť rýchlosť prehrávania, prehrávať od zadu do predu, pozastaviť prehrávanie ap.). Balíček `animate` veľmi dobre spolupracuje s obrázkami vytvorenými pomocou balíčka `TikZ` a taktiež s obrázkami vytvorenými v programe `Asymptote`. Navyše aj 3D objekty vytvorené pomocou `Asymptote` môžeme tiež animovať (meniť veľkosť, otáčať do rôznych smerov ap.).

1. Asymptote

Program `Asymptote: The Vector Graphics Language` vyvinuli Andy Hammerlindl, John C. Bowman (University of Alberta) a Tom Prince. `Asymptote` je Open Source prostriedok, ktorý je k dispozícii na základe licencie GNU Lesser General Public License (LGPL). `Asymptote` funguje samozrejme nielen pod Linux-om, ale aj pod Mac OS a windows. Je vhodný na kreslenie grafov matematických funkcií a to dvojrozmerných a aj trojrozmerných.

Na domovskej stránke stránke <https://asymptote.sourceforge.io/> nájdeme mnoho užitočných informácií (dokumentácia, zdrojové a inštalačné súbory, galérie 2D a 3D príkladov, index, tutoriál, fórum, odkazy na iné stránky, ...). Môžeme si z tejto stránky stiahnuť inštalačné súbory a **Asymptote** nainštalovať na svoj počítač. **Asymptote** výstižne charakterizuje hneď prvý odstavec na tejto stránke: „**Asymptote** je výkonný popisný jazyk vektorovej grafiky, ktorý poskytuje prirodzené prostredie pre kreslenie technických obrázkov a schém. Popisky a rovnice sú vysádzané pomocou T_EX-u, resp. L^AT_EX-u, de facto štandardu pre sadzbu matematiky.“ Hlavnou výhodou **Asymptote** oproti iným grafickým programom je, že je navyše aj programovacím jazykom. Je inšpirovaný METAPOST-om, syntax jazyka je odvodená zo syntaxe C++. Grafické príkazy sú implementované v samotnom jazyku **Asymptote**, čo umožňuje ich prispôsobenie konkrétnym situáciám. Spomeňme ešte niektoré zaujímavé vlastnosti **Asymptote** (viď <https://asymptote.sourceforge.io/>):

- Generovanie vysoko kvalitnej vektorovej grafiky s možnosťami výstupu do formátov PostScript (.ps, resp. .eps), .pdf, .svg, WebGL, PRC.
- Možnosť vkladania 3D vektorovej grafiky WebGL (Web Graphics Library pre JavaScript) do súborov .html.
- Možnosť vkladania 3D vektorovej grafiky PRC (Product Representation Compact) do súborov .pdf.

Ak máme nainštalovaný T_EX a neplánujeme **Asymptote** samostatne používať, potom ho nemusíme samostatne inštalovať, pretože je štandardnou súčasťou inštalácie T_EX-u a veľmi dobre spolupracuje s L^AT_EX-om pomocou balíčka **asymptote**. Pre nás je interesantné, že 3D obrázky vytvorené pomocou **Asymptote** a exportované do súborov .pdf, môžeme rôzne otáčať, zmenšovať ap. Dvojrozmerné obrázky dokážeme animovať pomocou balíčka **animate**. Balíčkom **animate** sa zaoberal P. Stríž na OSSConf 2010 v príspevku „Jak jsem se skamarádil s L^AT_EXovým balíčkom **animate**“ [4], pričom autor obrázky vytváral na úrovni METAPOST-u.

2. Asymptote a T_EX

Najväčšia výhoda **Asymptote** je už spomínaná spolupráca a prepojenie s T_EX-ovým, resp. L^AT_EX-ovým dokumentom. Po vložení príkazu

```
\usepackage[inline]{asymptote}
```

do preambuly L^AT_EX-ového zdrojového dokumentu môžeme **Asymptote** scripty písať priamo do textu do prostredia **asy**:

```
\begin{asy}
    zdrojový text pre Asymptote
\end{asy}
```

príčom za príkazom `\begin{asy}` nesmie byť prázdny riadok a príkaz `\end{asy}` musí byť na samostatnom riadku a musí byť na riadku jediný (ani T_EX-ovská

poznámka pomocou % nemôže byť za ním). L^AT_EX dokument sa prekladá na tri krát, preto je vhodné si vytvoriť script na takýto preklad. Napríklad, ak sa náš zdrojový súbor nazýva `subor.tex`, potom preklad by mal prebiehať nasledovne:

```
pdflatex subor.tex
asy subor-*.asy
pdflatex subor.tex
```

Prvý krok, preklad pomocou `pdflatex-u`, vytvorí v adresári súboru `subor.tex` súbory `subor-1.asy`, `subor-2.asy`, ... (ku každému prostrediu `asy` jeden súbor) a súbor `subor.pre`.

Druhý krok, preklad pomocou `Asymptote`, vytvorí v uvedenom adresári súbory `subor-1.tex`, `subor-2.tex`, ..., `subor-1.pre`, `subor-2.pre`, ..., a pomocné súbory `_subor-1.pdf`, `_subor-2.pdf`, ..., `subor-1_0.pdf`, `_subor-2_0.pdf`, Pri vytváraní 3D grafiky sa vygenerujú aj iné súbory, napr. `subor-?+0.prc`, `subor-?+0_0.pdf`, `subor-?_0.pbsdat`, `subor.out`.

Po druhom kroku je súbor `subor.pdf` nefunkčný, preto musíme urobiť ešte jeden preklad pomocou `pdflatex-u`. Ak sa pri písaní dokumentu údaje v prostrediach potrebných pre `Asymptote` nezmenia, preklad pomocou `asy` nemusíme robiť a postačí jeden preklad pomocou `pdflatex-u`.

Externý súbor `asysubor.asy` so zdrojovým kódom `Asymptote` môžeme do L^AT_EX-ovského dokumentu vložiť pomocou príkazu

```
\asyinclude[<voľby>]{asysubor.asy}
```

Druhá možnosť ako používať L^AT_EX s implementovanými zdrojovými kódmi `Asymptote` je použitie utility `latexmk` zo stránky <http://mirror.ctan.org/support/latexmk/>.

Nastavenia potrebné pre `Asymptote`, ktoré používame vo viacerých prostrediach `asy`, nemusíme písať samostatne v každom z nich. Môžeme ich definovať na začiatku (môžeme aj v preambule) v prostredí `asydef`, napr.

```
\begin{asydef}
  texpreamble("\usepackage[math]{iwona}");
  real cc=1.5; //hodnota cc > 0
  import graph;
\end{asydef}
```

Príkaz `texpreamble("\usepackage[math]{iwona}");` zariadi, že v celom dokumente, vrátane prostredí `asy` nastaví sanserifový font `iwona` (viď <https://tug.org/FontCatalogue/iwona/>). Ak tento príkaz použijeme iba v nejakom konkrétnom prostredí `asy`, tieto fonty sa nastavia v celom dokumente a iba v tomto prostredí `asy` a v ostatných prostrediach `asy` sa nenastavia. Ak dáme do preambuly príkaz `\usepackage[math]{iwona}` fonty sa nastavia iba v celom dokumente, ale nie v prostrediach `asy`.

Príkaz `import graph;` načíta modul `graph` na kreslenie grafov reálnych funkcií. Týchto modulov (Base modules) je okolo 40 (viď <https://asymptote.>

sourceforge.io/doc/, resp. súbor `asymptote.pdf`, ktorý dostaneme zadaním `texdoc asymptote` na príkazovej úrovni – v oboch prípadoch „Kapitola 8: Base modules“) a do prostredia `asy` sa načítavajú príkazom `import`. Základný modul je `plain`, ktorý je nastavený implicitne. Na animovanie dvojrozmerných obrázkov používame modul `animation`. Na vykresľovanie trojrozmerných grafov používame modul `graph3`. Z ostatných modulov spomeňme napríklad `interpolate` (modul implementuje Lagrangeovú, Hermitovú a štandardnú interpoláciu kubických spline do `Asymptote`), `geometry` (rôzne geometrické symboly, štruktúry, krivky ap.), `stats` (rôzne štatistické funkcie), `patterns` (rôzne vzory na vypĺňanie plôch), `markers` (rôzne možnosti na označovanie vrcholov, hrán, ciest, uhlov ap.), `tree`, `binarytree`, `drawtreetree` (binárne a jednoduché stromy), `palette` (rôzne nastavena farieb), `three` (možnosť konverzie dvojrozmerných kriviek na trojrozmerné a naopak), `grid3` (trojrozmerné mriežky), `flowchart` (vývojové diagramy), `contour` (vrstevnice).

```

\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage[inline]{asymptote}

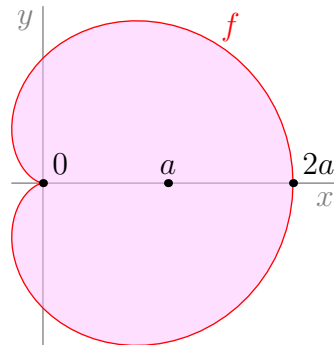
\begin{document}
\pagestyle{empty}
\begin{asy}
  import graph;size(0,5cm);

  xaxis("$x$",above=true,p=gray);
  yaxis("$y$",above=true,p=gray);

  real f(real t){return 1+cos(t);}
  path g=polargraph(f,0,2pi)--cycle;
  filldraw(g,pink+opacity(.5),drawpen=red);
  label("\large$f$",(1.5,1.25),red);

  dot("$0$",(0,0),N+E);
  dot("$a$",(1,0),N);dot("$2a$",(2,0),N+E);
\end{asy}
\end{document}

```



Obr. 1. Kardioda $f: \rho(\varphi) = a(1 + \cos \varphi)$, $\varphi \in \langle 0, 2\pi \rangle$, $a > 0$.

Rovnica krivky zvanej kardioda (srdcovka) má v polárnych súradniciach rovnicu $\rho(\varphi) = a(1 + \cos \varphi)$, $\varphi \in \langle 0, 2\pi \rangle$, $a > 0$. Táto krivka sa dá taktiež zostrojiť ako epicykloida, t. j. rovinná krivka, ktorú pri kotúlení kružnice k po vonkajšej strane pevne danej kružnice K opisuje daný bod B , ktorý je pevný vzhľadom na k . Kardiodidu dostaneme, ak majú obe kružnice rovnaký polomer a bod B leží na obvodě kružnice k . Na obr. 1 je kardioda vykreslená pomocou `Asymptote`.

Modul `animation` umožňuje generovať animované obrázky. Tieto animácie využívajú prevodný program `ImageMagick` na zlúčenie viacerých obrázkov do súborov vo formáte `.gif` alebo `.mpeg`. Vhodnejšie je použiť odvodený modul `animate`, ktorý generuje vyššiu kvalitu obrázkov vo formáte `.pdf` s možnosťami ovládania. Tento spôsob je podmienený inštaláciou balíčka `animate.sty` v `LATEX`-u a použiť ho môžeme v prostredí `asy` nasledovne:

```
usepackage("animate");
settings.tex="pdflatex";
defaultpen(.25);
import animation;
```

Ak animovanú časť označíme `animation a`; , potom ovládacie menu animácie dostaneme napríklad príkazom

```
label(a.pdf(delay=250,"buttonsize=10pt,controls,autoplay,
loop,palindrome",multipage=false));
```

Hypocykloida je rovinná krivka, ktorú pri kotúľaní kružnice k po obvodov vnútornej strany pevne danej kružnice K (kružnica k je vo vnútri kružnice K) opisuje daný bod B , ktorý je pevný vzhladom na k . Ak označíme $R > 0$ polomer kružnice K , $r > 0$ polomer kružnice k ($r < R$), $c \geq 0$ vzdialenosť bodu B od stredu kružnice k , potom môžeme hypocykloidu parametricky vyjadriť v tvare

$$x = (R-r) \cos \frac{rt}{R} + c \cos \frac{(R-r)t}{R}, \quad y = (R-r) \sin \frac{rt}{R} - c \sin \frac{(R-r)t}{R}, \quad t \in R.$$

Pohyb bodu B pre prípad $R = \frac{5r}{3}$, $c = \frac{3r}{2}$ je znázornený na obr. 2. Keďže sú pomery $\frac{r}{R} = \frac{3}{5}$, $\frac{(R-r)}{R} = \frac{2}{5}$ racionálne čísla, funkcie sínus a kosínus sú periodické s periódou 2π , pohyb sa uzavrie po piatich okruhoch, t. j. po hodnote $t = 10\pi$. Uvedená hypocykloida je periodická s periódou 10π a má tvar

$$x = \frac{2r}{3} \cos \frac{3t}{5} + \frac{3r}{2} \cos \frac{2t}{5}, \quad y = \frac{2r}{3} \sin \frac{3t}{5} - \frac{3r}{2} \sin \frac{2t}{5}, \quad t \in \langle 0, 10\pi \rangle.$$

Na začiatku zdrojového kódu sú definované základné hodnoty $c = cc$, $R = rv$, $r = rm$, rt koniec definičného intervalu, n počet vykresľovaných bodov grafu.

```
\begin{asy}
real cc=1.5,u=5,v=3,rv=u/v,rm=1,rt=2*u,rp=rv-rm;int n=90;
import graph;
usepackage("animate");settings.tex="pdflatex";defaultpen(.25);import animation;
size(0cm,5cm);
pair wheelpoint(real t)
{return (rp*cos(t*rm/rv)+cc*cos(rp*t/rv),rp*sin(t*rm/rv)-cc*sin(rp*t/rv));}
guide wheel(guide g=nullpath,real a,real b,int n)
{real width=(b-a)/n;for(int i=0;i<=n;++i)
{real t=a+width*i;g=g--wheelpoint(t);} return g;}
real tinterval=rt*pi,t1=0,t2=t1+tinterval;draw(circle((0,0),rv),olive+.75);
animation a;
pair z1=wheelpoint(t1);dot(z1,red);real dt=(t2-t1)/n;
for(int i=0;i<=n;++i) {save();
real t=t1+dt*i;real kx=rp*cos(rm*t/rv);real ky=rp*sin(rm*t/rv);
filldraw(circle((kx,ky),cc),.2paleblue+white,.2paleblue+white+.5);
```

```

draw((0,0)--(rv*cos(rm*t/rv),rv*sin(rm*t/rv)),lightblue);
if (t>0) {filldraw((kx,ky)--arc((kx,ky),rm,180*rm*t/rv/pi,
    -180*rp*t/rv/pi)--cycle,white+.75blue+opacity(.25),drawpen=lightblue);}
draw(circle((0,0),rv),olive+.75);label("$K$",(-.6*rv,-.75*rv),SW,olive);
draw(circle((0,0),rp),dotted+blue+white);
draw(circle((0,0),rp-cc),yellow+.35red);draw(circle((0,0),rp+cc),yellow+.35red);
    label("$x$", (rv+.25,0),N);draw((-rv-.25,0)--(rv+.25,0));
    label("$y$", (0,rv+.25),W);draw((0,-rv-.25)--(0,rv+.25));
draw(wheel(t1,t2,8*n),dotted+red);draw(circle((kx,ky),rm),blue+.75);
label("$k$", (kx-.6,ky-.75),SW,blue);draw((kx,ky)--wheelpoint(t),black+.625);
    dot((kx,ky));dot(wheelpoint(t),red+black);draw(wheel(t1,t,8*max(1,i)),red+.5);
    label("\scriptsize$t="+string(t,7)+"$", (.3*rv,-rv),SE,blue);
a.add();restore();} erase();
label(a.pdf(delay=250,"buttonsize=10pt,controls,autoplay,loop,
    palindrome",multipage=false));
\end{asy}

```

Obr. 2. Príklad animácie hypocykloidy

Na záver uvedieme príklad trojrozmerného grafu vytvoreného v **Asymptote**. Na obr. 3 je znázornený graf funkcie $z = \sin x \cdot \sin y$ na intervale $\langle -\pi, \pi \rangle \times \langle -\pi, \pi \rangle$.

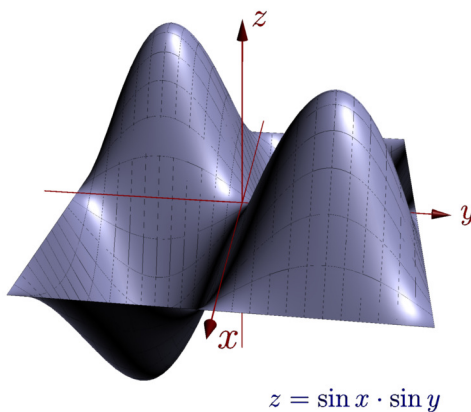
Záver

Iná možnosť na kreslenie je použitie balíčka **TikZ** (T ist kein Zeichenprogramm). ktorý je veľmi účinný nástroj na kreslenie jednoduchých aj zložitejších obrázkov, grafov, schém, grafov funkcií... Jeho autorom je Till Tantau. Základným princípom práce s týmto balíčkom sa zaoberal A. Kozubík na OSSConf 2012 v príspevku „Naučím vás kresliť alebo predstavenie balíčka TikZ“ [3]. Balíček je štandardnou súčasťou každej inštalácie T_EX-u. Manuál a mnoho užitočných informácií nájdeme na portáli CTAN <https://www.ctan.org/pkg/pgf>.

```

\begin{asy}
import graph3;import palette;
size(175,175,IgnoreAspect);
currentprojection=perspective(2,.55,0.25);
real f(pair z){return sin(z.x)*sin(z.y);}
draw(surface(f,(-pi,-pi),(pi,pi),nx=24,
    Spline),paleblue,render(merge=true));
surface f=surface(f,(-pi,-pi),(pi,pi),
    24,Spline);
draw(f,mean(palette(f.map(zpart),
    Rainbow(24))),brown);
draw(Label("$x$",1),
    (-1.2pi,0,0)--(1.2pi,0,0),brown,Arrow3);
draw(Label("$y$",1),
    (0,-1.2pi,0)--(0,1.2pi,0),brown,Arrow3);
draw(Label("$z$",1),
    (0,0,-1.2)--(0,0,1.2),brown,Arrow3);
label("$z=\sin{x}\cdot\sin{y}$",
    (.5,.5,-1.35),SE,deepblue);
\end{asy}

```



Obr. 3. Graf funkcie $z = \sin x \cdot \sin y$, $(x, y) \in \langle -\pi, \pi \rangle \times \langle -\pi, \pi \rangle$

Podakovanie. Príspevok vznikol s príspevom grantu KEGA 011–4/2014ŽU „Experimentálna matematika – zviditeľnenie neviditeľného“ podporeného Slovenskou kultúrno-edukačnou grantovou agentúrou.

Literatúra

- [1] HAMMERLINDL, A., BOWMAN, J., PRINCE, T.: *Asymptote: The Vector Graphics Language*, <https://asymptote.sourceforge.io/>.
- [2] KNUTH, D. E.: *The TeXbook*, Volume A of *Computers and Typesetting*, Addison-Wesley Publishing Company (1984), ISBN 0-201-13448-9.
- [3] KOZUBÍK, A.: *Naučím vás kresliť alebo predstavenie balíčka TikZ*, Zborník príspevkov medzinárodnej konferencie OSSConf 2012, 2.–4. júla 2012, Žilina, str. 91–96. ISBN 978-80-970457-2-2.
- [4] STRÍŽ, P.: *Jak jsem se skamarádil s LaTeXovým balíčkom animate*, Zborník príspevkov medzinárodnej konferencie OSSConf 2010, 1.–4. júla 2010, Žilina, str. 177–184. ISBN 978-80-970457-0-8.
- [5] TANTAU, T.: *TikZ & PGF*, Manual for Version 3.1.3, <http://mirrors.ctan.org/graphics/pgf/base/doc/pgfmanual.pdf>.

Kontaktná adresa

RNDr. Rudolf Blaško, PhD., Katedra matematických metód a operačnej analýzy, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko,
E-mailová adresa: beerb@frcatel.fri.uniza.sk

PROGRAMOVANIE V OS SUZUHA

MICHAL CHOVANEC (SR)

Abstrakt. Článok popisuje implementáciu podsystemu správ v OS Suzuha. Technika programovania pomocou správ - udalostí umožňuje efektívne využívať procesorový čas nezávisle od použitého hardvéru. Udalosti zovšeobecňujú hardvérové prerušenia a komunikáciu pomocou správ.

Kľúčové slová. operačný systém, systém reálneho času, udalosťami riadené programovanie, arm cortex.

PROGRAMMING IN SUZUHA OS

Abstract. This paper describes the implementation of the message subsystem in OS Suzuha. Techniques of programming by messages - events make efficient use of processor time independently of the hardware. Events generalize hardware interrupts and communicate via messages.

Keywords. operating system, real time system, event driven programming, arm cortex.

Úvod

Riadenie procesu pomocou mikrokontroléra vyžaduje včasnú reakciu na zmenu vstupných veličín. V súčasnosti má takmer každý riadiaci systém viac vstupov i výstupov. Dáta z jednotlivých vstupov prichádzajú v rôznych časových intervaloch (periodicky i aperiodicky) a podobne aj výstupy vyžadujú rôzne časovanie.

Jedným z riešení ako spracovať takéto informácie je implementácia udalosťami riadeného programovania. Mikrokontroléry k tomu poskytujú rozsiahly systém prerušení. Táto časť je však hardvérovo závislá a pri migrácii medzi rôznymi typmi mikrokontrolérov predstavuje často rozsiahle zmeny v programe.

Problém je možné riešiť zavedením podsystemu správ ako súčasť operačného systému. Toto riešenie zovšeobecní riadenie prerušeniami a medzivláknovú komunikáciu do jednotného udalostne riadeného systému.

Systém správ teda poskytuje vhodný prostriedok abstrakcie od hardvéru.

1. Popis OS Suzuha

Operačný systém Suzuha je open source systém reálneho času určený pre mikrokontroléry ARM Cortex. V súčasnosti bol systém testovaný na troch rôznych architektúrach:

- ARM Cortex M0: mikrokontrolér KL25Z128.
- ARM Cortex M3: mikrokontrolér STM32F100.
- ARM Cortex M4: mikrokontrolér TM4C123G.

Prípadná prenositeľnosť systému medzi ďalšími radami Cortex (Mx, Rx, Ax) je bezproblémová.

Systém je písaný striktno modulárne so zreteľom na jednoduchosť. Nesnaží sa zavádzať revolučné riešenia.

Systém je rozdelený do niekoľkých vrstiev

- Kernel: jadro systému.
- Systémové knižnice: ovládače hardvéru, mutexy, systém správ a štandardný výstup.
- Aplikáčne knižnice: knižnice závislé od aplikácie, tvorí ich používateľ.
- Aplikácia: vlastná používateľská aplikácia.

OS Suzuha je možné stiahnuť pod GNU GPL licenciou z [1]. Pre ďalší vývoj je vhodné začať jednoduchým jadrom Cortex M0, jeho popis je možné nájsť v [2].

2. Systém správ

Medzivláknová komunikácia pomocou správ je založená na modeli klient-server. Server sa registruje v systéme volaním:

```
msg_register(MSG_SERVER);
```

kde MSG_SERVER je jedinečné 32bitové číslo identifikujúce server. Tento identifikátor je nevyhnutný pre pripojenie sa k serveru.

Vláknko servera po registrovaní obvykle čaká na príjem správy:

```
while (1)
{
    struct sMsg msg;
    msg_get(&msg);
    ...
}
```

Pri čakaní na správu sa vláknko prepne do stavu WAITING a jadro systému s nim pri plánovaní ďalšieho procesu nepočíta. Spiace vlákna sú jadrom ignorované, šetrí sa tak výpočtový výkon a spotreba.

Vláknko klienta posielajú správy volaním

```
msg_raise(&msg);
```

Naplnenie štruktúry sMsg a odoslanie je implementované nasledovne:

```
while (1)
{
    ...

    struct sMsg msg;
```

```

msg.destination = MSG_SERVER; /* ID servera = prijemca */
msg.source = KLIENT_ID; /* ID odosielateľa */
msg.size = 4; /* veľkosť správy v bytoch */
msg.data = data; /* vlastná sprava */
msg_raise(&msg); /* poslanie správy */
}

```

Naplnením štruktúry `struct sMsg` adresou servera je zabezpečené doručenie správy k serveru. Funkcia `msg_raise(struct sMsg *msg)` čaká na prijatie správy serverom, je tak zabezpečené kompletne prijatie správy. Zo strany servera môže byť zaslaná odpoveď. Vlákno klienta však môže predčasne ukončiť činnosť, alebo môže dôjsť k jej zlyhaniu. K dispozícii je preto funkcia `msg_raise_async(struct sMsg *msg)`, ktorá nečaká na príjem - server preto nezostane čakať a môže obsluhovať ďalšie správy vo fronte.

3. Implementácia systému správ

Vlastná implementácia systému správ je písaná s ohľadom na dostupné technické prostriedky. Limitujúcim faktorom je najmä pamäť RAM. Spomenuté mikrokontroléry majú od 4KB po 32KB RAM pamäte.

V systéme je preto implementovaná len jedna fronta správ, spoločná pre všetky vlákna. Vzhľadom na modularitu však nie je problém nahradiť doterajšie riešenie viacfrontovým systémom, známym z operačných systémov pre PC. Štruktúra správy je nasledovná:

```

struct sMsg
{
    u32 destination, source;
    u32 data;
    u32 size;
};

```

Položky *destination* a *source* predstavujú ID cieľa a zdroja. Položka *data* predstavuje 32 bitové číslo, nesúce vlastnú správu. Premenné sú typu *u32* - obvykle odvodený od *unsigned int*. Podľa potreby je možné pretypovať ukazovateľ na pole dát a prenášať tak ľubovoľný objem dát. Nevyhnutné je potom správne nastaviť položku *size*:

```

data = (u32)bytes_array;
size = sizeof(bytes_array)/sizeof(u32);

```

Týmto spôsobom je prenášaný len ukazovateľ do zdieľanej pamäte. Je preto nevyhnutné aby odosielateľ nenechal obsah dát po odoslaní kým nedostane potvrdenie zo strany servera.

Pre efektívne využitie systému správ viacerými servermi je implementovaná fronta:

```

volatile struct sMsg __msg__[MSG_FIFO_SIZE];

```

Pre odosielanie správy je najdôležitejšia funkcia

```
u32 msg_raise_async(struct sMsg *msg)
```

Je nevyhnutné aby táto funkcia prebehla atomicky - nielen na logickej úrovni, ale aby skutočne počas jej vykonávania nedošlo k prepnutiu kontextu. V opačnom prípade môže dôjsť k predčasnému ukončeniu niektorého vlákna alebo zaplneniu fronty správ.

Najprv je nevyhnutné overiť, či existuje príjemca správy. Ten je overený prehľadom dostupných identifikátorov v poli `__msg_names__`:

```
i = 0;
while ((i < THREADS_MAX_COUNT) && (msg->destination != __msg_names__[i]))
    i++;
```

Po nájdení a overení príjemcu dôjde k samotnému zaradeniu správy do fronty:

```
/*najdenie volnej polozky vo fronte*/
for (i = 0; i < MSG_FIFO_SIZE; i++)
    if (__msg__[i].source == MSG_NULL)
    {
        /*vlastny prenos spravy*/
        __msg__[i].source = msg->source;
        __msg__[i].destination = msg->destination;
        __msg__[i].data = msg->data;
        __msg__[i].size = msg->size;

        /*navrat z funkcie*/
        sched_on();
        wake_up_threads();
        return MSG_SUCESS;
    }
```

Veľkosť fronty správ `__msg__` je volená pri kompilácii systému. Po nájdení prvej voľnej položky je vykonaný proces odoslania správy. Samotný návrat z funkcie vyžaduje ešte dva kroky:

`sched_on()`: odomknutie plánovača - povolenie prerušení,

`wake_up_threads()`: zobudenie spiacich vlákien.

Funkcia však môže byť predčasne ukončená a k doručeniu správy nedôjde. Najčastejšie z dôvodu neexistujúceho príjemcu alebo preplnenia fronty správ. Z toho dôvodu je implementovaná funkcia, ktorá garantuje zaradenie správy do fronty:

```
u32 msg_raise(struct sMsg *msg).
```

Funkcia čaká na uvoľnenie miesta vo fronte. K predčasnemu ukončeniu dôjde len ak príjemca správy neexistuje. Samotný proces čakania je riešený vynútením prepnutia kontextu na ďalšie vlákno - nie je opodstatnené cyklicky testovať hodnotu ktorá sa môže zmeniť len v ďalšom vlákne:

```
/*skusi poslat spravu*/
msg_res = msg_raise_async(msg);
```



```

/*prijemca neexistuje*/
if (msg_res == MSG_NO_REGISTERED)
    return msg_res;

/*fronta plna, vynutenie prepnutia kontextu*/
if (msg_res != MSG_SUCESS)
    sched_next();

```

Funkcia pre príjem správy:

```
void msg_get(struct sMsg *msg)
```

Čakanie na príjem správy prebieha v cyklickom prehladávaní fronty správ. Po nájdení položky patriacej aktuálnemu vláknu sa vykoná samotné prebratie správy:

```

if ( __msg__[i].destination == __msg_names__[tid] )
{
    /*prenos spravy z fronty na vystup*/
    msg->source = __msg__[i].source;
    msg->destination = __msg__[i].destination;
    msg->data    = __msg__[i].data;
    msg->size    = __msg__[i].size;

    /*vycistenie polozky vo fronte*/
    __msg__[i].source    = MSG_NULL;
    __msg__[i].destination = MSG_NULL;
    __msg__[i].data      = MSG_NULL;
    __msg__[i].size      = MSG_NULL;

    sched_on();
    return;
}

```

Funkcia preberie správu z fronty a uvoľní miesto pre ďalšiu správu. Funkcia zistí meno aktuálneho vlákna volaniami:

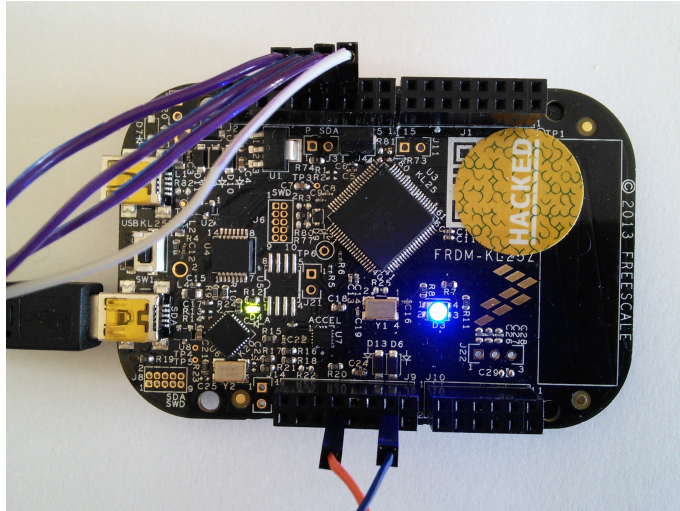
```

u32 tid = get_thread_id();
destination_name = __msg_names__[tid];

```

kde `get_thread_id()` je volanie jadra systému na získanie ID vlákna. Identifikátor `tid` sa preloží volaním `destination_name = __msg_names__[tid]`; na meno príjemcu správy. Tento zdvojený prístup je zvolený z toho dôvodu, že užívateľ vo všeobecnosti nemá kontrolu nad tým, aké `tid` dostane vlákno.

Čakanie na príjem správy je zefektívnené podobne ako jej odoslanie: po jednom cykle prehladávania fronty správ je vynútené prepnutie kontextu.



Obr. 1. Testovacia doska s mikrokontrolérom Kinetis Cortex M0+ [3]

Záver

Model systému na architektúre klient-server predstavuje robustné a modułárne riešenie pre spracovanie dát v procese riadenia. Systém tak dokáže pracovať aj s výpadkom niekoľkých zdrojov dát.

Kritickou časťou systému správ je požiadavka na stabilitu serverovej časti. Ďalšie smerovanie vývoja preto môže byť návrh decentralizovaného prístupu a možnosť posielat' správy nielen v rámci jedného mikrokontroléra, ale pomocou komunikačného rozhrania (RF, IRDA, SPI...) posielat' správy vzdialenému mikrokontroléru.

Literatúra

- [1] Zdrojové kódy OS suzuha: <http://sourceforge.net/projects/suzuhaos/files>
- [2] Popis jadra Cortex M0: http://infocenter.arm.com/help/topic/com.arm.doc.dui0497a/DUI0497A_cortex_m0_r0p0_generic_ug.pdf
- [3] Testovacia doska: http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FRDM-KL25Z

Kontaktná adresa

Ing. Michal Chovanec, Katedra technickej kybernetiky, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská republika,

E-mailová adresa: michal.nand@gmail.com

AUTOMATIZÁCIA TVORBY TABULIEK Z DÁT ULOŽENÝCH VO FORMÁTE .csv

ALEŠ KOZUBÍK (SK)

Abstrakt. Tabulky sú súčasťou väčšiny odborných alebo vedeckých textov. Obvykle bývajú spracovávané v niektorom z tabulkových procesorov. Ich manuálny prenos do cieľového dokumentu tak býva zdrojom mnohých chýb. Príspevok predstavuje balíček `csvsimple`, ktorý umožňuje určitú automatizáciu pre sadzbu tabuliek z dát, exportovaných do súboru vo formáte `csv`.

Kľúčové slová. sadzba, tabulky, `.csv` formát, balíček `csvsimple`.

TYPESSETTING AUTOMATION OF THE THE TABLES CONTAINING DATA STORED IN THE .CSV FORMAT

Abstract. The tables are part of the most scientific texts. Their content is usually processed in some of the spreadsheets. Their manual transmission in the target document is thus the source of many mistakes. The article represents the `csvsimple` package that allows some automation for table typesetting from data exported to a `csv` format.

Keywords. typesetting, tables, `.csv` format, `csvsimple` package.

Úvod

Prakticky žiadny odborný alebo vedecký text sa nezaobíde bez sprievodného tabulkového materiálu. Sadzba tabuliek je pri tom jednou z najnáročnejších súčastí sadzby textu. Je to zapríčinené nielen pomerne náročným spôsobom formátovania tabuliek, ale často aj samotným procesom zostavenia tabulky resp. prenosu jej obsahu do cieľového dokumentu. Práve tento proces býva zdrojom najpočetnejších a pri tom obtiažne odhaliteľných chýb.

Samotné tabulky, najmä ak údaje v nich obsiahnuté sú predmetom či výsledkom rozmanitých výpočtov, je potrebné manuálne opisovať do cieľového dokumentu. Neexistuje totiž nijaká možnosť automatizovaného exportu údajov z tabuliek, či už z tabulkových procesorov, alebo webových zdrojov priamo do formátu typografického systému $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ a ich následného pohodlného vloženia do cieľového dokumentu.

So samotnou tvorbou tabuliek v typografickom systéme $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ je možné sa stručne zoznámiť v článku [2] z predchádzajúcich ročníkov našich konferencií. Podrobnejší prehľad sadzby tabuliek je možné nájsť v knihách [4], alebo [3].

V tomto príspevku predstavíme balíček `csvsimple`, ktorý umožňuje určité zjednodušenie a zmechanizovanie prenosu dát do tabuľky z dátových súborov vo formáte `.csv`. Nakoľko ide o pomerne frekventovaný formát ukladania dát a väčšina tabuľkových procesorov disponuje možnosťou exportu dát do tohto formátu, tento balíček predstavuje veľmi užitočnú pomôcku s potenciálom výraznej redukcie chýb pri prepise.

1. Predstavenie balíčka `csvsimple`.

Balíček `csvsimple` je pomerne mladým balíčkom (k dispozícii je od 1.7.2016), ktorého autorom je Thomas F. Sturm. Tento balíček poskytuje jednoduché rozhranie systému `LATEX`, pre spracovanie `.csv` súborov. Balíček silne vychádza zo syntaxe `pgfkeys`, čoho dôsledkom je jeho pomerne jednoduché použitie. Osobitne treba uviesť podporu filtrovania dát a generovania tabuľky. Nakoľko balíček je stavaný ako ľahký nástroj, neposkytuje podporu pre triedenie dát a ukladanie údajových databáz.

Balíček sa do zdrojového súboru vkladá štandardným spôsobom, teda rovnako ako všetky ostatné balíčky príkazom:

```
\usepackage{csvsimple}
```

Zároveň s balíčkom `cvssimple` sú automaticky nahrávané aj balíčky `pgfkeys`, `etoolbox` a `ifthen`. Pri práci sú často potrebné aj balíčky `longtable` a `booktabs`, ktoré však automaticky nahrávané nie sú a je potrebné doplniť ich manuálne.

1.1. Základy práce

Ako sme už povedali, balíček spracúva súbory vo formáte `.csv`, teda údaje oddelené čiarkami. Pre správnu činnosť je potrebné zabezpečiť, aby každý riadok `csv` súboru obsahoval rovnaký počet údajov oddelených čiarkou. Pri tom je možné, aby boli pomocou skupinových `TEX`-ových zátvoriek `{}` vyznačené bloky obsahujúce čiarky, ktoré nemajú byť spracované ako separátory. To môže byť užitočné, pri spracovaní číselných údajov, kedy podľa zvyklostí našej notácie používame desatinné čiarky, narozdiel od anglo-saskej notácie pomocou desatinnej bodky. Tento „zátvorkový diskonfort“ je tiež možné odstrániť voľbou iného oddeľovača, čo väčšina tabuľkových procesorov umožňuje. Ako uvidíme neskôr, aj balíček `csvsimple` umožňuje nastavenie oddeľovačov na iný znak.

Štruktúra súboru `.csv`, je obvykle, aj keď nie vždy, taká, že prvý tzv. hlavičkový riadok obsahuje názvy jednotlivých stĺpcov. Ilustruje to nasledujúci výpis súboru `loko.csv`, ktorý používame na ilustráciu činnosti balíčka. Jeho obsahom sú vybrané údaje o niekoľkých parných lokomotívach, zosumarizované podľa [5].

```

Číslo,Prezývka,Hmotnosť,Výkon
387.0,Mikádo,93 ton,1546 kW
498.1,Albatros,113 ton,1950 kW
556.0,Štokr,99 ton,1472 kW
464.2,Rosnička,112 ton,1250 kW

```

Z týchto údajov je možné automatizovane zostaviť tabuľku pomocou jednoduchého príkazu:

```
\csvautotabular{loko.csv}
```

Odkaz `loko.csv` je samozrejme možné nahradiť vlastným názvom príslušného csv súboru, z ktorého má byť tabuľka vygenerovaná. Výsledkom je potom tabuľka 1.

Číslo	Prezývka	Hmotnosť	Výkon
387.0	Mikádo	93 ton	1546 kW
498.1	Albatros	113 ton	1950 kW
556.0	Štokr	99 ton	1472 kW
464.2	Rosnička	112 ton	1250 kW

Tabuľka 1. Základná tabuľka vytvorená pomocou balíčka `csvsimple`.

Číslo	Prezývka	Hmotnosť	Výkon
387.0	Mikádo	93 ton	1546 kW
498.1	Albatros	113 ton	1950 kW
556.0	Štokr	99 ton	1472 kW
464.2	Rosnička	112 ton	1250 kW

Tabuľka 2. Základná tabuľka vytvorená v spolupráci s balíčkom `booktabs`.

Podobne je možné zautomatizovať aj spoluprácu s balíčkom `booktabs`. Použijeme však mierne modifikovaný príkaz

```
\csvautobooktabular{loko.csv}
```

Výsledok vidíme v tabuľke 2.

1.2. Riadenie vzhľadu tabuľky

V predchádzajúcich jednoduchých ukážkach bolo formátovanie tabuliek plne v kompetencii balíčka `csvsimple`. Ak chceme prevziať plnú kontrolu nad formátovaním tabuľky, použijeme príkaz `\csvreader`. Nasledujúca ukážka ilustruje, ako sa položky z hlavičkového riadku csv súboru automaticky priradia \TeX -ovým makrám, ktoré môžeme následne voľne použiť. Dôležitá je voľba `[head to column names]` príkazu `\csvreader`. Taktiež treba upozorniť, že bola potrebná určitá úprava csv súboru na súbor `loko1.csv`, v ktorom boli odstránené diakritické znaky, ktoré ako pomenovania makier spôsobujú problémy. Príslušný zdrojový kód potom vyzerá takto:

```

\begin{tabular}{|l|l|l|l|}\hline%
\bfseries Číslo & \bfseries Prezývka & \bfseries Hmotnosť &
\bfseries Výkon
\csvreader[head to column names]{loko1.csv}{%

```

```
{\\cislo & \meno & \vaha & \vykon}%
\\hline
\end{tabular}
```

a zodpovedajúci výstup vidíme v tabuľke 3.

Číslo	Prezývka	Hmotnosť	Výkon
387.0	Mikádo	93 ton	1546 kW
498.1	Albatros	113 ton	1950 kW
556.0	Štokr	99 ton	1472 kW
464.2	Rosnička	112 ton	1250 kW

Tabuľka 3. Riadenie vzhľadu tabuľky pri použití príkazu `\csvreader`.

Číslo	Prezývka	Hmotnosť	Výkon
387.0	Mikádo	93 ton	1546 kW
498.1	Albatros	113 ton	1950 kW
556.0	Štokr	99 ton	1472 kW
464.2	Rosnička	112 ton	1250 kW

Tabuľka 4. Zmena vzhľadu tabuľky pri použití príkazu `\csvreader`.

Neprijemnosť s názvami makier je možné obísť úpravou parametrov príkazu `\csvreader`. Ak chceme potlačiť automatické pomenovanie makier a použiť svoje vlastné mená, použijeme príkaz v takejto podobe:

```
\csvreader[late after line=\\hline]{loko.csv}{Číslo=\cislo,
Prezývka=\meno,Hmotnosť=\vaha,Výkon=\vykon}
```

Volba `[late after line=\\hline]` potom určuje, že jednotlivé riadky tabuľky budú oddelené horizontálnou čiarou. Výsledok je možné pozorovať v tabuľke 4

O niečo pohodlnejšie formátovanie je možné pomocou definovania vhodných voliteľných kľúčov. To umožňuje definovanie vytvorenie `pgfkeys` štýlu pre celú tabuľku. V našom prípade môžeme príkaz `\csvread` modifikovať napríklad takto:

```
\csvreader[tabular=|l|r|l|,
table head=\hline \bfseries Lokomotíva&\bfseries Hmotnosť&
\bfseries Výkon\\hline\hline,
table foot =\hline]%
{loko.csv}{Číslo=\cislo,Prezývka=\meno,Hmotnosť=\vaha,
Výkon=\vykon}%
{\cislo~\meno & \vaha & \vykon}%
```

Výsledok nášho úsilia potom vidíme v tabuľke 5.

Lokomotíva	Hmotnosť	Výkon
387.0 Mikádo	93 ton	1546 kW
498.1 Albatros	113 ton	1950 kW
556.0 Štokr	99 ton	1472 kW
464.2 Rosnička	112 ton	1250 kW

Tabuľka 5. Zmena vzhľadu tabuľky pri použití voliteľných parametrov príkazu `\csvreader`.

Číslo	Prezývka	Hmotnosť	Výkon
387.0	Mikádo	93 ton	1546 kW
498.1	Albatros	113 ton	1950 kW
556.0	Štokr	99 ton	1472 kW
464.2	Rosnička	112 ton	1250 kW

Tabuľka 6. Tabuľka so vzhľadom riadeným pomocou štýlu definovaného prostredníctvom `\csvstyle`.

2. Pokročilejšie techniky

2.1. Vlastný štýl tabuľky

Nasledujúca ukážka ilustruje, ako si pomocou makra `\csvstyle` môžeme definovať štýl tabuľky. Tu opäť použijeme automatizované pomenovanie stĺpcov tabuľky s využitím voľby `head to column names`. Pri tomto prístupe však položky záhlavia nesmú obsahovať medzery, ani špeciálne znaky. Preto aj v ukážke používame opäť súbor `loko1.csv` s odstránenou diakritikou. Nasledujúci zdrojový kód ilustruje definíciu štýlu tabuľky:

```
\csvstyle{mojaTabulka}{tabular=|c|l|r|l|,
table head=\hline \bfseries Číslo& \bfseries Prezývka &
\bfseries Hmotnosť& \bfseries Výkon\\\hline\hline,
table foot=\hline,
head to column names}
```

Samotnú tabuľku potom vygenerujeme jednoducho pomocou príkazu `\csvread`, tak ako to vidíme v nasledujúcej ukážke:

```
\csvreader[mojaTabulka]{loko1.csv}{}%
{\cislo & \meno & \vaha & \vykon}
```

Výsledkom je potom tabuľka 6.

Syntax `pgfkeys` umožňuje vytvárať vlastné štýly pre opakované použitie a doiahnutie konzistentného a jednotného vzhľadu. Nasledujúci príklad je upravený tak, aby sa získalo viac alebo menej možností nastavení. Nový štýl, pomenovaný `mojeVlaky`, definujeme pomocou nasledujúcej postupnosti príkazov:

```
\csvset{mojeVlaky/.style={%
tabular=|r|l|r|,
table head=\hline \bfseries Číslo&\bfseries Lokomotíva &
\bfseries #1\\\hline\hline,
late after line=\\\hline,
column names={cislo=\cislo,meno=\meno}
}}
```

Takto definovaný štýl potom môžeme použiť na sadzbu dvoch tabuliek 7 a 8 pomocou príkazov

```
\csvset{mojeVlaky/.style={%
tabular=|r|l|r|,
table head=\hline \bfseries Číslo&\bfseries Lokomotíva &
\bfseries #1\\\hline\hline,
late after line=\\\hline,
column names={cislo=\cislo,meno=\meno}
}}
```

resp.

```
\csvreader[mojeVlaky={Váha}]{loko1.csv}{vaha=\vaha}%
{\cislo & \meno & \vaha}%
```

```
\hfill%
\csvreader[mojeVlaky={Výkon}]{loko1.csv}{vykon=\vykon}%
{\thecsvrow & \cislo~\meno & \vykon}%
```

Číslo	Lokomotíva	Váha
387.0	Mikádo	93 ton
498.1	Albatros	113 ton
556.0	Štokr	99 ton
464.2	Rosnička	112 ton

Tabuľka 7. Tabuľka so vzhľadom definovaným štýlom `mojeVlaky`.

Číslo	Lokomotíva	Výkon
1	387.0 Mikádo	1546 kW
2	498.1 Albatros	1950 kW
3	556.0 Štokr	1472 kW
4	464.2 Rosnička	1250 kW

Tabuľka 8. Pozmenená tabuľka so vzhľadom definovaným štýlom `mojeVlaky`.

3. Záver

V článku boli ilustrované možnosti automatickej tvorby tabuliek z dátových súborov uložených vo formáte `csv`. Základné postupy boli ilustrované na niekoľkých jednoduchých príkladoch tabuliek.

Poďakovanie. Tento príspevok vznikol s príspevom grantu KEGA-041ŽU-4/2017 podporovaného Slovenskou kultúrno-edukačnou grantovou agentúrou.

Literatúra

- [1] BLAŠKO, R.: *L^AT_EX nie je farba na maľovanie*, Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2010, Žilina, 1.–4. júla 2010, str. 43–52, ISBN 978-80-970457-0-8, <http://sospreskoly.org/files/OSSConf2010/ossconf2010-Blasko.pdf>.
- [2] BLAŠKO, R.: *L^AT_EX nie je farba na maľovanie, ale na písanie*, Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2010, Žilina, 1.–4. júla 2011, str. 43–52, ISBN 978-80-970457-0-8, http://ossconf.soit.sk/static/contributions2011/Blasko_farba.pdf.
- [3] KOPKA, H. – DALY, P. W.: *L^AT_EX – Podrobný príručce*, Brno, Computer Press, 2004, ISBN 80-722-6973-9.
- [4] RYBIČKA, J.: *L^AT_EX pro začátečníky*, Brno, KONVOJ 2003, ISBN 80-7302-049-1.
- [5] SCHRÖTTER, J. – FULTNER, B.: *Svět lokomotiv*, Brno, CPress, 2016, ISBN 978-80-264-1268-7.

Kontaktná adresa

RNDr. Aleš Kozubík, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská Republika,
E-mailová adresa: alesko@frcatel.fri.uniza.sk

VÝSLEDKY VÝUKY ZPRACOVÁNÍ TEXTŮ

JIŘÍ RYBIČKA (CZ)

Abstrakt. Výuka zpracování textů je na Provozně ekonomické fakultě MENDELU zavedena formou volitelného předmětu již více než 18 let. Koncepce předmětu se postupem času poněkud změnila z počátečního více technického pojetí k současnému, které se soustřeďuje na obecnější poznatky z typografie a zaměřením na odborné, zejména závěrečné práce. Článek se zabývá nástinem analýzy výsledků výuky v tomto předmětu, a to zpracováním vybraných výsledků ze zkouškových písemných prací.

Klíčová slova. zpracování textů, výuka, analýza výsledků.

THE RESULTS OF TEACHING TEXT PROCESSING

Abstract. Teaching of text processing is introduced at the Faculty of Business and Economics of Mendel University in Brno as an optional subject for more than 18 years. Concept of the subject has over time somewhat changed from the initial more technical concept to the current one focused on more general knowledge of typography and on technical texts, especially the final works. The article deals with the outline of the analysis of the learning outcomes in this subject, by processing selected exam results work.

Keywords. text processing, teaching process, results analysis.

Úvod

Aplikace počítačů v oblasti zpracování textů vykazují již od počátků v 80. a 90. letech minulého století masivní zastoupení napříč všemi obory. Je to pouze projevem skutečnosti, že jde o velmi potřebnou a prakticky použitelnou úlohu a ta je navíc podpořena i rozvojem technického vybavení, které umožnilo téměř ze dne na den zcela vyměnit technologii (psací stroje, horká sazba) za počítačové tiskárny. Tento prudký vývoj však nebyl doprovázen dostatečně masivní osvětou v oblasti dříve nedostupných technologií, jakými bezesporu byla sazba z liter odlévaných z kovu, a to ručním i strojním způsobem. Vznikla doslova propast mezi možnostmi dostupných počítačových programů a technických prvků a mezi znalostmi, které by umožňovaly jejich správné využití.

Tato mezera existuje dodnes a její zmenšování představuje dlouhodobý a náročný proces reprezentovaný existencí mnoha přístupů ve výuce na školách všech stupňů. Je přirozené, že snahou předmětů s touto tematikou je co nejefektivnějším způsobem vybavit studenty základními poznatky. Zůstává ovšem otázkou, do jaké míry se to daří.

1. Současný stav

1.1. Stručný přehled nabízených výukových aktivit

Výuce základů úpravy textu a typografie se věnuje na školách poměrně široká pozornost, i když stále zjevně chybí jakákoliv koordinace a ustálená náplň. Jen pro ilustraci můžeme uvést několik typů takových kurzů:

- Informace o vybraných pravidlech vhodných podle autora pro zpracování závěrečné práce. Texty tohoto typu jsou časté na vysokých školách (např. Raclavský, 2018).
- Učební texty zaměřené na seznámení s normou ČSN 01 6910 (2014), např. text určený pro střední školu Typografie – základy (2018) nebo text ze základní školy (Šošolík, 2018).
- Komplexněji pojaté texty zabývající se aplikací typografických pravidel v určité oblasti (oboru), například bakalářská práce Cvingrářové (2011) o využití počítačů v chemii.
- Texty o typografii doplněné o návody, jak prezentovaná pravidla realizovat (nejčastěji v MS Word) – například Typografická pravidla (2018).

Nepříjemnou vlastností některých těchto textů je nízká kvalita. Například již zmíněná práce Cvingrářové (2011) nerespektuje typografická pravidla, i když o nich přímo pojednává.

1.2. Výuka zpracování textů na Provozně ekonomické fakultě MENDELU

Předmět „Zpracování textů na počítači“ se vyučuje již od akademického roku 1999/2000. Do prvního běhu bylo zapsáno 69 studentů. V posledních letech se předmět vyučuje v zimním i letním semestru a za akademický rok jej odstuduje přes 200 studentů.

Původní náplň byla zaměřena zejména na představení technologie počítačové sazby systémem \TeX (přesněji \LaTeX) a částečně také Adobe InDesign, zatímco typografická pravidla tvořila spíše doplněk a nezbytný základ pro správné formátování dokumentů. Postupem času ovšem toto technické hledisko poněkud ustoupilo a stále větší důraz byl kladen na specifika odborných prací, a to zejména závěrečných prací. To se ukázalo jako poměrně účinný prostředek pro ztraktivnění předmětu i pro studenty, pro něž technická stránka věci neznamovala centrální bod zájmu, ale spíše se chtěli zorientovat při zpracovávání bakalářských a diplomových prací.

Konstelace učebních plánů a možnosti zapisovat volitelné předměty dokonce velmi zvýhodňuje studenty neinformatických oborů, což se odráží v minimálním počtu studentů informatiky, kteří si předmět zapíší. Pro většinu studentů je tedy technologie sazby a přehled možností sazbecích systémů nezajímavý. To bylo také příčinou postupné změny ve vedení předmětu. Cvičením stále zůstává technický

charakter a řeší se na nich zpracování dokumentů v systému (Xe)LaTeX, seznámení s principy práce v systému InDesign bylo zcela opuštěno, a přednášky pak obsahují typografická pravidla a jejich realizaci v programech typu MS Word nebo v jeho open source alternativách.

Rovněž závěrečné ověřování znalostí obsahuje z větší části typografické zásady, technickou stránku zpracování dokumentů představuje jen zápočtová úloha, na níž mají studenti prokázat základní orientaci v nejběžnějších rekvizitách systému (Xe)LaTeX.

1.3. Pohled fakulty

Tento stav je víceméně v souladu s potřebami fakulty. Z úrovně závěrečných prací je patrný posun oproti dřívějšímu stavu. Za „normální“ se nyní považuje práce, jejíž typografické a technické zpracování respektuje většinu běžných typografických pravidel, zatímco práce formátované laicky a napěchované nejružnějšími chybami a technickými nedostatky jsou již skutečnou výjimkou. Přispívá k tomu i existence šablony s předdefinovanými styly, šablona ovšem zdaleka nevyřeší všechno a bez aktivního přístupu studentů by kvalitní dokumenty nevznikaly.

Kvalita závěrečných prací je sledována jako jedna z priorit fakulty. Určitě se v této oblasti dá ještě mnoho věcí zlepšovat. V závěrečných pracích se i dnes vyskytují formální chyby, a to i u studentů, kteří předmět Zpracování textů na počítači absolvovali. Vzniká tedy logická otázka, nakolik jsou typografická pravidla studenty přijímána a osvojována. Prvotním záměrem výzkumu tedy bylo zjistit, jaká je úspěšnost studentů, a to nikoliv sumárně v celé škále poznatků předmětu, ale u zcela specifických a dílčích pravidel.

2. Materiál a metody

Sledované prvky byly rozděleny do skupin s podobným charakterem. Například mezerování je skupina jevů, kde se sleduje správné uvádění mezer u interpunkce, mezi číslem a jednotkou atd. U zkouškové práce byly sledovány pouze úlohy, v nichž jako zadání byl uveden text se záměrnými chybami a úkolem studenta bylo tyto chyby nalézt a vyznačit.

Pro analýzu byly v první fázi určeny pouze tři skupiny prvků. První skupinou byly jevy týkající se uvádění pomlčky a spojovníku. Ve druhé skupině bylo sledováno mezerování a poslední skupina obsahuje pravidla jazyková. Chyby náležející do poslední skupiny nejsou obsaženy ve výukovém obsahu předmětu, jde například o záměny písmen nebo nesprávně zapsané slovo (např. „úkol“). Proto má tato skupina charakter kontrolní – předpokládáme, že chyby tohoto typu jsou studenti bez jakéhokoliv zaváhání rozpoznat, otázkou pouze je, jak jsou pozorní.

Ze zkouškových prací byly vybrány dvě úlohy s podobně dlouhým zadáním s přibližně stejným počtem chyb a byly zaznamenány všechny jednotlivé případy odhalení či neodhalení příslušné chyby.

Základní charakteristikou, která byla sledována, byla úspěšnost odhalení jednotlivých chyb. Každá skupina měla 3 výskyty chyby, výjimkou byla skupina pomlček u druhé úlohy, kde byly pouze dva výskyty. Pro zjištění, zda úspěšnost odhalení chyby jedné skupiny nějak souvisí s odhalením chyby jiné skupiny byly vypočteny korelační koeficienty porovnávající průměrné úspěšnosti skupin.

3. Výsledky

Bylo analyzováno celkem $N = 139$ prací. Úspěšnosti jednotlivých skupin uvádí Tab. 1. Korelační koeficienty vypočtené mezi vektory úspěšností vždy dvou skupin uvádí Tab. 2.

Tabulka 1. Úspěšnosti odhalení chyb v jednotlivých skupinách

Skupina	Úspěšnost [%]
Pomlčky	50,5
Mezerování	69,8
Jazyk	75,1

Tabulka 2. Korelace mezi úspěšnostmi skupin

Dvojice	Korelační koeficient
Pomlčky–Mezerování	0,274
Pomlčky–Jazyk	0,205
Mezerování–Jazyk	0,298

4. Diskuse

Jak je patrné z výsledků, pomlčky a jejich správný zápis dělá studentům největší potíže z uvedených tří sledovaných skupin chyb. Lze konstatovat, že polovina chybně zapsaných případů zůstala neodhalena a koresponduje to také s četností chyb v pomlčkách v závěrečných pracích. Na podporu tohoto tvrzení sice neexistuje bohužel žádný dostatečně rozsáhlý výzkum, ale odpovídá to zkušenostem nejen autora, ale i kolegů sledujících kvalitu závěrečných prací u státních zkoušek.

O něco lépe dopadla skupina chyb sledujících správné mezerování. S téměř 70% úspěšností jsou studenti schopni tyto chyby detekovat. V kontrolní skupině jazykových chyb dosáhla úspěšnost tří čtvrtin. Z toho je tedy patrné, že znalost chyb v mezerování je studenty již osvojena velmi dobře.

Korelace mezi jednotlivými skupinami neprokázaly žádnou závislost. Lze tedy úspěšnost odhalení chyby jedné skupiny považovat za nezávislou na odhalení

chyby ve skupině jiné. Patrně tedy neplatí, že někteří studenti jsou pečlivější a odhalí chybu bez závislosti na druhu, zatímco jiným jakékoliv chyby unikají.

Můžeme si nyní položit otázku, proč chyby v psaní pomlček vykazují v četnosti znatelný rozdíl oproti chybám v mezerování. Jedním z faktorů může být, že způsob psaní pomlček (spojovníků) je pro studenty relativně novým prvkem, který je spojen až s počítačovou technologií sazby, zatímco mezerování je v základních obrysech v podstatě beze změny převzato už z technologie psacích strojů. Uvážíme-li, že studenti jsou masově ovlivňováni texty pořizovanými laickou komunitou (do níž bohužel musíme započítat i značnou část učitelů nižších stupňů škol), je jejich citlivost na tyto chyby značně otupělá. Oproti tomu i laický uživatel textového procesoru nebo učitel na základní škole tuší, že například za interpunkcí se mezeza píše, zatímco před ní ne, většinou tedy pozitivně ovlivňuje čtenáře správným zápisem, i když širší povědomí o typografických pravidlech nemá.

Druhou podstatnou otázkou je, zda kvalita výuky v předmětu Zpracování textů na počítači je dostatečná na to, aby studenti byli na závěr schopni reagovat na chyby tohoto druhu. Je samozřejmě potřebné vzít v úvahu i skutečnost, že předmět má daleko širší záběr, může se tedy snadno stát, že některé prvky zaniknou mezi ostatními. Autor se však spíše příkládní k názoru, že padesátiprocentní úspěšnost odhalení chyby v zápisu pomlčky je malá a o nadstandardní kvalitě výuky či přípravy studentů to nesvědčí.

Tyto odpovědi by si samozřejmě zasloužily potvrzení či vyvrácení podrobnějším výzkumem. Bylo by rovněž vhodné porovnat výstupní znalosti s počátečním stavem, a zjistit tak přidanou hodnotu ve studiu daného předmětu. Jistě by také byla zajímavá i analýza časové řady – za 18 let výuky je k dispozici poměrně značné množství dat.

5. Závěr

Výsledky získané z prvotní analýzy mohou být interpretovány jako stimul pro zvětšení úsilí ve výuce zpracování textů. Podobnou analýzu by bylo vhodné provést pro více skupin sledovaných jevů – východiskem by přitom mohla být i provedená analýza formální kvality závěrečných prací odevzdávaných na fakultě. Pozornost by si rovněž zasloužil výzkum časové řady, pro nějž existuje v této chvíli poměrně značné množství podkladových dat.

Všechny uvedené možnosti by měly vyústit ve zvýšení kvality výuky v předmětu Zpracování textů počítačem, a tím i ve zvýšení kvality závěrečných prací.

Reference

- [1] CVINGRÁFOVÁ, E. Počítače v chemii. Příloha bakalářské práce. [online] [vid. 12. 5. 2018] Dostupné na https://profesornovotny.webnode.cz/_files/20000061-50a70530ea/Z%C3%A1klady%20typografie.pdf

- [2] ČSN 01 6910. Úprava dokumentů zpracovaných textovými procesory. Praha: Ústav pro normalizaci a měření, 2014.
- [3] RACLAVSKÝ, J. *Nejdůležitější pravidla úpravy odborného textu*. [online] [vid. 15. 5. 2018] Dostupné na http://www.phil.muni.cz/fil/vyuka/uprava_odb_textu.html
- [4] ŠOŠOLÍK, I. *Typografická pravidla*. Prezentace ve formátu PPS [online] [vid. 15. 5. 2018] Dostupné na www.zsjablunka.cz/html/vyuka/informat/text/typograf.pps
- [5] *Typografická pravidla*. Dokument formátu DOCX, 10. 1. 2014. [online] [vid. 22. 5. 2018] Dostupné na www.spskarvina.cz/www/images/stories/files/2011/skola/2011/typ_pravidla.docx
- [6] *Typografie – základy*. [online] [vid. 15. 5. 2018] Dostupné na http://www.jardaz.cz/articles.php?article_id=66

Kontaktní adresa

doc. Ing. Jiří Rybička, Dr., Ústav informatiky, Provozně ekonomická fakulta, Mendelova univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika,
E-mailová adresa: rybicka@mendelu.cz, <http://akela.mendelu.cz/~rybicka>

POUŽITIE RÔZNYCH PÍSIEM V L^AT_EX-U I DYNAMICKÁ, PRECHODOVÁ A STATICKÁ ANTIKVA

RUDOLF BLAŠKO (SK)

Abstrakt. T_EX, resp. L^AT_EX disponujú veľmi pekným a hlavne veľmi kvalitným písmom Computer Modern (CS), ktoré vytvoril sám autor Donald Knuth. Toto písmo a jeho novšie rozšírenie Latin Modern (LM) s kvalitnejšími metrikami sú stále na vrchole počítačovej sadzby. Čo sa týka odbornej sadzby (s výrazmi matematickými, fyzikálnymi, chemickými atď.) sú nedostihnuteľné. Aj krásna literatúra vysádzaná týmito písmami vyzerá esteticky a príjemne sa číta (tým nemyslíme obsah ale kresbu litier). Pri sadzbe v L^AT_EX-u môžeme bez problémov použiť viac ako 450 fontov písiem a nemusíme použiť X_YL^AT_EX. Niektoré z nich sú kompletne typy písiem (napr. Libre Bodoni) s metrikami a ak ich nemáme, musíme ich inštalovať ako fonty L^AT_EX-u. Väčšina z nich je riešená pomocou balíčkov, nie všetky sú k dispozícii vo všetkých potrebných fontoch, ale na to aby iniciatívny pisateľ dokázal vytvoriť neesteticky a nečitateľný dokument s množstvom písiem to postačí. Na webe existuje „The L^AT_EX Font Catalogue“, kde si môžeme vhodné písmo vybrať podľa jeho zobrazenia. Pri každom písme je uvedené ako písmo použiť, adresa na CTAN, kde si ho môžeme stiahnuť a taktiež dátum poslednej aktualizácie. Drvivá väčšina týchto písiem je k dispozícii pri štandardnej inštalácii T_EXLive, takže ak nepotrebuje nejaké exotické písmo, stačí vložiť doporučený príkaz do preambuly a písať.

Kľúčové slová. Písmo, Computer modern, L^AT_EX, počítačová sadzba, antikva.

USE DIFFERENT FONTS IN L^AT_EX I DYNAMIC, TRANSITIONAL AND STATIC ANTIQUA

Abstract. T_EX, respectively, L^AT_EX includes a very nice and especially high-quality Computer Modern (CS) font, created by the author Donald Knuth himself. This font and its more recent Latin Modern (LM) extension with better metrics are still at the top of the desktop publishing quality rate. As far as professional rates (with mathematical, physical, chemical, etc.) are concerned, they are unmatched. As well the fiction typeset in these fonts also looks aesthetically pleasing and readable (not the content but the drawing of the letters). With L^AT_EX typesetting, we can use more than 450 fonts without problems, and we don't have to use X_YL^AT_EX. Some of them are complete font types (e.g. Libre Bodoni) with metrics and if we don't have them, we need to install them as L^AT_EX fonts. Most of them are distributed as packages, but not all of them are available in all necessary fonts. However, it is good enough for an initiative writer to create an unattractive and unreadable document with a multitude of fonts. In the web environment, there is “The L^AT_EX Font Catalog”, where we can choose a suitable font according to its display. Each font is listed with the instruction on how to use the given font, its CTAN address, where we can download it, as well as the date of the last update. The vast majority of these fonts are available with the standard T_EXLive installation, so if we don't need any exotic font, it is sufficient just to put the recommended command in the preamble and type.

Keywords. Typeface, Computer modern, L^AT_EX, Computer Typesetting, Antiqua.

Úvod

Klasifikácia tlačových písiem predstavuje ich členenie na základe charakteristických znakov. Od začiatku 20. storočia vzniklo mnoho členení, nie všetky sú vyhovujúce, nepoužívajú sa jednotne a ani dôsledne, čo komplikuje život grafikom a typografom. Dnes je k dispozícii niekoľko tisíc rôznych súborov písiem od rôznych výrobcov. Je pravda, že značná časť z nich je nekvalitná a kvalitné písma vo väčšine nie sú voľne prístupné. Znalosť základnej klasifikácie písma umožňuje typografovi¹ vybrať vhodné písma a vyvarovať sa chýb pri ich kombinovaní.

Medzinárodný typografický zväz (Association Typographique International) navrhol jednotné medzinárodné delenie písiem. Odlišnosti, ktoré existujú v jednotlivých štátoch, umožňujú túto kvalifikáciu dopĺňať a obmieňať. Určenie charakteru písma rešpektuje chronologický vývoj s dôrazom na prevládajúce grafické znaky písma. Základné charakteristické znaky písma sú **dynamický** (rukopisný) a **statický** (kresbový) **princíp**. Medzi nimi sa nachádza **prechodová forma** a tieto tri skupiny dopĺňa **štvrtá skupina**, kde prevládajú zmiešané prvky.

Dynamický princíp vychádza z písiem písaných plochým kaligrafickým perom, ktoré vytvára menej výrazné tieňovanie. Os oblých ťahov písma je šikmá k účiariu. Tieto písma majú diferencovanú šírkovú proporciu. **Statický princíp** vychádza z výrazného diferencovania čiar na tieňovacie a vlasové čiary. Os oblých ťahov písma je kolmá na úciarie. Šírková proporcia písmových znakov nie je diferencovaná. Okrem logických výnimiek (i, j, l, r, t, m, w) sú písmové znaky kreslené do rovnako opticky široko pôsobiacej plochy.

Antikva je základná forma latinkového písma, je odvodená z rímskeho nadpisového písma (verzálky) a z neskorších karolínskych mínusiek. Prvá tlačová antikva vznikla v polovici 15. storočia. Jej základným znakom sú rozdiely v hrúbke hlavných a vedľajších ťahov a ukončenie ťahov serfmi.

Pre základnú orientáciu je vhodná klasifikácia založená na prítomnosti/nepřítomnosti pätiiek, na ich tvare, tieňovaní a sklone písma. Táto klasifikácia zároveň rešpektuje historické vplyvy, pretože práve tieto znaky sa v priebehu času najvýraznejšie menili. Jej základom je rozdelenie, ktoré navrhol významný český typograf a tvorca písma *Jan Solpera* (uverejnené v časopise *Typografia* 9/1977). Všetky písma sú rozdelené do štyroch tried:

- Latinkové písma,
- nelatinkové písma pravosmerné (grécke, slovanské, orientálne, klingonské...),
- nelatinkové písma ľavosmerné (hebrejské, perské, arabské...),
- ostatné písma – exotické (čínske, japonské, korejské, barmské...).

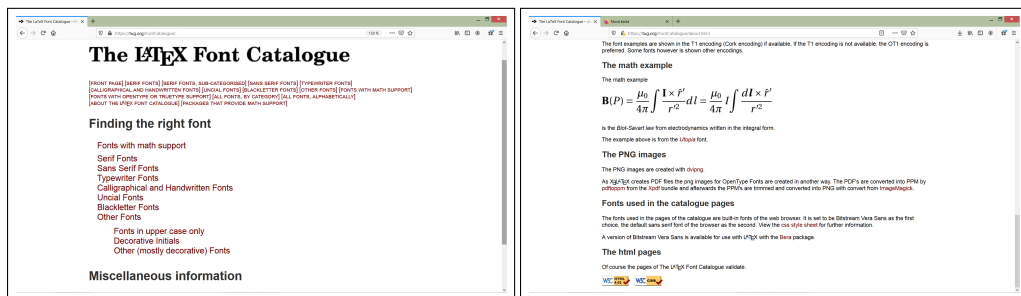
¹V mnohých prípadoch je to autor publikácie, ktorý vyberá a kombinuje rôzne písma iba na základe vlastných pocitov – a samozrejme nevhodne.

1. The L^AT_EX Font Catalogue

T_EX, resp. L^AT_EX disponujú veľmi pekným a hlavne veľmi kvalitným písmom Computer Modern (CS), ktoré vytvoril sám autor Donald Knuth. Toto písmo a jeho novšie rozšírenie Latin Modern (LM) s kvalitnejšími metrikami sú stále na vrchole počítačovej sadzby. Čo sa týka odbornej sadzby (s výrazmi matematickými, fyzikálnymi, chemickými atď.) sú nedostupné. Ale pri tvorbe dokumentov v L^AT_EX-u môžeme pomocou X_YL^AT_EX-u použiť praktický každý font, ktorý máme v počítači. V súčasnosti môžeme bez problémov použiť viac ako 450 fontov písiem a nemusíme použiť X_YL^AT_EX. Niektoré z nich sú kompletne typy písiem (napr. Libre Bodoni) s metrikami. V prípade, že naša distribúcia T_EX-u takýmto písmom nedisponuje, musíme si ho nainštalovať ako fonty T_EX-u. Najvýhodnejšie je ich umiestniť do adresára, ktorý T_EX prehliada, ale sa neprepisuje novými inštaláciami. V T_EXLive (najčastejšia distribúcia T_EX-u pod Linuxom aj pod komerčným windowsom) je to adresár `/usr/local/texlive/texmf-local` pod linuxom, resp. `\texlive\texmf-local` pod win. Ukážky zdrojových v príspevku sú optimalizované pre L^AT_EX.

Ako sme už spomínali, k dispozícii máme pomerne veľké množstvo fontov, ktoré môžeme použiť v L^AT_EX-u. Niektoré sú pre bežné použitie nepoužiteľné, ale v akcidenčnej tlači sa človeku s estetickým cítením medze nekladú. Dobrým pomocníkom pri orientácii sa v tomto množstve fontov je „The L^AT_EX Font Catalogue“ umiestnený na adrese <https://tug.org/FontCatalogue/> (obr. 1). Uvedený katalóg je dobre a prehľadne usporiadaný. V hornej časti každej stránky katalógu je menu, pomocou ktorého si môžeme vyberať fonty podľa rôznych kritérií:

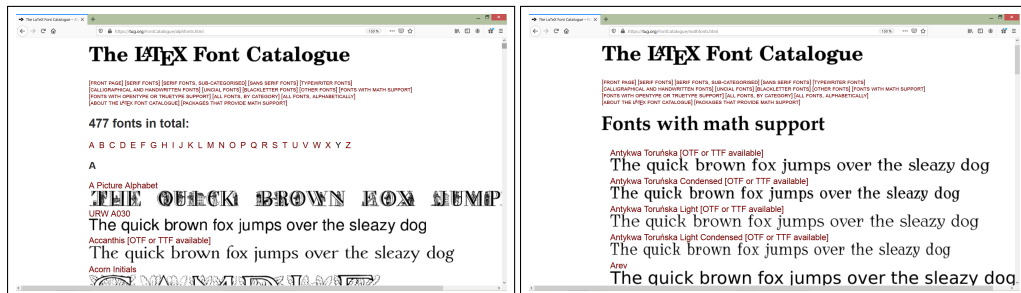
[Front Page] [Serif Fonts] [Sans Serif Fonts] [Typewriter Fonts]
 [Calligraphical and Handwritten Fonts] [Uncial Fonts]
 [Blackletter Fonts] [Other Fonts] [Fonts with Math Support]
 [All Fonts, by category] [All Fonts, alphabetically].



Obr. 1. The L^AT_EX Font Catalogue [<https://tug.org/FontCatalogue/>]

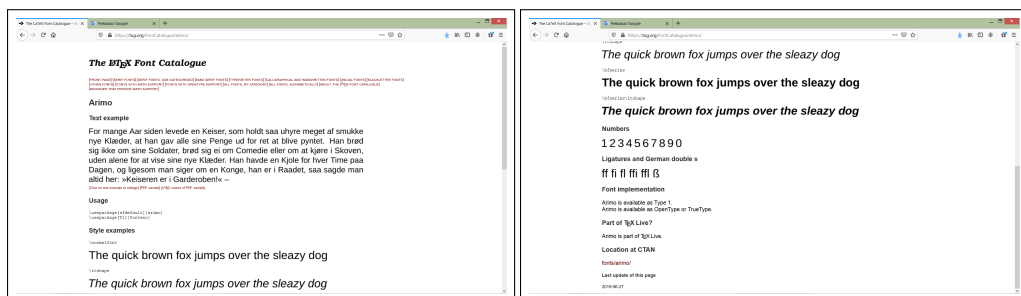
Taktiež sa tu nachádza mnoho užitočných informácií o katalógu: [About The L^AT_EX Font Catalogue] [Packages that provide math support].

Navyše na hlavnej stránke [Front Page] nájdeme odkaz na dokumentáciu Font documentation a špeciálne odkazy na fonty Fonts in upper case only, Decorative Initials, Other (mostly decorative) Fonts.



Obr. 2. Zoznam fontov podľa abecedy a zoznam fontov s matematickou podporou

Ak vyberieme nejakú skupinu fontov, zobrazí sa nám stránka so zoznamom použiteľných fontov, ktorý obsahuje názov fontu a grafickú ukážku (obr. 2). Po výbere konkrétneho fontu, získame informácie potrebné pre jeho použitie v systéme L^AT_EX. Ako ukážku uvádzame font arimo (obr. 3).



Obr. 3. Font arimo [<https://tug.org/FontCatalogue/arimo/>]

2. Klasifikácia latinkových tlačových písiem

Klasifikácia latinkových písiem je určená štvormiestnym kódom $x_1.x_2.x_2.x_3$. Napríklad 1.2.0.0 predstavuje Dynamickú rezezančnú antikvu.

- Prvé číslo x_1 určuje klasifikačnú skupinu: 1. Dynamická antikva. 2. Prechodová antikva. 3. Statická antikva. 4. Lineárne písmo serifové. 5. Lineárne bezserifové statické písmo. 6. Lineárne bezserifové konštruované písmo. 7. Lineárne bezserifové dynamické písmo. 8. Lineárne antikva. 9. Kaligrafické písma. 10. Voľne písané písma. 11. Písma lomené.

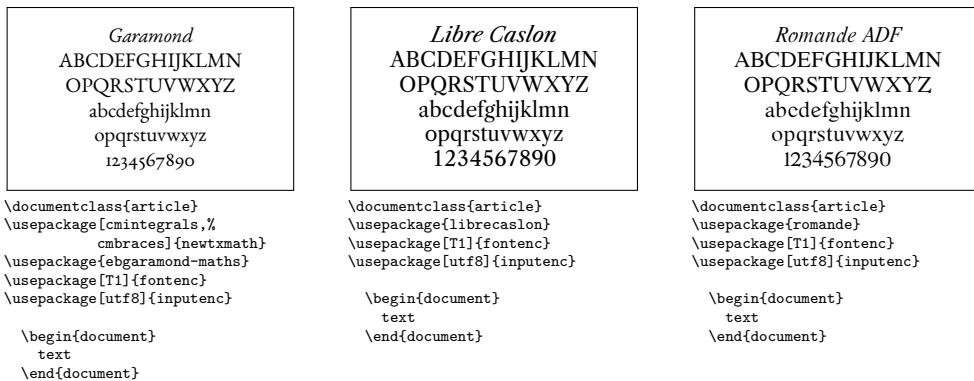
- Druhé číslo x_2 označuje základnú verziu.

- Tretie číslo x_3 klasifikuje vyznačovaciú naklonenú verziu. Rozoznávajú sa varianty dynamický, prechodový a statický.
- Štvrté číslo x_4 s rozsahom 1 až 6 označuje zdobené verzie.

1.0.0.0 Dynamická antikva

V tejto skupine sú renezančné antikvy vytvorené od druhej polovice 15. storočia do polovice 18. storočia a ich novodobé modifikácie. Používajú sa hlavne pre sadzbu krásnej literatúry, ale sú použiteľné univerzálne. Sú elegantné a ich kresba má dynamický (kaligrafický) charakter. Vychádzajú z písiem ručne písaných plochým kaligrafickým perom. Kresbu môžeme charakterizovať nasledovne (obr. 4):

- malé rozdiely v hrúbkach hlavných a spojovacích ťahov,
- os tieňovania oblých písmových ťahov je šikmá k úciariu (natočená doľava),
- pätky majú prehnutý alebo klinovitý nábeh k drieku,
- jednostranné horné serify mínusiek sú vždy šikmé,
- šírková proporcia je diferencovaná.



Obr. 4. Príklady písiem dynamickej antikvy vygenerovaných v L^AT_EX-u

- **1.1.0.0 Dynamická antikva benátská.** Priečka v písmene **e** je šikmá, napr. Roman, Jenson,² Centaur, Menhartova antikva.³

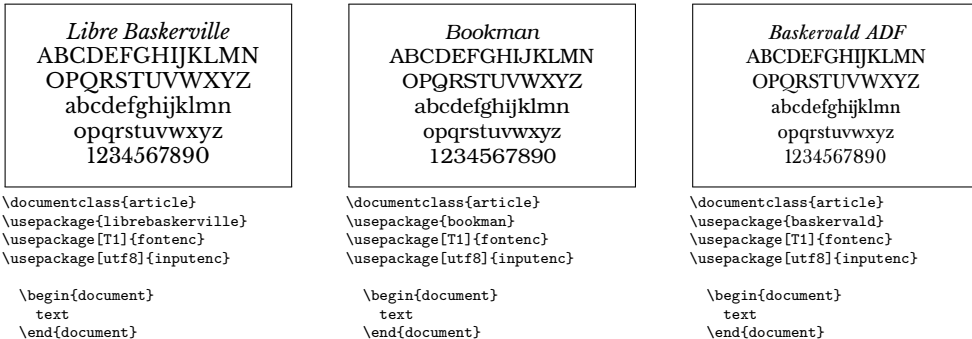
- **1.2.0.0 Dynamická antikva renezančná.** Rozdiely v hrúbkach hlavných a vedľajších ťahov zvýrazňujú písmo. Jednostranné horné serify malých písmen

²Nicolas Jenson (1420–1480) bol tlačiar a rytec, ktorý v Benátkach vytvoril najdokonalejší antikvové písmo ranného obdobia knihtače.

³Oldřich Menhart (1897–1962) bol český typograf, tvorca písma, kaligraf, knižný grafik.

sú vždy šikmé a majú výrazný nábeh, napr. Garamond,⁴ ITC Garamond, Bembo, Goudy,⁵ Palatino,⁶ ITC Galliard, Figural.

• **1.3.0.0 Dynamická antikva neskororenezančná.** Vyznačuje sa zväčšenou strednou výškou písma a v niektorých prípadoch výraznejšími rozdielmi v tieňovaní priamych ťahov. Písma tejto skupiny sú výborne čitateľné a v sadzbe dokumentov majú univerzálne použitie, napr. Times, Tempora, Caslon, Plantin.



Obr. 5. Príklady písiem prechodovej antikvy vygenerovaných v L^AT_EX-u

2.0.0.0 Prechodová antikva

Obsahuje barokové antikvy z druhej polovice 18. storočia a ich novodobé modifikácie. Písma tvoria prechod medzi dynamickými antikvami (ovplyvnenými ručne písanými formami) a statickými antikvami (konštruované písma). Majú univerzálne uplatnenie, najčastejšie sa používajú ako knižné písma. Patria sem napr. Baskerville,⁷ ITC New Baskerville, Bookman, ITC Century, Fournier,⁸ Menhartova romana, Týfova antikva,⁹ Kolektiv. Kresbu môžeme charakterizovať nasledujúcimi prvkami (obr. 5):

- výraznejší kontrast v hrúbkach hlavných a spojovacích ťahov,
- os tieňovania oblých písmových ťahov je takmer zvislá,
- serify sú jemné s miernym oblúkovým nábehom,
- jednostranné horné serify malých písmen nie sú také šikmé ako pri renezančnej dynamickej antikve, pri niektorých modifikáciach sú kolmé k ťahu písma,
- šírková proporcia je diferencovaná, ale smeruje k väčšej jednotnosti.

⁴Claude Garamond (1480–1561) bol francúzsky typograf, rezač písma a písmolejec.

⁵Frederic William Goudy (1865–1947) bol americký tlačiar, umelec, typograf a dizajnér.

⁶Giambattista Palatino (1515–1575) bol taliansky renezančný kaligrafický majster.

Písmo Palatino navrhol v roku 1948 nemecký typograf Hermann Zapf.

⁷John Baskerville (1707–1775) bol anglický podnikateľ, ktorý sa preslávil ako tlačiar a typograf.

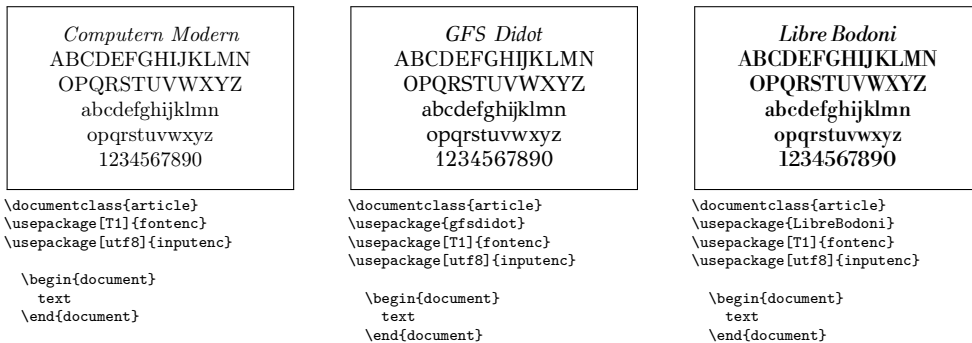
⁸Pierre Simon Fournier ml. (1712–1768) bol francúzsky kresliar a písmolejec.

⁹Josef Týfa (1913–2007) bol český typograf, knižný grafik a tvorca písma.

3.0.0.0 Statická antikva

Písma tejto skupiny uzavierajú historický vývoj foriem knižných písiem z konca 18. a začiatku 19. storočia. Obsahuje klasistické antikvy z tohto obdobia a ich novodobé modifikácie. Tvarovo nevychádzajú z charakteru písaného písma, sú to písma kreslené. Používajú sa hlavne ako knižné písma pre krásnu literatúru. Kresbu môžeme charakterizovať nasledujúcimi prvkami (obr. 6 a 7):

- veľké rozdiely v hrúbkach medzi jednotlivými hlavnými a vedľajšími ťahmi,
- os tieňovania oblých písmových ťahov je kolmá k účiariu (zvislá),
- pätky sú vlasové, vodorovné sú bez nábehu, zvislé majú zaoblený nábeh,
- šírková proporcia je jednotná.



Obr. 6. Príklady písiem statickej antikvy vygenerovaných v L^AT_EX-u

• **3.1.0.0 Statická antikva didotovská.** Vnútorný okraj kresby je kolmý k účiariu, vodorovné serify sú bez nábehov, zvislé serify majú geometricky konštruovaný zaoblený nábeh, napr. Bodoni,¹⁰ Didot,¹¹ ITC Fenice, Walbaum, ITC Zapf Book, Teimerova antikva,¹² Computer Modern,¹³ Latin Modern.¹⁴

• **3.2.0.0 Statická antikva anglická.** Tvarovo živšia konštrukcia znakov, výraznejšie serify s oblými nábehmi ku všetkým vedľajším ťahom, často sa používajú

¹⁰Giambattista Bodoni (1740–1813) bol taliansky typograf, tvorca písma, grafik, tlačiar.

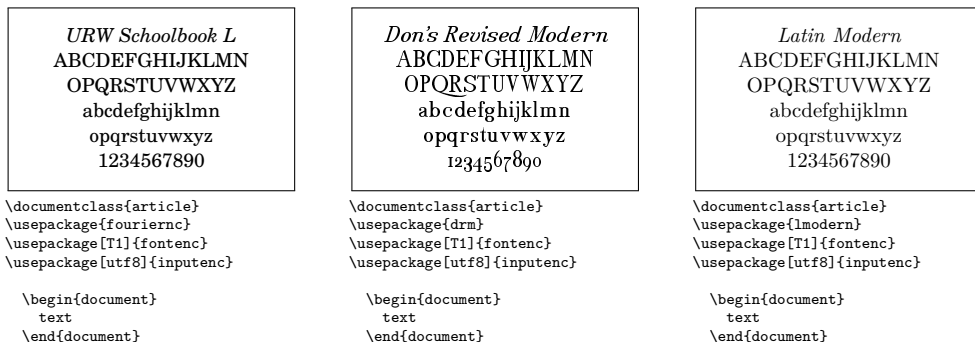
¹¹Didot je priezvisko francúzskych typografov, vydavateľov, spisovateľov a učencov 18. a 19. storočia. Francois-Ambroise Didot (1730–1804) zaviedol jednotku veľkosti písma (typografický bod 0,376 mm). Firmin Didot (1764–1836) vytvoril typografické písmo, ktoré je po ňom pomenované. Saint-Léger Didot (1767–1829) ako jeden z prvých zaviedol výrobu strojného (nekonečného) papiera.

¹²Pavel Teimer (1935–1970) bol český grafik a typograf.

¹³Písmo Computer Modern navrhnuté Donaldom Knuthom je základným písmom používaným v T_EX-ovských a L^AT_EX-ovských dokumentoch (implicitne nastaveným). Jeho rozšírením a vylepšením sú fonty Latin Modern.

¹⁴Písmo Computer Modern navrhnuté Donaldom Knuthom je základným písmom používaným v T_EX-ovských a L^AT_EX-ovských dokumentoch (implicitne nastavené). Jeho novším rozšírením s vylepšenými metrikami sú fonty Latin Modern.

na sádzanie titulkov, napr. Century Schoolbook, Modern, Monotype Modern, Normandia.



Obr. 7. Príklady písiev statickej antikvy vygenerovaných v L^AT_EX-u

Podakovanie. Tento príspevok vznikol s príspevím grantu KEGA-041ŽU-4/2017 podporovaného Slovenskou kultúrno-edukačnou grantovou agentúrou.

Literatúra

- [1] BRINGHURST, R.: The Elements of Typographic Style, Hartley Marks Publishers 2004, ISBN 0-88179-205-5.
- [2] PANÁK, J. – ČEPPAN, M. – DVONKA, V. – KARPINSKÝ, L. – KORDOŠ, P. – MIKULA, M. – JAKUCEWICZ, S.: *Polygrafické minimum*, Bratislava, Typoset, 2008, ISBN 978-80-970069-0-7.
- [3] ŠÍP, R.: *Typografické minimum*, Bratislava, SOU Polygrafické, Zväz polygrafie na Slovensku, 2000, ISBN 80-967598-5-X.
- [4] The L^AT_EX Font Catalogue, <https://tug.org/FontCatalogue/>.

Kontaktná adresa

RNDr. Rudolf Blaško, PhD., Katedra matematických metód a operačnej analýzy, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko,
E-mailová adresa: beerb@frcatel.fri.uniza.sk

POUŽITIE RÔZNYCH PÍSIEM V L^AT_EX-U II LINEÁRNE, KALIGRAFICKÉ, VOLNE PÍSANÉ A LOMENÉ PÍSMA

RUDOLF BLAŠKO (SK)

Abstrakt. Príspevok je pokračovaním článku „Použitie rôznych písiem v L^AT_EX-u I“ s podtitulom „Dynamická, prechodová a statická antikva“. Pokračuje v klasifikácii tlačových písiem ďalšími praktickými ukázkami v systéme L^AT_EX. Ako už bolo spomenuté pri sadzbe v L^AT_EX-u môžeme bez problémov použiť viac ako 450 fontov písiem a nemusíme použiť X_YL^AT_EX. Na webe existuje „The L^AT_EX Font Catalogue“, kde si môžeme vhodné písmo vybrať podľa jeho zobrazenia. Drvivá väčšina týchto písiem je k dispozícii pri štandardnej inštalácii T_EXLive, takže ak nepotrebujeme nejaké exotické písmo, stačí vložiť doporučený príkaz do preambuly a písať.

Kľúčové slová. Písmo, Computer modern, L^AT_EX, počítačová sadzba, antikva, lineárne písmo, volne písané písmo, kaligrafické písmo, lomené písmo.

USE DIFFERENT FONTS IN L^AT_EX II LINEAR, CALLIGRAPHIC, FREE HANDWRITING AND ANGLED FONTS

Abstract. The paper follows the article “Using various fonts in L^AT_EX I” with the subtitle “Dynamic, Transitional and Static Antiqua”. It continues to classify print fonts with other practical demonstrations in L^AT_EX. How it was mentioned in the previous paper, in the L^AT_EX typesetting, we can use more than 450 fonts without any problems, and we don’t have to use X_YL^AT_EX. In the web environment, there is “The L^AT_EX Font Catalog”, where we can choose a suitable font according to its display. The vast majority of these fonts are available with the standard T_EXLive installation, so if we don’t need any exotic font, it is sufficient just to put the recommended command in the preamble and type.

Keywords. Typeface, Computer modern, L^AT_EX, Computer Typesetting, Antiqua, Linear Font, Calligraphic Font, Free Handwriting Font, Angled Font.

Úvod

V príspevku „Použitie rôznych písiem v L^AT_EX-u I“ sme predstavili „The L^AT_EX Font Catalogue“ (<https://tug.org/FontCatalogue/>), kde nájdeme viac ako 450 fontov písiem použiteľných v systéme L^AT_EX. Predstavili sme klasifikáciu latickových písiem určenú štvormiestnym kódom $x_1.x_2.x_3$ a praktické použitie písiem patriacich do prvých troch skupín tejto klasifikácie: 1. Dynamická antikva. 2. Prechodová antikva. 3. Statická antikva. V tomto príspevku pokračujeme praktickými ukázkami použitia zostávajúcich ôsmich skupín tejto klasifikácie, pričom sa opäť budeme odvolávať na „The L^AT_EX Font Catalogue“.

4.0.0.0 Lineárne písmo serifové (egyptienka)

Patria sem písmo vychádzajúce z egyptienek od začiatku 19. storočia a ich novodobé modifikácie. Sú dobre čitateľné a spájajú prvky antikvy (serify, minimálne rozdiely v hrúbkach písmových ťahov) a lineárnych písom. Používajú sa hlavne v novinách, časopisoch, pri akcidenčnej tlači¹ a sadzbe titulkov. Sú nevhodné pre krásnu literatúru. Kresbu písom tejto skupiny môžeme charakterizovať nasledujúcimi prvkami (obr. 1):

- tieňovanie je nevýrazné alebo žiadne,
- výrazné serify (často hrubé ako ťahy písma) majú väčšinou štvorhranný, doskovitý alebo trojuholníkový tvar, sú bez alebo len s miernym nábehom,
- os oblých ťahov je vždy zvislá,
- šírková proporcia je statická (nemenná),
- zanedbateľné rozdiely v hrúbkach medzi jednotlivými ťahmi.



Obr. 1. Príklady lineárnych serifových písom vygenerovaných v L^AT_EX-u

¹Akcidenčné tlačoviny sú príležitostne tlačoviny pre osobnú, služobnú či verejnú potrebu (plagáty, vizitky, letáky, oznámenia, parte, katalógy, zoznamy ap.). Pri ich tvorbe zohrávajú veľký význam grafické a výtvarné práce. Využíva sa málo textu, čo umožňuje použiť aj písmo nevhodné pre rozsiahlejšie dokumenty alebo menej čitateľné, napr. písané, kaligrafické, lomené.

- **4.1.0.0 Lineárne písmo serifové (clarendon).** Predstavuje, čo sa týka kresby znakov, plynulejší variant egyptienky. Hrúbky hlavných a spojovacích ťahov sú mierne rozdielne, serify sú štvorhranné a rovnako hrubé ako spojovacie ťahy, sú vodorovne nasadené s oblúkovými nábehmi do hlavných ťahov, napr. Clarendon, Impressum, ITC American Typewriter, Egyptienne F, Public.

- **4.2.0.0 Lineárne serifové písmo s trojuholníkovými serifmi.** Písmo vzniklo ako titulkové v 19. storočí. Hrúbky hlavných a spojovacích ťahov sú mierne rozdielne, Serify sú výrazné, klinového tvaru, sú vodorovne nasadené s rôznymi uhlami skosenia, napr. Latin, Wide Latin, ITC Barcelona.

- **4.3.0.0 Lineárne serifové písmo s rovnými serifmi.** Písmo vzniklo ako akcidenčné na začiatku 19. storočia, niektoré jeho novodobé modifikácie sa používajú ako novinové, resp. časopisové písma. Hrúbky hlavných a spojovacích ťahov sú len nepatrne rozdielne alebo sú rovnaké, serify sú štvorhranné, doskovitého tvaru a rovnako hrubé ako spojovacie ťahy, sú vodorovne nasadené bez nábehov, stredná výška písma je zväčšená, napr. Beton, Serifa, Courier, Glypha, Rockwell.

5.0.0.0 Lineárne bezserifové statické písmo

Obsahuje grotesky 19. storočia a ich novodobé modifikácie. Sú to obľúbené písma vychádzajúce z klasistickej antikvy. Majú univerzálne použitie, používajú sa na prezentácie v elektronických médiach, sú absolútne nevhodné pre krásnu a odbornú literatúru. Kresbu môžeme charakterizovať nasledujúcimi prvkami (obr. 2):

- písmové ťahy sú málo diferencované,
- os tieňovania je málo výrazná, je vždy kolmá k účiariu (zvislá),
- nemá serif,
- otvorené ťahy mínusiek sú uzavreté, ukončené vodorovne alebo mierne šikmo,
- šírková proporcia nie je diferencovaná.

- **5.1.0.0 Lineárne bezserifové statické písmo (grotesk).** Otvorené ťahy pri mínuskach a, c, e, s sú šikmo zakončené, napr. Gothic, Akzidenz Grotesk, Slavia Grotesk.

- **5.2.0.0 Lineárne bezserifové statické písmo (neogrotesk).** Otvorené ťahy pri mínuskach c, e, s sú dôsledne vodorovné, stredná výška písma je zväčšená, napr. Univers, Helvetica, ITC Franklin Gothic.

6.0.0.0 Lineárne bezserifové geometrické konštruované písmo

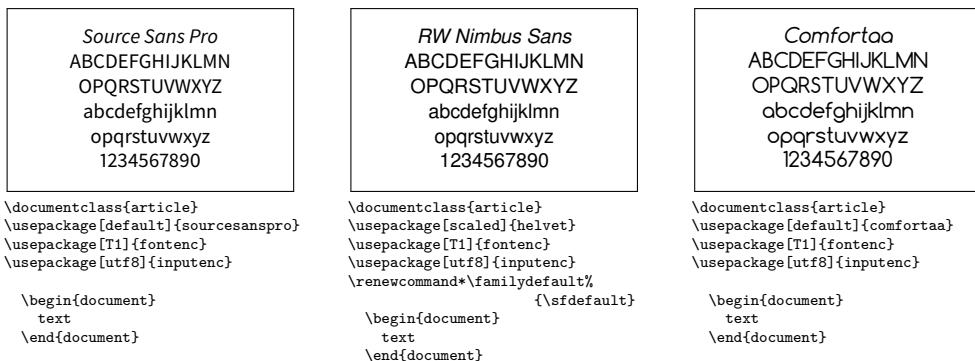
Písma tejto skupiny vychádzajú z konštruktivizmu dvadsiatych rokov 20. storočia. Konštrukcia písma je koncipovaná do geometrických tvarov (štvorec, obdĺžnik, kruh) alebo sa z nich odvodzujú. Majú univerzálne použitie, používajú



Obr. 2. Príklady lineárnych bezserifových statických písiem vygenerovaných v L^AT_EX-u

sa najmä v akcidenčnej sadzbe.¹ Nie sú vhodné pre krásnu a odbornú literatúru. Kresbu môžeme charakterizovať nasledujúcimi prvkami (obr. 3):

- nemá tieňovanie (je monolínearne),
- nemá serif.



Obr. 3. Príklady bezserifových geometricky konštruovaných v L^AT_EX-u

- **6.1.0.0 Lineárne bezserifové písmo odvodené zo štvoruholníka.** Šírková proporcia nie je diferencovaná, napr. Eurostile.

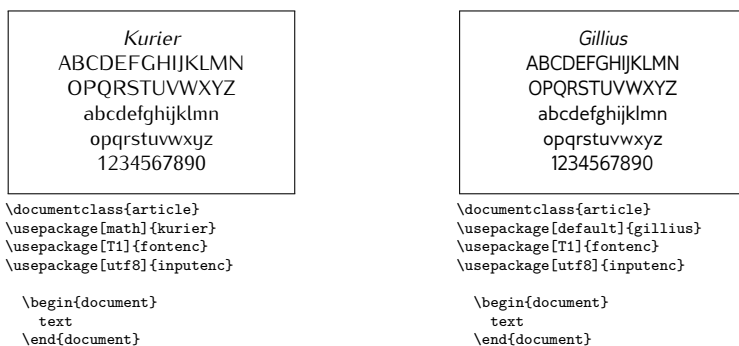
- **6.2.0.0 Lineárne bezserifové písmo odvodené z kruhu.** Šírková proporcia je diferencovaná, napr. ITC Avant Garde Gothic, Futura, Kabel, Bauhaus.

- **6.3.0.0 Lineárne bezserifové písmo odvodené z iného geometrického útvaru.** Šírková proporcia nie je jednotná, napr. Barell.

7.0.0.0 Lineárne bezserifové dynamické písmo

Konštrukcia písiem tejto skupiny vychádza z renezančnej dynamickej antikvy, nadväzuje na dynamické schémy tvarov a jednotlivé znaky starostlivo modeluje v súlade s tradíciou. Tieto písma môžeme veľmi výrazne zužovať a rozširovať. Písma sú vhodné pre noviny, časopisy a hlavne titulky, často sa používajú v akcidenčnej tlači. Kresbu môžeme charakterizovať nasledujúcimi prvkami (obr. 4):

- mierne rozdiely v hrúbkach hlavných a spojovacích ťahov,
- mierne tieňovanie,
- nemá serif,
- otvorené ťahy mínusek sú ukončené zvisle,
- výbehové oblúky pri mínuskach *a*, *c*, *e*, *s* nie sú uzavreté,
- diferencované šírkové proporcie.



Obr. 4. Príklady bezserifových dynamických písiem vygenerovaných v L^AT_EX-u

- **7.1.0.0 Lineárne bezserifové dynamické písmo so zvislou osou.** Minimálne rozdiely v šírkových proporciách, napr. Antique Olive, Frutiger.²

- **7.2.0.0 Lineárne bezserifové dynamické písmo s naklonenou osou.** Výrazné rozdiely v šírkových proporciách, písmové znaky sú vyvažované vzhľadom na doľava naklonenú os, napr. Gill,³ Gill Sans, Syntax, Maršov grotesk.⁴

²Adrian Frutiger (1928–2015) bol švajčiarsky tvorca písma, typograf a grafik, ktorý ktorý výrazne ovplyvnil smerovanie typografie na konci 20. a začiatku 21. storočia.

³Arthur Eric Rowton Gill (1882–1940) bol anglický sochár, grafik a tvorca písma.

⁴Stanislav Maršo (1910–1976) bol český grafik, typograf a tvorca písma.

8.0.0.0 Lineárna antikva

Skupina písiem (staré označenie Groteskantikva), u ktorých sa spájajú charakteristické kresbové a tvarové znaky antikvy a groteskov, Najčastejšie sa používajú v akcidenčnej sadzbe. Kresbu môžeme charakterizovať nasledujúcimi prvkami (obr. 5):

- môžu mať značné rozdiely v tieňovaní ťahov, resp. sú bez tieňovania,
- drieky a koncové ťahy sú zreteľne rozšírené,
- serify môžu byť malé, ale i skryté (takmer neviditeľné) a majú plynulý nábeh,
- šírková proporcia nie je jednotná, najčastejšie je rovnaká.



Obr. 5. Príklady písiem lineárnej antikvy vygenerovaných v L^AT_EX-u

- **8.1.0.0 Lineárna antikva s jemnými serifmi.** Krátke serify iba opticky uzatvárajú kresbu písmových znakov, často bez tieňovania, napr. Copperplate Gothic, ITC Elan, ITC Symbol, Insignia.

- **8.2.0.0 Lineárna antikva so skrytými serifmi.** Písmové ťahy sú zakončené výrazným rozšírením (skrytými serifmi), obyčajne mierne rozdiely v tieňovaní znakov, výrazné rozdiely v šírkových proporciách, napr. Friz Quadrata, ITC Friz Quadrata, Icone, Albertus, Marsia.

- **8.3.0.0 Lineárna antikva s rozšírenými koncovými ťahmi.** Písmové ťahy sú zakončené zreteľným rozšírením (skrytými serifmi), ale menej výrazným ako skryté serify, väčšinou rozdiely v šírkových proporciách a značne kontrastné písmové ťahy, napr. *Optima*.

9.0.0.0 Kaligrafické písma

Kaligrafické písma (tiež sa nazývajú **script**) vychádzajú z historicky písaných (kaligrafických) písiem a napodobňujú medirYTECKÉ písma. Obyčajne majú veľké rozdiely v šírkach ťahov. Používajú sa v akcidenčnej sadzbe a pre titulky (obr. 6). Napr. *ITC Zapf Chancery*, *Monotype Corsiva*, *Zurich Calligraphic*, *Park Avenue*, *Kuenstler Script*.



Obr. 6. Príklady kaligrafických písiem vygenerovaných v L^AT_EX-u

- **9.1.0.0 Kaligrafické dynamické písmo.** Nadväzuje na písané písma zo 16. a 17. storočia, jeho forma je odvodená z písma písaného plochým perom šikmo k účiariu.

- **9.2.0.0 Kaligrafické prechodové písmo.** Nadväzuje na písané písma z konca 17. storočia, jeho forma je odvodená z písma písaného plochým perom rovnobežne k účiariu.

- **9.3.0.0 Kaligrafické statické písmo.** Nadväzuje na písané písmo z konca 18. storočia, jeho forma je odvodená z písma písaného ostro zrezaným perom, pričom rozdiely v hrúbke ťahov sa dosahujú zmenou tlaku na pero.

- **9.4.0.0 Kaligrafické písmo voľne kaligrafované.** Je odvodené z reklamných skriptov 20. storočia, jeho forma je odvodená z kaligrafie písanej plochým štetcom.

10.0.0.0 Voľne písané písmo

Písmá vychádzajú zo súčasných rukopisných skriptov s malými rozdielmi v šírke ťahov. Majú charakter písma písaného nástrojom sa širokou stopou (napr. štetcom). Šírka ťahov je len mierne diferencovaná (obr. 7). V dlhších textoch nevyzerajú dobre, preto sa používajú iba v akcidenčnej sadzbe (hlavne osobné tlačoviny, ako sú napr. vizitky, oznámenia ap.), napr. El Greco, Brody, Mistral.



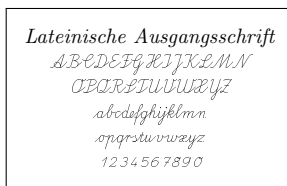
```
\documentclass{article}
\usepackage{calligra}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\usepackage{graphicx}

\begin{document}
text
\end{document}
```



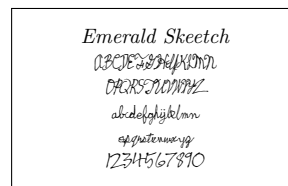
```
\documentclass{article}
\usepackage{pbsi}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

\begin{document}
{\bsifamily text}
\end{document}
```



```
\documentclass{article}
\usepackage{wela}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

\begin{document}
{\wela text}
\end{document}
```



```
\documentclass{article}
\usepackage{emerald}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}

\begin{document}
{\ECFSkeetch text}
\end{document}
```

Obr. 7. Príklady voľne písaných písiem vygenerovaných v L^AT_EX-u

11.0.0.0 Lomené písmo

Nesprávne sa tiež nazývajú Švabach. Sú to písmo s výrazne lomenými ťahmi, prešli zložitým vývojom od gotiky až po 19. storočie. Lomené písmo vznikajú

ojedinele aj v súčasnosti. Šírka kresbových ťahov je značne diferencovaná, je tu výrazné tieňovanie, lomené ťahy, komplikovaná kresba najmä pri veľkých písmenách. V minulosti (najmä v nemecky hovoriacich krajinách) sa používali na sadzbu kníh, v súčasnosti sa používajú v akcidenčnej sadzbe (obr. 8), napr. Šwabach, Alte Schwabacher, Fractura, Chaucer.

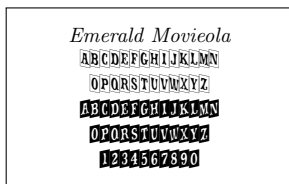


Obr. 8. Príklady lomených písiem vygenerovaných v L^AT_EX-u

Podakovanie. Príspevok vznikol s príspevom grantu KEGA 011–4/2014ŽU „Experimentálna matematika – zviditeľnenie neviditeľného“ podporeného Slovenskou kultúrno-edukačnou grantovou agentúrou.

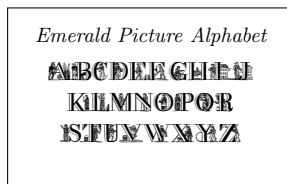
Literatúra

- [1] BRINGHURST, R.: The Elements of Typographic Style, Hartley Marks Publishers 2004, ISBN 0-88179-205-5.
- [2] PANÁK, J. – ČEPPAN, M. – DVONKA, V. – KARPINSKÝ, L. – KORDOŠ, P. – MIKULA, M. – JAKUCEWICZ, S.: *Polygrafické minimum*, Bratislava, Typoset, 2008, ISBN 978-80-970069-0-7.
- [3] ŠÍP, R.: *Typografické minimum*, Bratislava, SOU Polygrafické, Zväz polygrafie na Slovensku, 2000, ISBN 80-967598-5-X.
- [4] The L^AT_EX Font Catalogue, <https://tug.org/FontCatalogue/>.



```
\documentclass{article}
\usepackage{emerald}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
```

```
\begin{document}
{\ECFMovieola text}
\end{document}
```



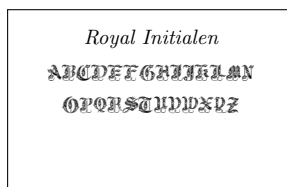
```
\documentclass{article}
\usepackage{emerald}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
```

```
\begin{document}
{\ECFPicture text}
\end{document}
```



```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\input Starburst.fd
\newcommand*\initfamily%
{\usefont{U}{Starburst}{xl}{n}}
```

```
\begin{document}
{\initfamily text}
\end{document}
```



```
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[utf8]{inputenc}
\input RoyalIn.fd
\newcommand*\initfamily%
{\usefont{U}{RoyalIn}{xl}{n}}
```

```
\begin{document}
{\initfamily TEXT}
\end{document}
```

Obr. 9. Príklady zaujímavých písom vygenerovaných v L^AT_EX-u

Kontaktná adresa

RNDr. Rudolf Blaško, PhD., Katedra matematických metód a operačnej analýzy, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovensko,
E-mailová adresa: beerb@frcatel.fri.uniza.sk

JAK SE ZMĚNIL QGIS PROCESSING MODELER VE VERZI 3

ZDENA DOBEŠOVÁ (CZ) A MICHAL KUČERA (CZ)

Abstrakt. Článek přináší přehled změn grafického editoru Processing Modeler pro tvorbu workflow diagramů v QGIS. Porovnání se týká starší verze 2 a novější verze 3. Došlo nejen úpravě rozhraní, je nová sada symbolů a byla rozšířena i funkčnost. Uživatelé si nemusí zvykat na výrazně nové prostředí. Přibyla možnost přibližování a oddalování modelu. Velká změna nastala v nabídce typů dostupných vstupních dat. Bohužel z hlediska kognice nedošlo k výraznému zlepšení rozlišitelnosti symbolů a ani sémantické transparentnosti symbolů. Základní symboly jsou stále obdélníky, změnila se jen barva výplně a jsou jiné funkční ikony uvnitř symbolů. Nový je formát ukládání dat. Uživatelé se dočkali vylepšení z hlediska funkčnosti.

Klíčová slova. vizuální programovací jazyky, grafická notace.

HOW IT CHANGED QGIS PROCESSING MODELER IN VERSION 3

Abstract. The article presents an overview of changes in the Processing Modeler for creating workflow diagrams in QGIS. The comparison applies to older version 2 and newer version 3. There was not only an interface modification; a new set of symbols and functionality was extended. Users do not have to get used to a significantly new environment. There is a possibility of zooming in and out of the model. A big change occurred in the offer of available input data. Unfortunately, in terms of cognition, there was no significant improvement in the discriminability of the symbols or the semantic transparency of the symbols. Basic symbols are still rectangles, only the fill color has changed, and there are other functional icons inside the symbols. New is the data storage format for model. However, due to usability, users have seen improvements in functionality.

Keywords. visual programming languages, graphical notation.

Úvod

Grafický editor Processing Modeler je součástí programu QGIS a slouží pro grafický návrh modelu (workflow diagramu) pro automatizované zpracování dat. Grafický způsob zápisu algoritmu se řadí mezi vizuální programovací jazyky. Je potěšující zprávou, že open source program QGIS disponuje možností používat vizuální programovací jazyk již od verze QGIS Duror 2.0 z roku 2013 podobně jako komerční GIS software [1]. S příchodem nové verze QGIS 3 v roce 2018 došlo i k podstatnějším změnám v editoru Processing Modeler [2]. Dřívější vývoj v rámci série verzí 2.x se spíš ubíral nepatrnými změnami s důrazem na zlepšení

stability komponenty. S příchodem verze 3.x došlo k výraznějším změnám. Processing Modeler se spouští stále volbou z menu Processing -> Graphical Modeler. První drobnou změnou je změna základní ikony tohoto editoru. Tři ozubená kolečka, červené, modré a šedé, nyní symbolizují modely místo původních třech malých spojených koulí. Změny a vylepšení celého editoru budou postupně popsány v článku s ilustrací změn.

1. Rozhraní grafického editoru Processing Modeler

Nejprve se zmíníme o změnách v samotném rozhraní komponenty. Základní uspořádání zůstalo beze změny. Vlevo jsou dva překryvné panely Inputs a Algorithms, mezi kterými se přepíná pro výběr dat nebo algoritmů. Součástí panelu Algorithms zůstalo nahoře velmi užitečné vstupní textové pole Search... pro automatické vyhledávání operace podle názvu. Vlastní velké okno pro návrh modelu zůstalo vpravo (obr. 1).

Změnilo se umístění dvou textových oken pro pojmenování vlastního modelu a pojmenování skupiny modelů, kam se má model zařadit. Dříve byla tato textová okna vpravo nad velkým oknem modelu, nyní se nachází úplně vlevo v panelu nazvaném Model properties. Jednoduchá filosofie způsobu zařazování modelů do skupin (Group) zůstala. Pokud nově zadáte jméno skupiny, tak její jméno začne existovat. Při zadání stejného jména skupiny u jiného modelu, jsou tyto modely nabízeny se stejné skupině v panelu Algorithms. Fyzicky se žádná struktura pro skupiny nevytváří. Je důležité zmínit, že všechny tři panely jsou nově plovoucí a uživatel je může v rozhraní přemístit podle potřeby na jiné místo než je původní pozice vlevo.

Změny nastaly v horní řádce ikon pro základní ovládání. Změnila se vlastní ikona pro spuštění modelu z ikony ozubených koleček na zelenou šipku (ikona úplně poslední vpravo). Co je však důležitější jsou čtyři nově přidané ikony přiblížení a oddálení modelu, zobrazení na sto procent a zobrazení modelu přesně do okna. To je přínos zejména při tvorbě rozsáhlého modelu, kdy nelze zobrazit celý model do okna a pracovat v něm, neboť by byly symboly moc malé. Naopak je užitečné přiblížit pouze nějakou detailní část a pracovat v ní. Možnost přiblížování a oddalování modelu umožňuje mít tvůrci neustále kontrolu nad modelem a mít přehled, kde se právě v modelu nachází.

Nově přibyly možnost exportu. Kromě původně dostupného grafického formátů PNG lze také model uložit do formátu SVG a PDF. Dostupnost editace vlastní nápovědy a postup tvorby nápovědy pro model a jeho parametry zůstal beze změny. V prvních verzích QGIS 3.0 až do verze 3.4 (Long Term Release) zmizelo tlačítko exportu do Python skriptu, které bylo dostupné ve verzi 2. Nyní je ve verzi 3.6 Noosa toto tlačítko opět dostupné. Nicméně důvodem jsou související změny ve psaní Python skriptů mezi verzemi QGISu.

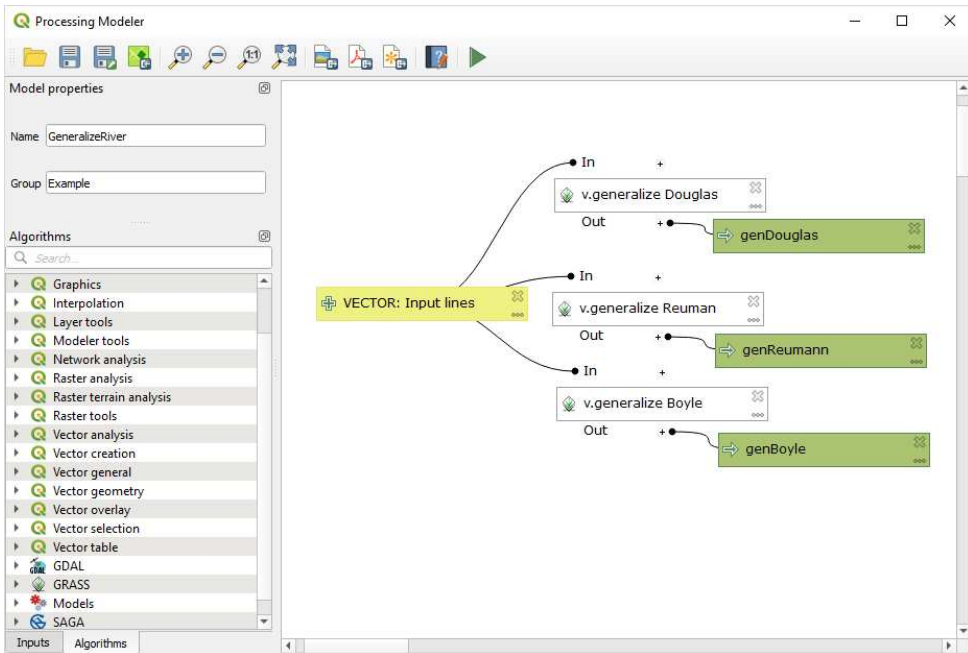


Figure 1. Rozhraní Processing Modeler verze 3 a ukázka modelu

2. Slovník symbolů

Viditelnou změnou je barevná změna základních grafických symbolů. Zůstal počet a tvar základních symbolů. Jsou tři a všechny symboly mají obdélníkový tvar. Symbolu pro operace zůstala bílá barva. Fialový obdélník pro vstupní data má nyní žlutou barvu. Modrý obdélník pro výstupní data má nyní zelenou barvu. Tato výrazná změna upozorní uživatele již na první pohled, že se jedná o model z řady QGIS 3. Porovnání starých a nových symbolů je na obrázku 2.

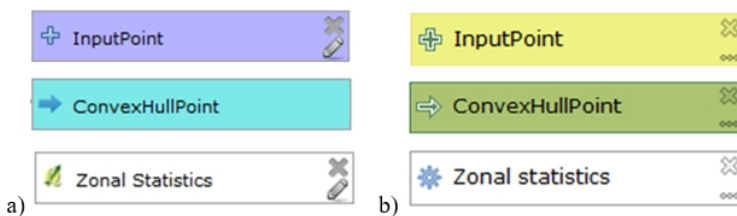


Figure 2. Porovnání základních symbolů a) verze 2.x, b) verze 3.x

Vnitřní významové ikony v levé části symbolů jsou beze změny. Ikona plus je u vstupních dat a ikona šipky je u výstupních dat. Změnila se opět barva těchto

ikon a mají bílé ohraničení. Funkční ikona křížku pro smazání symbolu, které je vpravo zůstala stejná. Změnila se funkční ikona pro editaci v pravé části symbolů. Ikona tužky pro editaci se změnila na symbol tří teček. U zeleného výstupního symbolu původně vůbec funkční ikony nebyly, ve verzi 3 se objevily. Toto není jen vizuální změna. Souvisí to s možností, že operace může mít výstupní data pouze dočasná nebo konkrétně pojmenovaná a uložená na paměťovém médiu (tak to bylo i u verze 2). Zelený symbol se tedy automaticky přidá do modelu s přidáním symbolu operace, když jsou výstupní data konkrétně pojmenována. Když ve verzi 3 přes novou funkční ikonu křížku smažeme symbol výstupních dat, tak data operace jsou automaticky změněna na data dočasná v nastavení operace. Tento způsob změny nastavení výstupních dat ve verzi 3 je rychlejší než původní editace symbolu operace.

U bílého symbolu operace se nyní objevuje více významových ikon v levé části. Dříve to byly pouze ikony zdrojových knihoven (QGIS, GRASS, SAGA,...). Nyní mají i některé GIS operace z QGISu (jako je buffer, dissolve apod.) svoji specifickou ikonu, která má asociovat význam operace. Ikony jsou ale velice malé a pouze jednobarevné, takže jejich srozumitelnost není pro uživatele moc vysoká. V případě, že se do modelu vloží jiný existující model, tak se opět v místě levé ikony objeví ikona celého Processing Modeleru - tři ozubená kola.

Na hodnocení slovníku symbolů lze uplatnit teorii Physics of Notations [3]. Tato teorie klade důraz na efektivní kognici. Ze zkušeností z hodnocení vizuálních programovacích jazyků podle této teorie nedoznala verze 3 výrazného vylepšení. Slovník má malý počet symbolů a symboly jsou přetížené, tzn. jeden symbol vstupních dat vyjadřuje velké množství různých druhů vstupních dat (rastr, vektor, číslo, text, extent apod.). Transparentnost symbolů má umožnit asociaci významu symbolu. Je škoda, že levé ikony jsou tak malé a špatně rozeznatelné. Jejich větší velikost a barevnost by zvýšila transparentnost symbolů. Dokonce lze pomýšlet na vypuštění funkčních ikon vpravo, které lze nahradit ovládním myší (dvojklik vyvolá editaci, kontextová nápověda nabídne smazání symbolu). Zbylé místo po ikonách lze využít pro delší popisy nebo větší významové ikony.

Do slovníku symbolů spadají i spojné čáry, které spojují symboly. Ty zůstaly zaoblené i v nové verzi. Navíc ve slovníku přibyla tečkovaná spojná čára, které spojuje symbol pole atributové tabulky a nadřazené vrstvy, pokud jsou vloženy do modelu. Toto znázornění je určitě přínosné pro vyjádření této souvislosti. Nicméně oblé spojné čáry jsou někdy nevhodně ohnuty, nebo jsou zakryty symboly. Sledování jejich průběhu při přílišném automatickém zakřívování není jednoduché. Škoda, že nebyly tyto čáry nahrazeny přímými spojnicemi, které by nezabíraly tolik místa v modelu.

3. Změny ve funkčnosti

Z hlediska funkčnosti je nutné zmínit výrazné rozšíření nabídky typů vstupních dat. Panel Inputs nabízí 22 typů vstupních dat (obr. 3). Původně verze 2 nabízela 9 typů vstupních dat. Požadavky na různá vstupní data jsou vyvolány i rozšířením dostupných operací. Z nabídky operací zmizela možnost vkládat operace z Orfeo Toolboxu a Model Only Tools. Zatím nelze vkládat existující Python skripty jako ve verzi 2. Nově je možné uložit model v rámci projektu QGISu, což má smysl, když je model úzce navázaný na data v projektu. Tato možnost se vyvolá stiskem ikony Embed in Project [4].

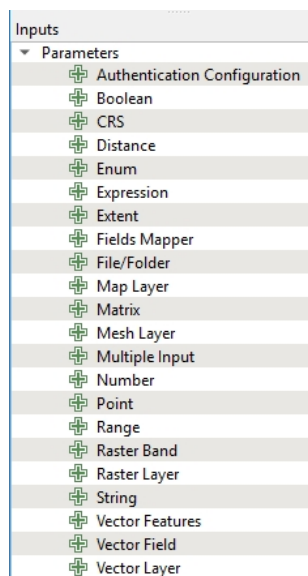


Figure 3. Nabídka vstupních dat na panelu Inputs

Došlo i ke změně formátu ukládání modelů. Stále se modely ukládají ve formě textového souboru, avšak s jiným vnitřním formátováním. Původní koncovka `.model` je nyní `.model3`. Ukázka uložení modelu ve verzi 3 je na obrázku 4. Formáty nejsou kompatibilní, nelze importovat modely ze starší verze do nové, je tedy třeba modely vytvořit znovu [5].

4. Závěr

Uvedené změny ukazují, že uživatelé obdrželi vylepšený grafický editor na tvorbu modelů zpracování (workflow), který zautomatizuje a urychlí hromadné zpracování dat v QGIS. Asi nejviditelnější změnou je změna barvy symbolů. Ostatní změny jsou jen drobné a to např. změna podoby vnitřních ikon. Lze konstatovat, že z hlediska efektivní kognice nedošlo k výraznému zlepšení. Z hlediska

```

<!DOCTYPE model>
<Option type="Map">
  <Option name="children" type="Map">
    <Option name="gdal:polygonize_1" type="Map">
      <Option name="active" type="bool" value="true"/>
      <Option name="alg_config"/>
      <Option name="alg_id" type="QString" value="gdal:polygonize"/>
      <Option name="component_description" type="QString" value="Pol">
      <Option name="component_pos_x" type="double" value="318"/>
      <Option name="component_pos_y" type="double" value="309"/>
      <Option name="dependencies"/>
      <Option name="id" type="QString" value="gdal:polygonize_1"/>
      <Option name="outputs"/>
    </Option>
  </Option>
</Option>

```

Figure 4. Zdrojový text modelu v.3

nabídky typů vstupních dat došlo naopak k výraznému navýšení typů vstupních dat. Zůstala možnost vkládání existujících modelů, což respektuje programátorský princip modularity. V rozhraní je přínosem možnost přibližovat a oddalovat model tak, aby bylo možné zobrazit i obsáhlý model.

Poděkování. Tento příspěvek vznikl s přispěním studentského grantu IGA PrF 2019 014 Univerzity Palackého v Olomouci.

References

- [1] Dobesova, Z.: Data flow diagrams in geographic information systems: a survey. Proceeding of 14th SGEM GeoConference on Informatics, Geoinformatics and Remote Sensing, Albena, Bulgaria. 2014, 541-548 s. ISBN 978-671-7105-10-0, DOI: 10.5593/sgem2014B21.
- [2] GIS Geography. The Hidden Powers of QGIS 3: 33 Truly Underappreciated Features and Plugins. <https://gisgeography.com/qgis-3/>
- [3] Moody, D. L.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering, 35(6), 756-779. 2009, doi:10.1109/tse.2009.67
- [4] Changelog for QGIS 3.4 LTR Changelog for QGIS 3.4 LTR <https://qgis.org/en/site/forusers/visualchangelog34/>
- [5] SCHWARZE, J.: Using QGIS 2.x processing model in QGIS 3.0? 2019 <https://gis.stackexchange.com/questions/277373/using-qgis-2-x-processing-model-in-qgis-3-0>.

Kontaktné adresy

doc. Ing. Zdena Dobešová, PhD., Katedra geoinformatiky, Přírodovědecká fakulta, Univerzita Palackého, 17. listoopadu 50, Olomouc, Česká republika,

E-mailová adresa: zdena.dobesova@upol.cz, <http://www.dobesova.upol.cz/>

Mgr. Michal Kučera, Katedra geoinformatiky, Přírodovědecká fakulta, Univerzita Palackého, 17. listoopadu 50, Olomouc, Česká republika,

E-mailová adresa: misakucera@gmail.com, <http://geoinformatics.upol.cz/>

SADZBA AKTUÁRSKÝCH SYMBOLOV V SYSTÉME \LaTeX

ALEŠ KOZUBÍK (SK)

Abstrakt. Aktuárske vedy je možné stručne charakterizovať ako špecifické aplikácie matematiky a ostatných prírodných, technických a ekonomických vied v oblasti poisťovníctva. Ako taká má svoju vlastnú symboliku, ktorá sa nie vždy zhoduje so symbolikou bežnou v matematike. Svedčí o tom už samotný fakt existencie piatich rozličných pozícií pre vyznačovanie rôznych vlastností poistenia alebo vodorovné odsúvanie indexov a ich zarovnanie na rôznych pozíciách pre odlíšenie typu poistenia. To prináša nové požiadavky na matematickú sadzbu. V príspevku bude predstavený balíček `actuarialsymbol`, ktorý prináša vhodné riešenia.

Kľúčové slová. aktuárske vedy, sadzba,

TYPESET OF THE ACTUARIAL SYMBOLS IN \LaTeX

Abstract. Actuarial sciences one can briefly characterize as specific applications of mathematics, natural, technical and economic sciences in the field of insurance. As such, it has its own symbolism, which does not always coincide with the symbolism commonly used in mathematics. This is evidenced by the very existence of five different positions for indicating different characteristics of insurance or horizontal shift of indices and their alignment in different positions to differentiate the type of insurance. This brings new requirements for mathematical typesetting. In this paper, the actuarial symbol package will be introduced, bringing appropriate solutions.

Keywords. actuarial sciences, typeset,

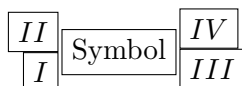
Úvod

Pre nezainteresovaných čitateľov treba hneď na úvod objasniť pojmy ako *aktuár* a *aktuárske vedy*. Ide o špecifickú oblasť aplikovanej matematiky v oblasti poisťovníctva. Medzinárodná asociácia aktúarov na svojich webových stránkach [5] charakterizuje profesiu aktuára ako *vysoko kvalifikovaného odborníka, ktorý analyzuje finančný dopad rizika na organizácie, ako sú poisťovne, správcovia dôchodkových fondov a pod. Na základe prísnych štandardov praxe uplatňuje svoje matematické znalosti na predikciu a minimalizáciu finančného dopadu neistoty*. Práve v táto multioborová kombinácia vedných disciplín so sebou prináša celý rad rozličných symbolík, ktoré je potrebné vhodne skombinovať a zosúladiť ich do zmysluplného celku.

Podrobný výklad bežne používaných symbolov je možné nájsť v učebniciach [2] alebo [1]. My si v nasledujúcom odstavci predstavíme len jej najzákladnejšie princípy tak, aby sme pochopili určité odlišnosti od bežnej matematickej sadzby v systéme \LaTeX , ktorá je popísaná napr. v [4] alebo [3]. Tak ako pri každej odlišnosti od štandardu má užívateľ dve možnosti. Buď bude všetky odchýlky riešiť sám, prostredníctvom vlastných makier, alebo využije dostupný balíček. Ako druhú alternatívu si predstavíme balíček `actuarialsymbol`, ktorý je určený pre sadzbu aktuárskych symbolov.

1. Základy aktuárskej symboliky

Aktuárske symboly sa vždy skladajú zo základného symbolu, ktorým je jednotlivé písmeno, určujúce obvykle druh poistky. Tento základný symbol potom je kombinovaný s pomocnými symbolmi, ktoré sa rozmiestňujú na 4 indexových pozíciách podľa nasledujúcej schémy:



Ako už bolo spomenuté, základný symbol určuje obvykle druh poistky. Napr. symbol E znamená poistenie pre prípad dožitia, A poistenie pre prípad úmrtia a pod. Symboly môžu byť niekedy doplnené akcentom, napr. \ddot{a} pre vyznačenie predlehotnosti. V prípade, že je základný symbol zložený z dvoch písmen, obvykle sa zobrazujú v zátvorkách. Jednotlivé pozície indexov sa potom využívajú takto:

- Na pozícii indexu I sa vyznačuje odklad resp. poistná doba. Napr. ${}_nE_x$ označuje poistnie na dožitie konca poistnej doby n rokov pre poisteného so vstupným vekom x .
- Na pozícii indexu II sa vyznačujú tzv. pomocné symboly pre rôzne dopĺňujúce údaje. Napr. 2A_x znamená, že pri výpočte sa používa diskontný faktor v^2 namiesto v .¹
- Na pozícii indexu III sa vyznačuje vstupný vek a poistná doba. Napr. $A_{x:\overline{n}}^1$ označuje dočasné poistnie na úmrtie pre poisteného so vstupným vekom x a dobou posietenia n rokov.
- Na pozícii indexu IV sa vyznačuje področnosť a iné špecifikácie. Napr. $a_x^{(m)}$ označuje poistenie doživotného dôchodoku vyplácaného m -krát ročne. Ako obvykle x predstavuje vstupný vek poisteného.

Pre označenie vybraných aktuárskych veličín používame rozšírené symboly, ktoré sú kombináciou písmen P pre bežne platené poistné, V pre poistnú rezervu, O pre odbytné a podobne. Do zátvoriek sa potom uvádza symbol pre druh poistky, ku ktorému sa hodnota vzťahuje. Celková štruktúra potom vyzerá takto:

¹Diskontný faktor zodpovedajúci úrokovej miere i je $v = \frac{1}{1+i}$. Hodnota v^2 sa potom využíva pri výpočte rozptylu ako charakteristiky rizika poistky.

$$\begin{array}{c} \boxed{II} \\ \boxed{I} \end{array} \boxed{P} \boxed{IV} \left(\boxed{\text{Symbol}} \boxed{III} \right)$$

Význam jednotlivých pozícií je rovnaký ako v predchádzajúcom prípade. Napr. ${}_kP(A_{x:\overline{n}})$ označuje bežne platené poistné za poistenie na úmrtie s poistnou dobou n rokov, vstupným vekom poisteného x a platené počas k rokov.²

Oproti bežnej matematickej sadzbe je potrebné na niekoľko odlišností, ktoré sa vyskytujú pri sadzbe aktuárskych symbolov. najmarkantnejšie zhrnieme do troch bodov:

- Pri sadzbe dvojpísmenných symbolov (DA) sú písmená v matematickom režime príliš vzdialené. V aktuárskej symbolike sú priblížené do podoby (DA) .
- Pre vyznačenie doby poistenie sa používa špecifická „strieška“ $a_{\overline{n}}$, ktorá v \LaTeX -u nie je bežne dostupná.
- Všetky pozície v symboloch sú indexy, teda horné indexy nie sú exponenty. Niekedy je potrebné ich posúvať. Napr. symbol ${}_nE_x$ má alternatívny zápis $A_{x:\overline{n}}^{\frac{1}{}}$ (na rozdiel od $A_{x:\overline{n}}^1$).

Tieto odlišnosti je možné pri sadzbe riešiť rôznymi trikmi, napr. vytvárať predné indexy $\{\}_tA_x$ pre ${}_tA_x$, vzdialenosti „doladovať“ úpravou kerningu písma a pod. je však pohodlnejšie využiť možnosti baôička `actuarialsymbol`, ktorý prináša hotové riešenia úre všetky situácie.

2. Balíček `actuarialsymbol`

2.1. Inštalácia balíčka

Balíček `actuarialsymbol` je súčasťou štandardnej inštalácie \TeX -live až od roku 2019. V starších verziách je preto potrebné z archívu CTAN alebo z adresy <https://github.com/vigou3/actuarialsymbol> stiahnuť inštaláčne súbory `actuarialsymbol.dtx` a `mosaic.jpg`.³ Súbor `actuarialsymbol.dtx` je potrebné skompilovať príkazom

```
pdflatex actuarialsymbol.dtx
```

Ako výsledok kompilácie získame súbor `actuarialsymbol.sty`, ktorý už len skopírujeme do adresára, kde ho \TeX dokáže nájsť.

2.2. Použitie balíčka `actuarialsymbol`

Tak ako pri všetkých balíčkoch, pre sprístupnenie príkazov definovaných v balíčku `actuarialsymbol` je potrebné tento balíček načítať. To vykonáme štandardným postupom, teda príkazom

²Zrejme je $k < n$.

³Obrázok `mosaic.jpg` je potrebný len ako súčasť dokumentácie.

```
\usepackage{actuarialsymbol}
```

v preambule dokumentu.

Základom pre použitie aktuárskych symbolov sa v balíčku definuje príkaz `\actsymb`. Jeho úplná syntax má tvar:

```
\actsymb[ld][lh]{symbol}{pd}[ph]
```

Ako vidíme, príkaz má dva povinné argumenty, ktorými sú samotný `symbol` a pravý dolný index `pd`. Ostatné argumenty sú voliteľné a zodpovedajú indexom, konkrétne `ld` je ľavý dolný, `lh` ľavý horný a `ph` pravý horný index. Aj keď aktuárska symbolika má svoje špecifiká, príkaz zapisujeme do matematického režimu sadzby. Použitie príkazu ilustruje nasledujúca krátka ukážka:

Zdrojový kód	Výsledok
<code>\actsymb{A}{x}</code>	A_x
<code>\actsymb[n]{A}{x}</code>	${}_n A_x$
<code>\actsymb[n][2]{A}{x}</code>	${}_n^2 A_x$
<code>\actsymb[n][2]{A}{x}[(m)]</code>	${}_n^2 A_x^{(m)}$

Porovnajme teraz jednodtlivé výstupy, pri bežnej matematickej sadzbe a použití balíčka `actuarialsymbol`. Pri sadzbe horných a dolných indexov prispôsobuje \TeX ich vzájomnú polohu. Výsledok matematickej sadzby a aktuárskeho balíčka je v tomto prípade rovnaký, čo si môžeme všimnúť na ukážkach.

Zdrojový kód	Výsledok
<code>A_x \quad A_x^2</code>	$A_x \quad A_x^2$
<code>\actsymb{A}{x} \quad \actsymb{A}{x}[2]</code>	$A_x \quad A_x^2$

Ak však dôjde na použitie predného indexu, je potrebné zaistiť horizontálne vyrovnanie pozícií indexov. Výsledok môžeme opäť porovnať na výstupoch v ukážke:

Zdrojový kód	Výsledok
<code>{}_t A_x \quad {}_t A_x^2</code>	${}_t A_x \quad {}_t A_x^2$
<code>\actsymb[t]{A}{x} \quad \actsymb[t]{A}{x}[2]</code>	${}_t A_x \quad {}_t A_x^2$

Pre príkaz `\actsymb` existuje aj rozšírená syntax v podobe:

```
\actsymb[ld][lh][typ]{Symbol}{pd}[ph]
```

Pri porovnaní so základným tvarom syntaxe vidíme, že pribudla položka `typ`. Tu môžu figurovať rôzne písmená, ako napr.

- `P` je bežne platené poistné,
- `V` je poistná rezerva,
- `O` nárok na odbytné a pod.

Použitie rozšírenej symboliky si opäť ilustrujeme na niekoľkých ukážkach.

Bežne platené poistné, poistenie na úmrtie
vstupný vek x , doba poistenia n rokov

`\actsymb [] [] [P]{A}{x:\angln}` $P(A_{x:\overline{n}})$

poistná rezerva v k -tom roku poistenia dočasného dôchodku,
vstupný vek x , doba poistenia n rokov

`\actsymb [k] [] [V]{\ddot{a}}{x:\angln} []` ${}_kV(\ddot{a}_{x:\overline{n}})$

Nárok na odbytné v k -tom roku poistenia,
poistenie dočasného dôchodku, vstupný vek x ,
doba poistenia n rokov, bežne platené poistné m -krát ročne

`\actsymb [k] [] [O]{\actsymb [] [] [P]{A}{x:\angln} [(m)]}{x:\angln} []`
 ${}_kO(P^{(m)}(A_{x:\overline{n}}))$

2.3. Dvojpísmenné symboly

Ako už bolo spomenuté v úvodných paragrafoch, niektoré základné aktuárske symboly sa skladajú z dvoch písmen. Pre vkládanie takýchto dvojpísmenných symbolov využívame príkaz `\twoletsymb`. Jeho úplná syntax má tvar:

`\twoletsymb [d] {symbol_1}{symbol_2}`

Kým význam povinných argumentov je zrejmy zkontextu, voliteľný údaj d určuje dĺžku, o ktorú sa zmenší kerningu písmen. Požitie ilustruje niekoľko jednoduchých ukážok.

`\twoletsymb {I}{\ddot{a}}` $(I\ddot{a})$
`\twoletsymb [0.8pt] {I}{\ddot{a}}` $(I\ddot{a})$
`\twoletsymb [2pt] {I}{\ddot{a}}` $(I\ddot{a})$

Pri použití tohto príkazu sa predpokladá, že tento príkaz sa nebude zapisovať priamo v jednotlivých rovniciach, ale na definovanie vlastných symbolov. Napríklad pre označenie rastúcej poistnej sumy dejujeme vlastný príkaz `\myIA` takto:

`\newcommand{\myIA}{\ensuremath{\twoletsymb [2pt] {I}{A}}}`

Pri vlastnej sadzbe potom použijeme dvojpísmenný symbol `\myIA` s výsledkom: (IA) . Podobne pri použití vo vzorci `\actsymb [] [] [P]{\myIA}{x:\angln}` získame výsledok v tvare:

$$P((IA)_{x:\overline{n}})$$

2.4. Presuny pozícií indexov

Pri aktuárskych symboloch je niekedy potrebné manipulovať s polohou indexov, ktoré, ako už bolo povedané, nehrajú úlohu exponentov ale ich polohou sa mení význam poistenia. Všeobecný príkaz pre posun horných indexov je `\nthtop` resp. dolných indexov `\nthbottom`. Ich úplná syntax má podobu

```
\nthtop[d]{cislo}{poloha}
\nthbottom[d]{cislo}{poloha}
```

Kým význam povinných argumentov je zrejmý, nepovinný argument `d` upravuje implicitnú zvislú medzeru pre daný znak. Použitie a výsledky pôsobenia oboch príkazov ilustrujú ukážky.

Zdrojový kód	Výsledok
<code>\actsymb{A}{\nthtop{1}{x}:\angln}</code>	$A_{x:\overline{n}}^1$
<code>\actsymb{A}{x:\nthtop{1}{\angln}}</code>	$A_{x:\overline{n}}^{\frac{1}{}}$
<code>\actsymb{A}{\nthtop{1}{x}y:\nthtop{2}{\angln}}</code>	$A_{xy:\overline{n}}^1$
<code>\actsymb{A}{\nthbottom{1}{x}:\angln}</code>	$A_{x:\overline{n}}$
<code>\actsymb{A}{x:\nthbottom{1}{\angln}}</code>	$A_{x:\overline{n}}^1$

3. Záver

V príspevku sme ukázali odlišnosti aktuárskej symboliky od bežnej matematickej sadzby. Pri riešení týchto detailov môžeme postupovať buď tvorbou vlastných makier pre riešenie potrebných situácií alebo použiť rozširujúci balíček, čo je bezpochyby pohodlnejší a rýchlejší postup. Vysvetlili sme a na viacerých ukážkach ilustrovali, ako používať balíček `actuarialsymbol`. V balíčku sú definované tri príkazy, ktoré umožňujú jednoducho docieľiť požadovaný výsledok.

Podakovanie. Tento príspevok vznikol s prispením grantu KEGA-041ŽU-4/2017 podporovaného Slovenskou kultúrno-edukačnou grantovou agentúrou.

Literatúra

- [1] BOWERS, N.L., GERBER, H.U., HICKMAN, J. C., JONES, D. A., and NESBITT C. J.: *Actuarial Mathematics*. Society of Actuaries, Schaumburg, IL, second edition, 1997, ISBN 978-093895-946-5
- [2] CIPRA, J.: *Pojistná matematika v praxi*. Edice HZ, Praha, 1994, ISBN 80-9014-951-0
- [3] KOPKA, H. – DALY, P. W.: *L^AT_EX – Podrobný průvodce*, Brno, Computer Press, 2004, ISBN 80-722-6973-9.
- [4] RYBIČKA, J.: *L^AT_EX pro začátečníky*, Brno, KONVOJ 2003, ISBN 80-7302-049-1.
- [5] http://www.actuaries.org/iaa/IAA/About_the_IAA/About_Actuaries/About%20Actuaries.aspx

Kontaktná adresa

RNDr. Aleš Kozubík, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská Republika,
E-mailová adresa: alesko@frcatel.fri.uniza.sk

SADZBA TABULIEK Z TABULKOVÝCH PROCESOROV V SYSTÉME \LaTeX

ALEŠ KOZUBÍK (SK)

Abstrakt. Často sa stretávame so situáciami, kedy potrebujeme do dokumentu preniesť zobrazenie tabuľky z tabuľkového procesora. Je to typické napr. pri sadzbe učebníc, kde nám nepostačujú iba tabelované hodnoty ale potrebujeme preniesť celú tabuľku aj s tabuľkovým výpočtovým algoritmom. Jednou možnosťou je vyhotoviť screenshoty tabuľkového procesora. Jej cenou je však bitmapový obrázok, ktorý nie je možné ľubovoľne zväčšovať. Druhou alternatívou je síce vektorové, ale pomerne prácne vykreslenie celej schémy pomocou TikZ-u. V príspevku predstavíme balíček `pas-tableur`, ktorý prináša riešenia väčšiny situácií vrátane zobrazenia výberu buniek či oblastí v tabuľkách.

Kľúčové slová. tabuľky,sadzba, \LaTeX .

TYPESET OF THE SPREADSHEET TABLES IN \LaTeX

Abstract. We often encounter situations where we need to transfer a table from a spreadsheet processor into a document. It is typical eg. for textbook typesetting, where not only tabulated values are sufficient but we need to transfer the whole table with the table computational algorithm. One possibility is to make screenshots of spreadsheets. However, its price is a bitmap image that cannot be magnified at will. The second alternative is a vector but rather a laborious rendering of the whole scheme using TikZ. In this paper we will introduce the `pas-tableur` package, which brings solutions to most situations, including displaying the selection of cells or areas in tables.

Keywords. tables,typeset,

Úvod

Situácie, kedy je potrebné preniesť do textu tabuľku, ktorá bola vytvorená, alebo aj prepočítaná, v niektorom z tabuľkových procesorov. Neraz je pri tom potrebné preniesť nielen samotné hodnoty z tabuľky, ale zobraziť aj výpočtový algoritmus v jednotlivých bunkách. Pri tom je už potrebné zachovať aj dizajn tabuľky tak, ako vyzerá v tabuľkovom procesore. Táto situácia je typická napríklad pri písaní učebníc, manuálov alebo príručiek pre používanie softvérových nástrojov. Jedným z riešení, ktoré sa ponúka v takejto situácii je vytvoriť screenshot tabuľky a tento ako obrázok preniesť do dokumentu. Tento prínos však znamená zaradenie bitmapového obrázka, čo znižuje kvalitu ľubovoľného zväčšenia dokumentu, alebo toto zväčšenie úplne znemožňuje pre nečitateľnosť výsledku. Druhou

alternatívou je vykreslenie celej tabuľky s využitím balíčka `TikZ`. Tak síce získame vo výsledku vektorovú grafiku, avšak postup je veľmi prácny a zdĺhavý.

V tomto príspevku si predstavíme balíček `pas-tableur`, ktorý predstavuje sadu nástrojov, ktoré do značnej miery zjednodušujú a tým urýchľujú prácu pri rekonštrukcii tabuliek pomocou balíčka `TikZ`. Autorom tohto balíčka je Stéphane Pasquet a je súčasťou distribúcie `TEXLive`. Pri svojej činnosti spolupracuje s balíčkami `TikZ` s knižnicou `calc`, `xkeyval` a `xstring`, ktoré sú automaticky načítané spolu s balíčkom `pas-tableur`. Pre prácu s týmto balíčkom postačia základné schopnosti pre prácu v typografickom systéme `TEX` resp. `LATEX` v rozsahu základných učebníc [1] a [3]. Pre samotnú prácu s vektorovou grafikou potom postačí základné oboznámenie s balíčkom `TikZ`, ako boli prezentované napr. v [2].

1. Vytvorenie tabuľky

Vzhľadom na to, že ide o vektorovú grafiku, vytvára sa tabuľka ako obrázok v prostredí `tikzpicture`. Základnými príkazmi pre vytvorenie tabuľky sú príkazy `\tableur` resp. jeho hviezdikovaná verzia `\tableur*`, ktorá umožňuje vytvárať stĺpce s rozdielnymi šírkami. Základná štruktúra príkazu pre vytvorenie tabuľky so spreadsheetovým dizajnom je:

```
\begin{tikzpicture}
\tableur[pr]{stlpce}
\end{tikzpicture}
```

Voliteľný argument `pr` udáva počet riadkov tabuľky, ktoré budú zobrazené. Ak hodnota nie je uvedená, zobrazí sa jeden riadok. Povinný argument `stlpce` potom definuje viditeľné stĺpce tabuľky. Ukážku jednoduchej tabuľky vidíme na obrázku 1. Kompletný kód, ktorý tabuľku vygeneroval potom vyzerá takto:

```
\begin{tikzpicture}
\tableur[3]{A,B,C}
\end{tikzpicture}
```

	A	B	C
1			
2			
3			

Obr. 1. Ukážka jednoduchej tabuľky, vytvorenej pomocou príkazu `\tableur`.

Ako už bolo zmienené, hviezdikovaná verzia príkazu `\tableur*` nám umožňuje meniť jednotnú šírku stĺpcov, ktorá je v balíčku preddefinovaná na 7em. Upravením príkazu podľa tohto vzoru:

```
\tableur*[3]{A/1.5cm,B/2cm,C/0.3em,D/40pt}
```

Získame tabuľku v podobe zobrazenej na obrázku 2.

	A	B	C	D
1				
2				
3				

Obr. 2. Ukážka jednoduchkej tabuľky, s upravenou šírkou stĺpcov.

2. Úpravy vzhľadu tabuľky

2.1. Rozmery buniek

Balíček `pas-tableur` preddefinováva niektoré rozmery prvkov tabuľky. Tieto rozmery sú implicitne nastavené takto:

- výška každého riadku 1.57 em,
- šírka jednotlivých stĺpcov tabuľky 7 em,
- šírka prvého stĺpca, ktorý je podfarbený a obsahuje čísla riadkov 3 em,
- počet zobrazených riadkov 1.

Tieto jednotné rozmery je možné zmeniť pomocou príkazov:

- `\tabcolwidth{s}`, kde `s` je šírka stĺpcov v akýchkoľvek prípustných jednotkách,
- `\tablineheight{v}`, kde `v` je požadovaná výška riadkov v akýchkoľvek prípustných jednotkách,
- `\tabnumlinewidth{s}` pre nastavenie šírky `s` stĺpca s číslami riadkov.

Tu treba zdôrazniť, že ide o zmenu implicitných nastavení, ktoré budú naďalej platné pre celý zvyšok dokumentu, pokiaľ nebudú opätovne pozmenené rovnakými príkazmi, alebo pre jednotlivú tabuľku tak, ako sme videli v prípade tabuľky na obrázku 2. Po sekvencii príkazov:

```
\tabcolwidth{2cm}
\tablineheight{0.5cm}
\tabnumlinewidth{1cm}
\tableur[3]{A-D}
```

teda získame tabuľku tak, ako ju vidíme na obrázku 3.

2.2. Zmeny farieb

Pre zobrazenie tabuľky je definovaných niekoľko farieb, ktoré je samozrejme možné meniť. Pre bunku záhlavia tabuliek sú definované farby `grayTopCell` a `grayBottomCell`, ktoré sú implicitne nastavené ako sivé. Ich zmenu môžeme vyvolať štandardným príkazom balíčka `TikZ`, teda `\definecolor`. Tretou farbou

	A	B	C	D
1				
2				
3				

Obr. 3. Ukážka jednoduchkej tabuľky, po definovaní nových rozmerov.

je , ktorá určuje farebnosť čiar mriežky, oddeľujúcej jednotlivé bunky v tabuľke. Tak napríklad zaradením príkazov

```
\definecolor{grayTopCell}{cmyk}{0.28 ,0.05 ,0.06 ,0}
\definecolor{grayBottomCell}{cmyk}{0.4 ,0.07 ,0.08 ,0}
\definecolor{graySepCell}{cmyk}{0.4 ,0.07 ,0.08 ,1}
```

pred samotné príkazy pre vytvorenie tabuľky pozmeníme farebnosť záhlavia, tak ako to ilustruje obrázok 4¹.

	A	B	C	D
1				
2				
3				
4				

Obr. 4. Tabuľka so zmenou farebnosti.

Niektoré ďalšie definované farby si predstavíme pri popise zobrazenia výberu bunky alebo oblasti v tabuľke. Aj tieto bude samozrejme možné meniť a určiť si farebnosť podľa vlastného výberu.

3. Vkladanie obsahu do buniek

3.1. Vloženie textu do bunky

Pre vloženie textu do bunky slúžia príkazy `\celtxt` resp. hviezdičkovaná verzia `\celtxt*`, ktorá sa využije v prípade, že potrebujeme do bunky vložiť obsah písaný v matematickom režime sadzby (napr. indexované symboly a pod.). Príkazy majú niekoľko povinných a voliteľných argumentov a ich plná syntax má tvar:

```
\celtxt [volby]{stlpec}{riadok}{text}
```

¹V tlačenom čierno-bielom dokumente nemusí byť výsledok dobre viditeľný. Preto autor odporúča vlastné experimentovanie s tabuľkami, alebo si rozdiel pozrieť v online verzii zborníka na stránkach konferencie.

Kým význam povinných argumentov je zrejmý z ich pomenovania, možné voliteľné argumenty sú takéto:

- `l` pre zarovnanie textu v bunke doľava,
- `c` pre zarovnanie textu v bunke na stred,
- `r` pre zarovnanie textu v bunke doprava,
- `width=` pre určenie šírky bunky,
- `color=` pre zmenu farby textu, ak sme použili `\tableur*`,

Ukážku tabuľky s vloženými textami vidíme na obrázku 5. príslušný zdrojový kód potom vyzerá takto:

```
\tableur*[3]{K/15mm,L/15mm,M/40mm,N/15mm}
\celtxt[width=15mm,color=red]{N}{2}{=K2^3}
\celtxt[c]{M}{3}{Text v bunke}
```

	K	L	M	N
1				
2				=K2 ³
3			Text v bunke	

Obr. 5. Tabuľka s vložením textu do buniek. Text v bunke N2 je v originále červený, farebne sa tak odlišuje ukážka možnosti zobrazenia výpočtového algoritmu v bunke.

Na obrázkoch 6 a 7 je ukážka tej istej tabuľky, kde na obrázku 6 je zobrazený výpočtový algoritmus a na obrázku 7 výsledná tabuľka po výpočte. Pre vloženie indexovaných premenných v matematickom režime sadzby bol použitý hviezdičkovaný príkaz, napr. `\celtxt*[width=10mm]{B}{1}{${a}_{n-2}$}`.

	A	B	C	D
1	n	a_{n-2}	a_{n-1}	a_n
2	2	1	1	=B2+C2
3	3	=C2	=D2	=B3+C3
4	4	=C3	=D3	=B4+C4

Obr. 6. Tabuľka so zobrazením výpočtového algoritmu Fibonacciho postupnosti.

	A	B	C	D
1	n	a_{n-2}	a_{n-1}	a_n
2	2	1	1	2
3	3	1	2	3
4	4	2	3	5

Obr. 7. Tabuľka s vypočítanými hodnotami Fibonacciho postupnosti.

3.2. Vyznačenie výberu bunky a oblasti

Stáva sa, že potrebujeme v obrázku vyznačiť výber niektorej konkrétnej bunky alebo celej oblasti buniek. Pri práci s tabuľkovým procesorom býva takáto bunka

vyznačená rámkom a príslušné záhlavia radku a stĺpca sú farebne podsvietené. V prípade celej oblasti je táto oblasť farebne podsvietená spolu s príslušnými políčkami v záhlaviach riadkov a stĺpcov. To dosiahneme aj pri reprodukcii tabuliek s balíčkom `pas-tableur` pomocou príkazov `\selecCell` a `\multiSelec`. Ich plná syntax má potom tvar:

```
\selecCell{stlpec}{riadok}
```

resp.

```
\multiSelec{stlpec-p1}{riadok-p2}
```

kde hodnoty `p1`, `p2` udávajú koncový stĺpec resp. riadok vyznačeného výberu.

Výber jednej bunky potom môžeme vyznačiť pomocou kódu

```
\tableur[4]{A-C}
```

```
\selecCell{B}{2}
```

príčom výsledok vidíme na obrázku 8.

	A	B	C
1			
2			
3			
4			

Obr. 8. Ukážka tabuľky s vyznačením výberu bunky B2.

Pre vyznačenie výberu celej oblasti potom použijeme sekvenciu:

```
\tableur[4]{A-C}
```

```
\multiSelec{B-2}{C-3}
```

Jej výsledkom je potom tabuľka s vyznačením výberu celej oblasti buniek tak, ako ilustruje obrázok 9.

	A	B	C
1			
2			
3			
4			

Obr. 9. Ukážka tabuľky s farebným vyznačením výberu súvislej oblasti.

Pre vyznačenie vybraných buniek resp. oblastí sú definované ďalšie tri farby `blueSelecCellTop`, `blueSelecCellBottom` a `blueSelec`. Túto farebnosť opäť

pozmeníme štandardným príkazom balíčka `TikZ`. Ak predefinujeme farby pomocou príkazov:

```
\definecolor{blueSelecCellTop}{cmyk}{0.32,0.47,0,0}
\definecolor{blueSelecCellBottom}{cmyk}{0.42,0.67,0,0.1}
\definecolor{blueSelec}{cmyk}{0.13,0.26,0,0}
```

tak tabuľka 9 zmení farebnosť do podoby na obrázku 10.

	A	B	C
1			
2			
3			
4			

Obr. 10. Ukážka tabuľky s farebným vyznačením výberu súvislej oblasti so zmenou farebnosti označenia.

3.3. Spolupráca s balíčkom `TikZ`

Ako už bolo spomenuté, balíček `pas-tableur` je vlastne predstavuje balíček `makier` založený na balíčku `TikZ`. To umožňuje efektívnu spoluprácu týchto dvoch balíčkov v rámci jedného obrázku. Výhodou je, že balíček `pas-tableur` vytvára jednotlivé bunky tabuľky ako pomenované uzly. Tieto mená sú priradené ako `cell` nasledované bežným označením pozície bunky, teda napr. `C-3`. Tak je možné doplnením príkazov balíčka `TikZ` vytvoriť napríklad ukazovátka na konkrétnu bunku tabuľky, čo ilustrujeme na obrázku 11. Do zdrojového kódu pritom stačí po vygenerovaní tabuľky vložiť jednoduchý príkaz balíčka `TikZ`.

```
\draw[<-,>= latex] (cellB-2.center) to [bend right =30]
($ (cellB-2)+(2,-1.7)$) node[right]
{Toto je bunka B2};
```

	A	B	C
1			
2			
3			

Toto je bunka **B2**

Obr. 11. Tabuľka s ukazovátkom na bunku `B2`.

Tento príkaz potom vykreslí šípku, ktorá ukazuje do stredu uzla, teda bunky B2. Pozíciu v bunke, na ktorú ukazuje šípka je možné zmeniť označením pozície bežným v balíčku TikZ. Ak teda namiesto `cellB-2.center` použijeme napríklad označenie `cellB-2.south`, šípka bude ukazovať na dno bunky a pod.

4. Záver

V príspevku sme ilustrovali možnosť zjednodušenia práce pri editácii tabuliek prenášaných z tabuľkových procesorov. Túto problematiku významne zjednodušuje balíček `pas-tableur`, čo bolo ilustrované na viacerých príkladoch. Jeho činnosť je založená na niekoľkých jednoduchých makrách a súčinnosti s balíčkom TikZ.

Poďakovanie. Tento príspevok vznikol s príspevom grantu KEGA-041ŽU-4/2017 podporovaného Slovenskou kultúrno-edukačnou grantovou agentúrou.

Literatúra

- [1] KOPKA, H. – DALY, P. W.: *L^AT_EX – Podrobný príručka*, Brno, Computer Press, 2004, ISBN 80-722-6973-9.
- [2] KOZUBÍK, A.: *naučím vás kresliť alebo predstavenie balíčka TikZ*, Zborník príspevkov medzinárodnej konferencie OSSConf 2012, 2.–4. júla 2012, Žilina, str.91–96. ISBN 978-80-970457-2-2.
- [3] RYBIČKA, J.: *L^AT_EX pro začátečníky*, Brno, KONVOJ 2003, ISBN 80-7302-049-1.

Kontaktná adresa

RNDr. Aleš Kozubík, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská Republika,
E-mailová adresa: alesko@frcatel.fri.uniza.sk

AUTOMATION OF MASS COMPUTER CLASSROOMS MANAGEMENT USING ANSIBLE

MATEJ MADEJA (SK), MIROSLAV BIŇAS (SK) AND JAROSLAV PORUBĀN (SK)

Abstract. Managing a large number of computers has been generally a major challenge for many years, especially at universities. Linux systems administrators are mostly familiar with scripting that is quite reliable, but managing Windows-based devices is still an issue. Many tools try to help with the management of such devices, e.g. using group policies of the operation system or external tools. This paper describes the use of IT automation tool *Ansible* for Windows-based computer classrooms administration at the Technical University of Košice. The paper describes 2 years of practical experience at university and presents particular implementations within the project *aninstall*.

Key words and phrases. automatisisation, ansible, mass computer management, administration, installation.

AUTOMATIZÁCIA MASOVEJ SPRÁVY POČÍTAČOVÝCH UČEBNÍ POMOCO U ANSIBLE

Abstrakt. Správa veľkého počtu počítačov je vo všeobecnosti veľkou výzvou už dlhé roky, a to najmä v univerzitnom prostredí. Administrátori linuxových systémov sú väčšinou zvyknutí používať skriptovanie, ktoré je relatívne spoľahlivé, problémom je však správa zariadení s operačným systémom (OS) Windows. Rôzne skupiny nástrojov pomáhajú so správou takýchto zariadení, napríklad využitím skupinovej politiky systému alebo externými nástrojmi. V článku popisujeme využitie automatizačného nástroja *Ansible* pre správu počítačových učební s OS Windows na Technickej univerzite v Košiciach. Článok popisuje získané skúsenosti počas jeho 2 ročného používania v praxi a prezentuje aj konkrétne riešenia v rámci projektu *aninstall*.

Kľúčové slová. automatizácia, ansible, masová správa počítačov, administrácia, inštalácia.

Introduction

Software in today's operating systems is a large part of the user experience and end-user motivation to work with the device. In particular, for user are important the simplicity and purpose of the software. The user uses the application in order to simplify or speed up a particular task, meaning software brings a benefit to him/her. To use the software, it is also necessary to install, regularly update and, in general, manage the application. Most of mentioned tasks are often covered by an operating system (OS). According to Bacchus [3] in today's OS can be seen an effort to simplify the software maintenance process so it is as automated as possible and relieve the end user from unnecessary tasks. At the same time, it is

an effort to perform software maintenance activities at the time when the device is not used or when the user does not mind.

Software management of one device is relatively sufficient covers the operating systems with end-user cooperation. The issue is management of multiple devices at the same time, such as at large companies or schools that manage hundreds to thousands of devices at once. The reality is that devices are often divided between multiple administrators who manage a particular group of devices. The software update is usually handled by the operating system itself, but not always in the desired time. At universities it often happens that an OS update starts during the test when the device usage time is limited. A similar situation can also occur in a business environment.

The biggest issue is installing new and configuring already installed software. If the administrator manages 50 computers, it is necessary to perform the same operations on each device, which is very time consuming. In addition, errors can occur during manual configuration, therefore, different devices can be differently set up and device inconsistencies may occur with subsequent difficult troubleshooting. The variety of target operating systems also makes installation difficult.

This paper describes the project *Anstall*¹, used for mass management of computer classrooms at *Department of Computers and Informatics of Technical university of Košice*. Project is available at <https://github.com/madeja/anstall> and is based on open-source IT automation tool *Ansible*, focusing mainly on OS *Windows*. In the following sections, some approaches to support mass management of devices, the design of the project structure and practical usage experiences are described.

1. Software distribution approaches

There are currently several approaches that help distribute software and can be categorized into the following groups:

Cloning: The initial installation is done on one device and then a system or a disk image template is created. Finally the image is distributed to other computers. This is only possible if the hardware features of all computers are the same, otherwise deployment problems may occur, for example, not all OS can adjust the hardware change or the disk image will not be usable due to lack of space on the target device. This is quite useless for deploying software after it is already installed, but it's very useful for the initial setup. Examples are *Clonezilla*², *HDCone*³ or *Macrium Reflect*⁴.

¹Title created by joining words *ansible* and *install*.

²<https://clonezilla.org/>

³<https://www.miray.de/products/sat.hdclone.html>

⁴<https://www.macrium.com/reflectfree>

Scripting: It is possible to prepare scripts and/or logon scripts to run actions (e.g. install new software) on a system. This solution is platform dependent and assumes scripting knowledge for a particular O shell. For example, for Unix-based systems it is possible to use *bash* scripts, for OS Windows using Powershell *cmdlets* [2]. The disadvantage of this solution is the high complexity and the necessity of programming relatively complex programs, where it is often necessary to wait for the operation completion, wait for resources, verify the availability, etc.

Group Policies: According to [5] if a windows installer file (MSI) is available, so the software is packaged, installing is pretty easy. However, this solution is very limited to MSI files and cannot guarantee the same environment for each device. For example, OS settings may be manually changed and the current configuration may cause installation failure, thereby failure identification and special attention to each device is required.

MS Intune: Microsoft ⁵ offers the ability to manage computers and mobile devices through the cloud and also offers *Intune for Education*⁶, which was created exclusively for school purposes, but both are proprietary and charged. The functionality is very similar to our solution but the ability to manage all devices from a browser provides a more user friendly environment. Since *Ansible* is primarily targeted on Unix-based OS, it cannot provide all the benefits of the *Intune*. On the other hand, it provides a more global system configuration, such as service control, registers, etc.

Other vendors: Specialized tools that are mostly proprietary and provide remote computer management. However, these solutions are more business oriented and often not adapted to the university environment.

Very often solution are so-called *thin clients* that have a different architecture organization in the network, which consists of a server and clients. In recent years thin clients have proven to be very advantageous in the educational environment (see [1, 4, 6]), especially for easy client management, high availability and energy efficiency. Despite mentioned facts these solutions are often not enough powerful for programming courses, so classical classrooms with multiple computers are often built today in technical schools.

2. *Anstall* as a general purpose management solution

Because funds for such tools, mentioned in Section 1, mostly are not reserved in the education institutions, we were looking for an open-source solution with the ability to manage multiple Windows-based nodes. Since 2017, when the *Ansible* 1.8 released with Windows modules support, we started use it as the main computer management platform. *Anstall* project is the result of our two-year

⁵<https://www.microsoft.com>

⁶<https://www.microsoft.com/en-ca/education/intune/default.aspx>

experience of installing various software for different courses at our university. All directory paths presented in the following sections refer to the *Anstall* project.

2.1. Task groups

For grouping relevant tasks to be performed on target devices we use ansible playbooks. In its basis, ansible uses adhoc task execution mode, using playbooks (`./playbooks`) tasks can be performed in a specific order, which is very well suited to deploying complex setup. In our solution, a playbook represents one classroom and it can contain multiple plays in a YAML file (see Listing 1). Particular play executes different roles where a role represents a group of tasks with a specific goal, e.g. in Listing 1 *general* role includes base manipulation with computers (reboot, messaging user, services management, etc.) and *courses* role includes tasks focused on software installation for courses taught in the classroom (install new software, configuration, etc.). Sometimes it is not necessary to run all plays of the playbook, so we use tags that distinguish a particular play or play group, that creates a set of related logical units.

Listing 1. Example of play with 2 roles for classroom B512 playbook.

```
- name: General manipulation with computers and courses install
  hosts: b512
  roles:
    - general
    - courses
  gather_facts: no
  tags:
    - general_courses
```

Each play performs defined roles, which are contained in `./playbooks/roles` directory. The role is a separate unit for which variables, templates, necessary files and the like can be defined. The most important part is the **tasks** in particular role directory, where the tasks to be performed are located. The tasks are performed by modules⁷, which creates a base of ansible automation. We did not use templates in our case because we did not implemented such a complex configuration. The necessary files can be distributed using the **files** directory placed in particular role using classical copying or by downloading using http protocol.

2.2. Different classrooms installation and permissions

Roles can be used multiple times and for different classrooms, so they should be implemented for general use because their main advantage is simple distribution by copying. Devices in each classroom are defined by thier IP addresses in the

⁷https://docs.ansible.com/ansible/latest/modules/list_of_windows_modules.html

`./hosts` inventory file and can be grouped, too (Listing 2). One device can also be included in multiple groups. It is possible to create multiple inventory hosts files and use a specific one when executing ansible (using `-i` switch of `ansible-playbook` utility), but in our case each group is represented by a specific playbook. This approach has proven to be more useful for sleeping than to define separate inventory files. This approach has proven more useful in managing classrooms than defining separate inventory files.

Listing 2. Hosts file defining groups by classrooms.

```
[b512]                # classroom B512
147.232.34.215
147.232.34.205

[a537]                # classroom A537
147.232.34.227
147.232.34.228
```

Each computer has both administrator (admin) and user account (student). The administrator account is considered main maintenance account, e.g. for software installation, so for remote control it is necessary to provide login and password for individual classrooms (groups). Variables of this type can be defined directly in the inventory files, but for clarity we keep them in separate files for every single classroom (`./group_vars`), example in Listing 3. If it is necessary to install a user specific software, admin account installation must be copied for the student account, so it is not possible to install under another account. For storing passwords it is recommended to use Ansible Vault⁸, for presentation purposes we store password as plain text.

Listing 3. Example of classroom specific variables.

```
ansible_user: admin
ansible_password: mypwd
ansible_port: 5986
ansible_connection: winrm
ansible_winrm_server_cert_validation: ignore
```

2.3. Package manager for Windows

There is the *Chocolatey*⁹ package manager for Windows that makes it easy to install new software. In the past, the package manager had to be pre-installed and subsequently ansible could use it. In the current version, ansible offers automatic installation of *Chocolatey* when using the `win_chocolatey` module.

⁸<https://docs.ansible.com/ansible/2.4/vault>

⁹<https://chocolatey.org/>

2.4. Running playbooks

The best way how to execute an ansible playbook is using a server that is accessible by all nodes that need to be configured. This way it is possible to make a remote installation from anywhere. It is possible to use following statement to run a specific playbook:

```
ansible-playbook playbooks/b512.yml \  
    --tags "prog-block , programming-course "
```

Defining tags only a specific logical group of tasks or plays can be performed. If you need to run multiple playbooks at the same time, you can create a new playbook, which includes other playbooks (Listing 4).

Listing 4. Multiple playbook execution, using playbook of playbooks.

```
- include: playbook-one.yml  
- include: playbook-two.yml
```

3. Special solutions from practice

The academic environment has led us to special solutions to the problems that have occurred. In the sample examples of *aninstall* project following solutions are partially implemented.

3.1. Daily cleanup

When working with computers students often leave changes to the file system that are not needed for others. In general, most file-spam folders in Windows OS are *Downloads*, *Desktop*, *Documents*, etc. In Windows there is possible to create a temporary user account with automatic deletion of changes made. However, this method is too computationally difficult because the account is created every time the user log in and deleted immediately after user logout. The other issue is account configuration, which needs to be implemented after every recreation.

Our solution is to define the particular structure and files to be included in the folders. E.g. *Downloads* and *Documents* directories can be empty for our purposes, on the other hand, *Desktop* directory will contain links to specific applications (see role `./playbooks/roles/cleanup`). Other files can be removed periodically, in our case every day at night.

3.2. Faster licence key deployment

Some tools, e.g. integration development environment (IDE), require manual license confirmation on each device. In this case, it is difficult to automate this action and needs to be done manually. Also in this case ansible can be partially helpful, e.g. copying the license key file into clipboard using windows Powershell (module `win_shell` and executing `type license-key.txt | Set-Clipboard`).

The key is automatically copied to clipboard and maintainer only needs to paste it into the desired field.

3.3. Automatic updates

Automatic updates are often a nightmare, as they often start to execute at system boot or shutdown, which is the worst time period in a university environment. Our solution is to disable Windows update service and enable it only if it is needed or periodically in the night hours. During updates, maintainers should be specially careful because ansible configuration on destination device can be disrupted, e.g. the Windows 10 version 1809 update caused ansible to malfunction and manual re-configuration has been required. For updating Windows there is a `win_update` module which is very helpful. An example of our solutions is presented in `./playbooks/roles/general`.

3.4. Exam restrictions

Restricting the internet to prevent cheating during exams is a common academic issue. We use the classic `win_shell` module to modify firewall settings causing blocking/unblocking necessary websites in a second. Using `win_msg`, we are able to show instructions for student before the exam, where they can find materials they are allowed to use and how to behave during the exam. If a student is using a prohibited application it is possible to alert him/her without disturbing other participants or terminate the application. An example of internet restriction is presented in `./playbooks/roles/general`.

4. Conclusions and future work

This paper presented a mass management solution used for computer classrooms at the *Technical University of Košice* using automation IT tool *Ansible*. We created project *Anstall*, whose structure was developed over two years of experimentation. The authors described the current possibilities of software distribution approaches and subsequently described the possibilities of the *Ansible* tool and its unrivaled possibilities suitable not only for the university environment.

The solutions for interesting issues, often arising during computer management of multiple computers at universities, has been described, such as daily user account cleanup, optimization of license key deployment, controlling the automated updates and setting internet restrictions for exams. The project is also publicly available and ready to use by other universities or companies.

Ansible offers much more possibilities than described in this paper, in which we highlighted the most important in terms of use at a university. The structure of the project is continuously changes and is optimized depending on the teachers' software requirements, used for teaching. In the future, we would like to fully

automate the management of computers with Wake-on-LAN support, because currently it is necessary to turn on computers manually.

Acknowledgment. This work was supported by project KEGA No. 053TU-KE-4/2019: Learning Software Engineering via Continues Challenges and Competitions.

References

- [1] AHMED, S., GHASHEM, I. A., AALSALEM, M. Y., AND KHAN, W. Z. Thin client technology for higher education at universities of saudi arabia: Implementation, challenges and lesson learned. In *2017 International Conference on Computer and Applications (ICCA)* (Sep. 2017), pp. 195–199.
- [2] AIELLO, J., SCHONNING, N., MARRAMAQUE, R., AND WHEELER, S. Cmdlet overview, 09 2016. URL: <https://docs.microsoft.com/en-us/powershell/developer/cmdlet/cmdlet-overview>.
- [3] BACCHUS, A. Microsoft to make “easier, faster” updates starting with windows 10 version 1809, 08 2018. URL: <https://www.onmsft.com/news/microsoft-to-make-easier-faster-updates-starting-with-windows-10-version-1809>.
- [4] MAHDZAR, R. Using technology to improve education for cambodian children, 2015. URL: <http://geeksincambodia.com/using-technology-to-improve-education-for-cambodian-children/>.
- [5] MICROSOFT CORPORATION. How to use group policy to remotely install software in windows server 2008 and in windows server 2003, 06 2018. URL: <https://support.microsoft.com/en-us/help/816102/how-to-use-group-policy-to-remotely-install-software-in-windows-server>.
- [6] WORLD EDUCATION, INC. Technical brief, using computer technologies to improve basic education in cambodia: Thin client labs, 2013. URL: https://www.worlded.org/WEIInternet/inc/common/_download_pub.cfm?id=13309&lid=3.

Contact addresses

Ing. Matej Madeja, Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia,
E-mail address: matej.madeja@tuke.sk, <http://madeja.github.io/>

Ing. Miroslav Biñas, PhD., Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia,
E-mail address: miroslav.binas@tuke.sk

doc. Ing. Jaroslav Porubän, PhD., Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovakia,
E-mail address: jaroslav.poruban@tuke.sk

ZPRACOVÁNÍ TEXTŮ A ZÁVĚREČNÉ PRÁCE

JIŘÍ RYBIČKA (CZ)

Abstrakt. Závěrečné práce představují vizitku každé vzdělávací instituce a jsou na ně kladeny různorodé požadavky. Pracoviště ÚI PEF MENDELU se dlouhodobě zaměřuje na formální stránku (nejen) závěrečných prací a jeho výuka zaměřená na zpracování textů je obsažena v různých předmětech. Příspěvek přináší hodnocení možnosti ovlivnit kvalitu formální stránky závěrečných prací.

Klíčová slova. zpracování textů, výuka, závěrečné práce.

TEXT PROCESSING AND THESES

Abstract. The final theses represent the visage of each educational institution and are up to them different requirements. The Department of informatics of the FBE MENDELU is for a long time focused on the formal side of (not only) final theses. Lessons focused on the word processing taught by Department of informatics is contained in various courses. This paper brings evaluation of the possibility to influence the quality of the formal aspects of the theses.

Keywords. text processing, teaching process, theses.

Úvod

Oblast zpracování textů počítačem má na Ústavu informatiky Provozně ekonomické fakulty Mendelovy univerzity v Brně dlouhodobě pevné místo. Cílem výuky této oblasti bylo a je dosáhnout co nejvyšší úrovně formální stránky dokumentů, a to především závěrečných prací studentů.

Změnit všeobecně známý pohled na zpracování dokumentů, ať už po stránce typografické, technologické nebo organizační, je ovšem náročná, dlouhodobá a systematická práce. Vybudovat určité povědomí o této oblasti a posunout standard prací na úroveň neobsahující alespoň nejhrubší chyby a zásadní nedostatky vyžaduje integrovat důležité pasáže do řady předmětů a spojit je s dalšími logicky navazujícími částmi učiva. Tento proces má neustále svůj vývoj a do značné míry lze konstatovat, že často musí překonávat jistý odpor ze strany studentů i pedagogů a také musí respektovat těžko překročitelné mantinely dané hodinovou dotací a pohledem vedení fakulty.

1. Vliv výuky

1.1. Kde se zpracování textů učí

Oblast působnosti pedagogů ústavu informatiky se neomezuje pouze na domovskou fakultu, ale přesahuje i na další součásti univerzity a v jednom případě i na jinou brněnskou vysokou školu.

V současné době se kapitoly týkající se zpracování textů v nějaké podobě vyskytují v předmětech souhrnně uvedených v tab. 1.

Tabulka 1. Seznam předmětů obsahujících kapitolu z oblasti zpracování textů na počítači

Název	Součást	P/V	Hod.	Technologie
Bakalářský seminář	PEF MENDELU	P	7	T _E X
Zpracování textů na počítači	PEF MENDELU	V	42	T _E X, Word
Moderní didaktická technika prez.	ICV MENDELU	P	22	Word
Moderní didaktická technika komb.	ICV MENDELU	P	8	Word
Informatika v regionálním rozvoji	FRRMS MENDELU	PV	10	Word
2 předměty	DVPP	P	8	Word
Zpracování technické dokumentace	FAST VUT	V	26	Word (T _E X)
Informatika	FAST VUT	P	15	Word

Vysvětlivky zkratk:

P/V – povinný/volitelný předmět

Hod. – hodinová dotace (celkem za předmět)

DVPP – vzdělávání v rámci akreditovaného studia Koordinátorů informačních a komunikačních technologií na základních a středních školách

PEF – Provozně ekonomická fakulta

ICV – Institut celoživotního vzdělávání

FRRMS – Fakulta regionálního rozvoje a mezinárodních studií

FAST VUT – Stavební fakulta Vysokého učení technického v Brně

Od roku 2018 je v procesu akreditace studijní program Ekonomika a management a studijní program Hospodářská politika a správa, v nichž se nachází předmět Moderní informační technologie s kapitolou zpracování textů o rozsahu 30 hodin, je vyžadována technologie Word.

1.2. Co se učí

Jak vyplývá z tab. 1, jedním z mantinelů je použitá technologie. V tabulce je stručně uvedeno označení T_EX pro technologii založenou na principu T_EXu (jde konkrétně o XeL_AT_EX) a označení Word pro technologii založenou na principu textového procesoru Word (podle dostupných finančních prostředků se používají různé verze, nikde však není použita open source alternativa).

Koncepce výuky však není použitou technologií ovlivněna beze zbytku, neboť podstatnou součástí jsou vždy typografická pravidla a obecné technologické zásady využitelné při tvorbě dokumentů.

Hodinová dotace je rozvrstvena od minimální hodnoty 7 h až po maximální 42 h, od toho se také odvíjejí cíle, kterých se snažíme v daném předmětu dosáhnout. Za úspěch lze považovat zejména hodinové dotace v povinných předmětech, protože tam může mít systematické působení největší dopad.

Na závěrečné práce jsou přímo navázány tři z uvedených předmětů – Bakalářský seminář na PEF, Moderní didaktická technika na ICV a Zpracování technické dokumentace na FAST. Bakalářský seminář má ovšem velmi nízkou dotaci, protože při jeho konstrukci se předpokládalo, že potřebné znalosti získají studenti v předchozím studiu (zejména ve volitelném předmětu Zpracování textů počítačem) a v bakalářském semináři se praktická část doplní pouze o práci s předpřipravenou šablonou a s literárními citacemi. Předmět vznikl na základě analýzy závěrečných prací, kterou provedla Devátová (2009). Předmět Zpracování technické dokumentace na Stavební fakultě je volitelný a studuje ho mizivé množství studentů, přestože je zařazen mezi tzv. volné předměty, které si může zapsat student z libovolné fakulty VUT. Nejvýznamnějším z této trojice je tedy předmět Moderní didaktická technika na ICV, jehož prakticky celý obsah je věnován úpravě závěrečné práce. Název je sice poněkud zavádějící, ale to souvisí s uvedenými mantinely, do nichž je potřebné se při konstrukci akreditovaného studia vejít.

1.3. Proč Word a jak na to

Otázkou, proč se ve výuce preferuje právě Word a nikoliv něco jiného (minimálně volně dostupné alternativy téhož principu), se v této chvíli budeme zabývat spíše okrajově – důvodů je více a jejich rozbor není pro zaměření příspěvku podstatný. Co je ovšem podstatné, jsou znalosti tohoto nástroje, se kterými studenti přicházejí z předchozích škol. Lze říci, že v mnoha ohledech pracujeme s takovými funkcemi a akcentujeme takové přístupy, s nimiž se většina studentů ve svém předchozím (samo)vzdělávání nesetkala. Boříme tím zakořeněné mýty týkající se způsobu zpracování dokumentů, což samo o sobě přirozeně musí vzbuzovat jistou nedůvěru až odpor. Do značné míry vzniká i nekonzistence chování fakulty – podíváme-li se například na „oficiální“ doporučení úpravy závěrečné práce na FRRMS (Borůvková, 2010), je v přímém rozporu s doporučeními prezentovanými ve výuce na téže fakultě. Studenti tím mohou být zmateni, ale horší je, že vedoucí závěrečných prací jiné informace nemají a tlačí své studenty do tvorby zmetkových dokumentů.

Word je velmi problematický program. Lze objektivně prokázat, že je plný různých koncepčních a programátorských nedostatků či přímo chyb od drobných až po zásadní a hrubé. Je-li používán pro tvorbu kvalifikačních prací, pak jde o užití,

na něž vůbec není vhodný, a to zejména pro značnou funkční nespolehlivost a vytrvalou nekompatibilitu souborových formátů.

Zeptáme-li se orientačně studentů například na ICV, FRRMS nebo FAST, v jakém programu dělali nebo hodlají dělat svou kvalifikační práci, téměř vždy je v dané skupině ve 100 procentech jmenován právě Word. Skutečně přímo Microsoft Word, nikoliv jeho libovolná volně dostupná alternativa. Jen nemalým dlouhodobým úsilím na PEF je dosažen stav, že u studentů nestudujících informatiku a tam některý z nich použije L^AT_EX (s připraveným sázecím stylem).

Chceme-li tedy oblastí zpracování textů oslovit podstatnou část studentů a pokusit se je připravit na tvorbu kvalifikační práce, která vyhovuje alespoň základním formálním požadavkům, je volba technologie bohužel vcelku jednoznačná. K tomu rovněž přispívá i skutečnost, že norma ČSN 01 6910 (2014) skrytě obsahuje implementační doporučení platná právě pro tento systém.

Celý postup výuky lze shrnout do následujících kroků:

1. **Pravidla pravopisu a příprava hladkého textu.** Zahrnuje volbu základního knižního písma, speciální znaky a jejich použití v hladkém textu.
2. **Rozpoznání jednotlivých prvků dokumentu.** Upravovaný dokument je podroben analýze, v níž jsou určeny významy jednotlivých prvků – nadpisy, citáty, položky bibliografických citací, zdůrazněné texty atd.
3. **Návrh typografické úpravy.** Řeší se vzhledová reprezentace rozpoznávaných prvků, k čemuž se využívají typografická pravidla. Aplikují se dvě zásady – zásada jednotnosti a zásada kontrastu.
4. **Technologická implementace navržené typografické úpravy.** Zatímco předchozí body postupu jsou obecné a platí prakticky pro jakoukoliv technologii, v tomto kroku se plně řeší možnosti použitého programu. U Wordu jsou jednotlivé prvky reprezentovány pomocí stylů, jejichž parametry a návaznosti umožňují zahrnout stanovená typografická pravidla a navíc vedou k velmi efektivnímu formátování a údržbě dokumentu.

Pro samotnou tvorbu závěrečných prací jsou k dispozici šablony (XeL^AT_EX, Word), takže v některých předmětech jsou podrobněji popisovány a cvičně využívány.

1.4. Co produkují studenti

Budeme se nyní věnovat předmětu Moderní didaktická technika vyučovaném na ICV MENDELU. V prezenční formě je hodinová dotace 22 hodin, v kombinované pouze 8 hodin ve dvou čtyřhodinových blocích. Mezi požadavky na ukončení pro prezenční a kombinované studenty však není žádný rozdíl. Pro úspěšné zakončení předmětu je potřebné vytvořit seminární práci na téma z oblasti didaktické techniky (to je věcná část práce), ale tuto práci je nutné vypracovat podle zásad zpracování dokumentů (formální část práce). Zdrojem obsahu může

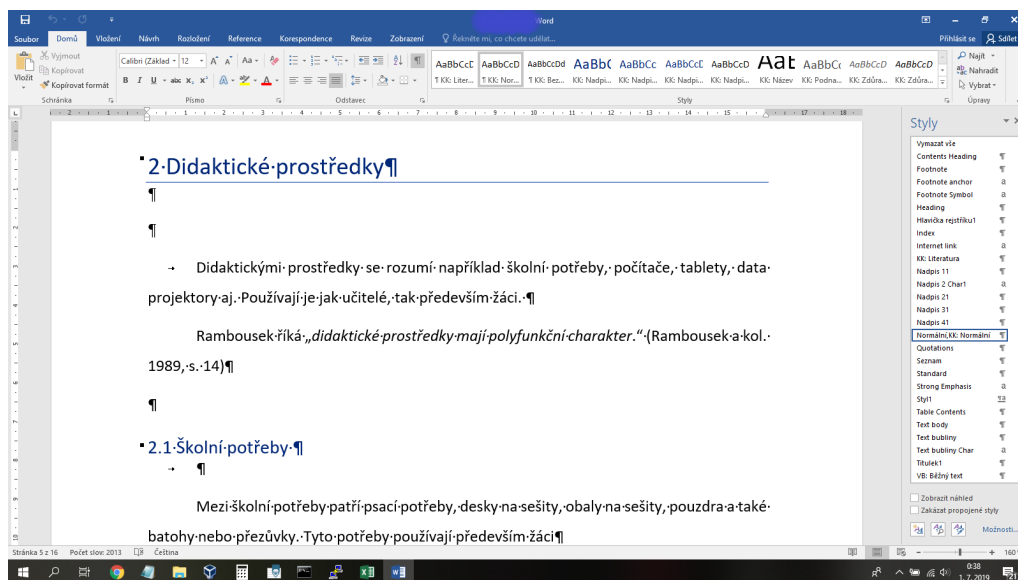
být libovolný text převzatý z internetu, z vlastní produkce nebo z jiné literatury. Samotné didaktické technice je v předmětu věnována čtyřhodinová úvodní část pouze v prezenční formě. U práce je však hodnocena výhradně formální část podle kritérií uvedených v následujícím seznamu:

1. zpracování hladkého textu,
2. typografický návrh dokumentu,
3. implementovaný systém formátovacích stylů,
4. stránková úprava,
5. tvorba automatického obsahu,
6. zpracování minimálně jednoho obrázku,
7. zpracování povinné tabulky (je individuálně zadána),
8. bibliografické citace a odkazy v textu.

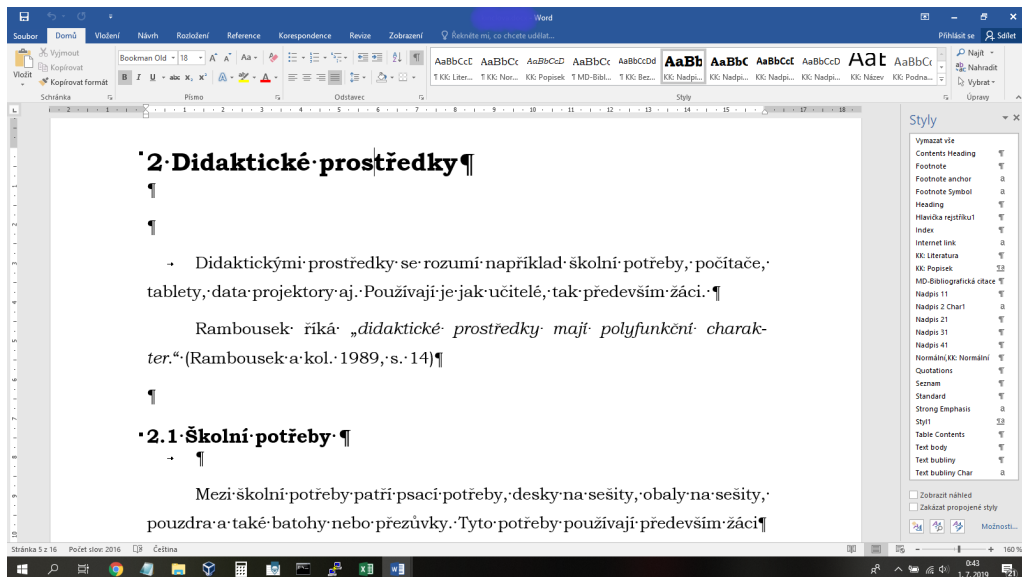
Za hodně důležitá jsou považována zejména tři kritéria – zpracování hladkého textu, systém formátovacích stylů a zpracování povinné tabulky. Pokud práce splňuje tato tři kritéria a z těch dalších alespoň určité minimum, je udělen zápočet. Neúspěšná práce je studentovi vrácena k přepracování, počet takových přepracování není dopředu limitován.

Pro ilustraci, jaké dokumenty někteří studenti produkují a jak probíhá hodnocení, si všimneme jedné z prací, která byla odevzdávána 5×.

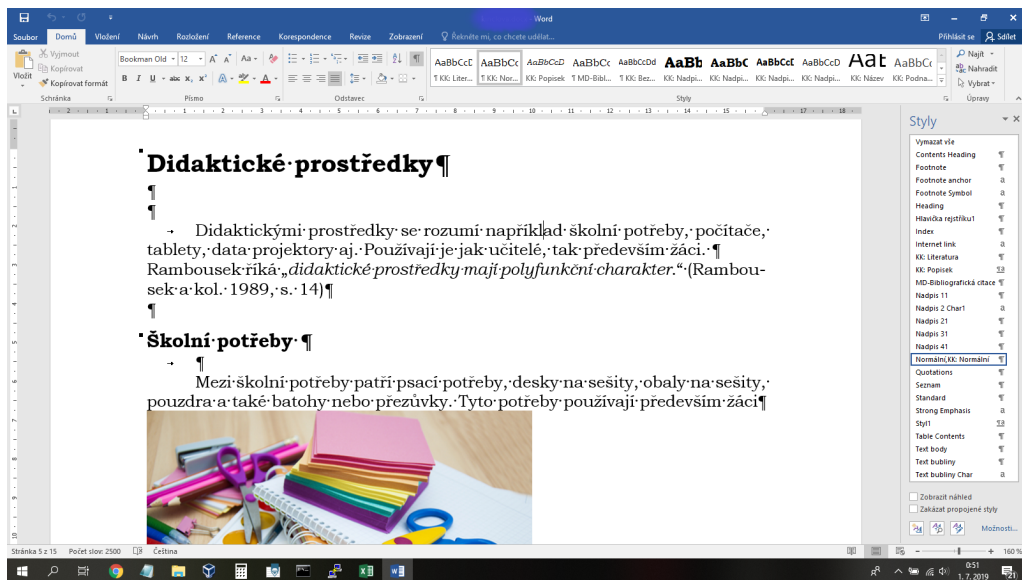
Pro srovnání uvedeme sejmuté obrazovky vzniklé z identického úseku práce ve čtyřech prvních verzích (pátá verze byla v tomto úseku shodná se čtvrtou verzí).



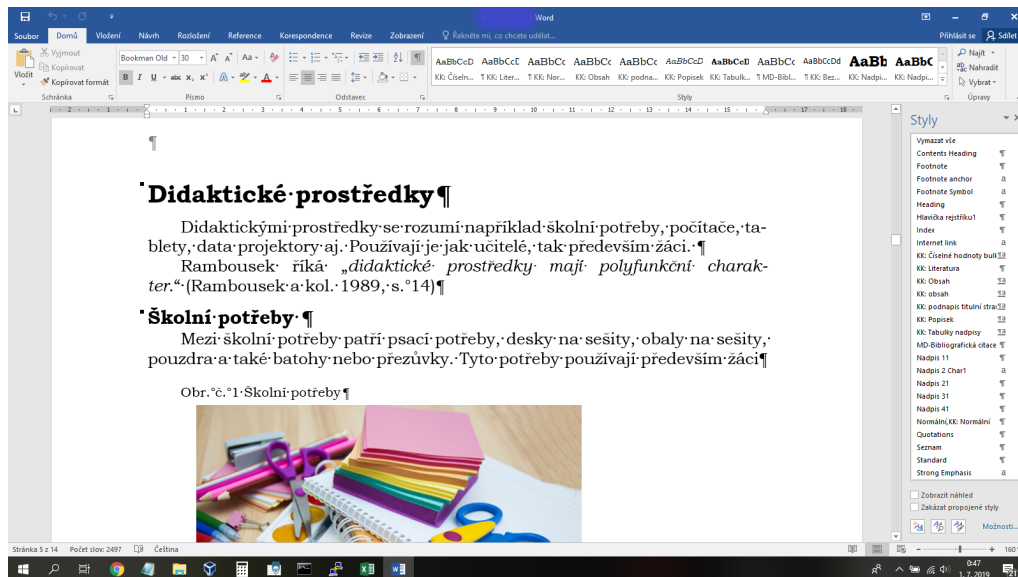
Obrázek 1. Výsek odevzdané práce – verze 1



Obrázek 2. Výsek odevzdané práce – verze 2



Obrázek 3. Výsek odevzdané práce – verze 3



Obrázek 4. Výšek odevzdané práce – verze 4 a 5

Na zmíněném úseku je možné v první verzi hodnotit:

- **zpracování hladkého textu:** je zapnuto zobrazování skrytých znaků, je tedy na první pohled vidět, že nejsou vloženy nezlomitelné mezery za jednoznakové předložky a spojky, za zkratku „s.“, že odstavcová zarážka je chybně řešena stiskem tabulátoru (ale nesystematicky – pouze ve dvou případech ze čtyř viditelných výskytů) a svislé mezery mezi odstavci jsou dosaženy opakovaným stiskem klávesy Enter.
- **typografický návrh dokumentu:** parametry odstavcové sazby jsou zjevně nevhodné (řádkování, zarážka), nadpisy nemají odpovídající koncepci (je převzata implicitní podoba vestavěných typograficky nevyhovujících stylů).
- **implementovaný systém formátovacích stylů:** v zobrazeném seznamu stylů v pravé části snímku je vidět, že základní text je řešen stylem „Normální“, nadpisy pak předdefinovanými styly „Nadpis 1“, „Nadpis 2“ atd., včetně vizuálních úprav vedoucích k parazitním stylům „Normální, KK: Normální“, „Nadpis 22 Char 1“ apod.

Je vidět, že autor dokumentu zcela ignoruje prakticky vše, co je obsahem předmětu. Dokument vykazuje typické znaky naprosto nepoučeného zpracování textu.

Podobně lze vyhodnotit druhou verzi: došlo ke změně písma, avšak zcela laickým technickým postupem, přičemž téměř všechny ostatní chyby zůstávají. V této

fázi se student stále ještě domnívá, že přístup ke zpracování dokumentů, který dosud zná a používá, je zcela správný. Dochází ke střetu s novou realitou.

Teprve ve třetí verzi začíná docházet k posunům – některé parametry sazby jsou upraveny, zbytek nedostatků ovšem stále přetrvává. Student se začíná seznamovat s obsahem předmětu, zatím je ovšem na začátku.

Až čtvrtá verze představuje využití vlastních stylů s potřebnými parametry sazby, byly použity (ovšem nedůsledně) nezlomitelné mezery, zmizely prázdné odstavce. Dokument obsahoval ještě řadu dalších nedostatků v jiných částech, byl tedy znovu vrácen a teprve v páté verzi byl přijat (s výhradami, protože nebyl upraven důsledně).

Student odevzdávající tuto práci pochází ze skupiny, jejíž výuka probíhala kombinovanou formou. Pro tuto skupinu je typické, že práci odevzdávají vícekrát, zatímco skupina studentů, která absoluuje prezenční výuku, dosáhne přijatelné podoby ve většině případů na první či druhý pokus. Není samozřejmě vůbec překvapivé, že větší hodinová dotace má na výsledky pozitivní vliv. Nejde však jen o samotný počet hodin ani o složitost řešeného problému. Studenti kombinované formy jsou většinou starší a jsou daleko více přesvědčeni o zcela vyhovující kvalitě dokumentů, se kterými se setkávají a které produkuje. Je tedy obtížnější je z tohoto mylného dojmu vyvést a přesvědčit je (pokud možno objektivními argumenty), že principy zpracování dokumentů jsou od základu jiné.

Předmět má pro studenty v kombinované formě tedy zvláštní charakter. Velmi často jej podceňují a při vypracování a opakovaném odevzdávání zápočtové práce teprve zjišťují, co všechno je potřebné zvládnout. Je logické, že zásadní změna počátečního očekávání relativně jednoduchého předmětu vede k určitému odporu a k negativnímu hodnocení ze strany studentů. Důležité však v tomto okamžiku vždy je, že získané znalosti a dovednosti jsou uplatnitelné při zpracování závěrečné práce.

Uvedený rozbor jednoho zápočtového dokumentu nemůže zastupovat podrobný výzkum výsledků výuky, ukazuje však na některé typické jevy.

2. Vliv dalších faktorů

2.1. Vedení fakulty

Snahou vedení fakulty by mělo bezesporu být zvyšování úrovně závěrečných prací. Zdaleka ne vždy je však v této snaze zahrnuta formální stránka prací.

Pokud jde o Provozní ekonomickou fakultu MENDELU, ta poskytuje určitý prostor zpracování textů ve výuce. Samotný předmět Bakalářský seminář byl původně vytvořen s cílem pomoci studentům nejen s procesem hledání vedoucích prací a celým harmonogramem zpracování bakalářské práce, ale také pomoci studentům při technickém řešení a způsobu psaní. Později došlo k různým přesunům, ale nikoliv ke zrušení nebo zmenšení tohoto prostoru.

Důležitým aspektem je rovněž kvalifikovaně vytvořená vyhláška děkana (Vyhláška děkana... , 2018), která zohledňuje aktuální doporučení a pravidla (pravidla pravopisu, ČSN 01 6910 apod.), a také péče o podporu studentů formou šablon a sázecích stylů.

Rovněž Institut celoživotního vzdělávání se nechal inspirovat uvedenými kroky, vytvořil prostor pro řešení formální stránky závěrečných prací ve zmíněném předmětu Moderní didaktická technika a disponuje šablonou pro závěrečné práce.

Vezmeme-li však další fakulty MENDELU a Fakultu stavení VUT, s nimiž má autor osobní zkušenosti, blíží se zde podpora formální stránky závěrečných prací nule.

2.2. Učitelé

Učitelé mají na úroveň závěrečných prací zcela zásadní vliv. Je přirozené, že ne každý musí umět všechno, velmi negativní však je, když student má širší rozhled než vedoucí jeho závěrečné práce, a navzájem spolu bojují o to, jak má závěrečná práce vypadat.

Širší povědomí o formální stránce dokumentů lze považovat za vážný problém, v němž musí sehrát opět důležitou úlohu vedení fakulty a alespoň formou podpůrných materiálů, vyhlášek a školení přispět k posunu učitelské kvalifikace v této oblasti.

3. Závěr

Pokud řešíme otázku, zda lze zvýšit kvalitu závěrečných prací, je odpověď kladná – zvýšit kvalitu **lze**, ale jak bylo postupně ukázáno, musíme si uvědomit skutečnosti soustředěné do následujícímu seznamu:

- **Potřebné úsilí** investované do výuky této oblasti je nemalé. Podle zkušeností z různých předmětů často hrozí podcenění probíraných prvků ze strany studentů a nutnost překonávat bezmyšlenkovitě přeжатé a všeobecně rozšířené nesprávné postupy.
- **Odpor ze strany studentů** vyplývá z nízkého všeobecného povědomí o základních pravidlech, principech a technických možnostech zpracování textů. Velmi negativně působí různá nekvalifikovaná doporučení, jimž se často přisuzuje závazný charakter.
- **Hodinová dotace** pro výuku alespoň základních principů zpracování dokumentů se pohybuje kolem 20 hodin přímé výuky.
- **Podpora ze strany fakulty (ústavu)** je nezbytná, a to zejména pro eliminaci chybných pravidel, vyhlášek nebo podobných dokumentů. Na úrovni vedení musí existovat alespoň základní povědomí o pravidlech (nejen jazykových) souvisejících se správně vytvořenými dokumenty a musí být otevřen prostor pro předměty, které studenty mohou vybavit potřebnými znalostmi a dovednostmi.

- **Technická podpora** může být vyjádřena existencí šablon nebo sázecích stylů usnadňujících technické řešení závěrečných prací a obsahujících základ, který zmenšuje množství chyb určitého typu v dokumentu.
- **Podpora ze strany učitelů** souvisí rovněž s určitým minimálním povědomím o zpracování bezchybných dokumentů, které se uplatňuje při vedení závěrečných prací, případně při kontrole semestrálních nebo jiných prací v průběhu studia v různých předmětech.

Pro přesnější a důkladnější argumentaci by bylo vhodné provést podrobnější výzkum. Nabízí se nová analýza závěrečných prací podle metodiky Devátové (2009), přičemž by bylo možné porovnat vybrané prvky a jejich vztah k prováděné výuce a porovnat stav s desetiletým odstupem. Rovněž lze detailněji vyhodnotit výsledky samotné výuky zpracování textů (Rybička, 2018). Kvantitativní argumenty mohou vést k optimálnímu rozvržení učiva v odpovídajících předmětech, k vytvoření lepších podpůrných materiálů a ke zvýšení povědomí o formální stránce dokumentů. Kvalita závěrečných prací si jistě takovou péči zaslouží.

Reference

- [1] BORŮVKOVÁ, J. *Jak napsat bakalářskou práci*. Brno: FRRMS MENDELU, 2010.
- [2] ČSN 01 6910. *Úprava dokumentů zpracovaných textovými procesory*. Praha: Ústav pro normalizaci a měření, 2014.
- [3] DEVÁTOVÁ, M. *Metody hodnocení kvality závěrečných prací*. Diplomová práce. Brno: PEF MENDELU, 2009.
- [4] RYBIČKA, J. Výsledky výuky zpracování textů. Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2018 a 2019, Žilina, 2.–4. júla 2018, ISBN 978-80-554-1627-4.
- [5] *Vyhlasška děkana 4/2018, o diplomových a bakalářských pracích* [online]. [cit. 7. 6. 2019] Dostupné na https://www.pef.mendelu.cz/wcd/web-pef/de/vyhlaska_1804_zp.pdf.

Kontaktní adresa

doc. Ing. Jiří Rybička, Dr., Ústav informatiky, Provozně ekonomická fakulta, Mendelova univerzita v Brně, Zemědělská 1, 613 00 Brno, Česká republika, *Aktuální adresa*: SOIT, 010 01 Žilina, Slovensko,
E-mailová adresa: rybicka@mendelu.cz, <http://akela.mendelu.cz/~rybicka>

AKO PÍSAŤ DOBRÉ BASH SKRIPTY

Miroslav Biñas

Znalosť príkazového riadku a tvorby bash skriptov dnes patrí k základným znalostiam každého DevOps-áka. Ak si však pri ich písaní nedáte pozor, môže sa ich písanie a ďalšia údržba stať veľmi rýchlo peklom. V príspevku vám preto predstavím niekoľko odporúčaní, vďaka ktorým už zlý skript nenapíše. Maximálne nefunkčný.

ČO VŠETKO SOM SA PRI PROGRAMOVANÍ ARDUINA NAUČIL ZLE

Miroslav Biñas

Arduino je otvorená platforma, ktorá sa s veľkým úspechom používa vo výučbovom procese. Najčastejšie sa používa vo výučbe základov elektroniky alebo v súčasnosti populárneho IoT, ale nezriedka sa dá stretnúť aj s využitím Arduina vo výučbe základov programovania a algoritmickej. A keď nie priamo vo výučbe, tak sa dá s týmto prístupom stretnúť napríklad v niektorých knihách alebo v seriáloch v IT časopisoch. V tomto príspevku vám ukážem, čo všetko svojimi rozhodnutiami ovplyvnili autori Arduina a vďaka čomu sme ako študenti získali niektoré zlé návyky.

NIM – LEPŠÍ PYTHON?

Michal Kaučič

To bude na dlhé presviedčanie.

ARDUINO AKO KONTROLÉR PRE ALFANUMERICKÉ DISPLEJE A INÉ SVETIELKA

Tomáš Majer

V rýchlosti predstavím a predvediem ako pripojiť k Arduinu rôzne typy displejov a iných vizuálnych prvkov, napríklad LED pásikov (tzv. NeoPixel).

TERMINOLÓGIA V INFORMAČNÝCH TECHNOLOGIÁCH, INFORMATIKE A INFORMATIZÁCII

Martin Šechný

Vývoj v informačných technológiách je rýchly a rovnako rýchlo sa mení aj používaná terminológia. Odborný žargón sa tvorí a používa v relatívne malej skupine IT špecialistov, ale prenikanie odborného jazyka do súčasného (hovorového) jazyka zasahuje celú spoločnosť. Informačné technológie majú blízky vzťah s informatikou, čo je vedný odbor aj školský predmet. Terminológia vedného odboru a školského predmetu by mala byť precízne formovaná, lebo bude používaná danou generáciou po celý život. Informatizácia je dlhodobý proces zavádzania informačných technológií do praxe, obyčajne na základe legislatívy, v ktorej je terminológia zásadná. Ako vyzerá súčasná terminológia v informačných technológiách, informatike a informatizácii na Slovensku? Ako možno prispieť k zlepšeniu? Vysvetlím, prečo by mala byť práca na terminológii otvorená odbornej aj laickej verejnosti, najlepšie prostredníctvom otvorených IT riešení. Predstavím niekoľko návrhov, ako prepojiť tvorbu terminologickej databázy s inými jazykovými a encyklopedickými zdrojmi a s použitím terminológie v praxi, školstve, legislatíve SR, legislatíve EÚ, otvorených dátach, automatizovanom preklade.

OUTPUT RUTINA T_EXU A JEJÍ POUŽITÍ

Jan Šustek

V přednášce projdeme vnitřností programu TeX a ukážeme si, jak se jednotlivé znaky vstupního souboru *.tex postupně dostanou až do výstupního souboru *.pdf.

Cestou se zdržíme u algoritmu řádkového zlomu, který bez debat výrazně předběhl svou dobu. Vhodnou kombinací jeho parametrů lze nastavit nejen zarovnání textu do bloku, na střed nebo na praporek, ale ukážeme si i mnoho dalších možných způsobů zarovnání textu.

Na konci naší cesty se nachází output rutina, která má za úkol umístit vysázený text na stránku. Ukážeme si, jak lze nastavit různá záhlaví a zápatí a jak lze jednoduše udělat hlavičkový papír. Také si ukážeme různé praktické aplikace output rutiny. Přijde řeč i na problematiku zjišťování pozice konkrétního bodu sazby, což s output rutinou úzce souvisí.

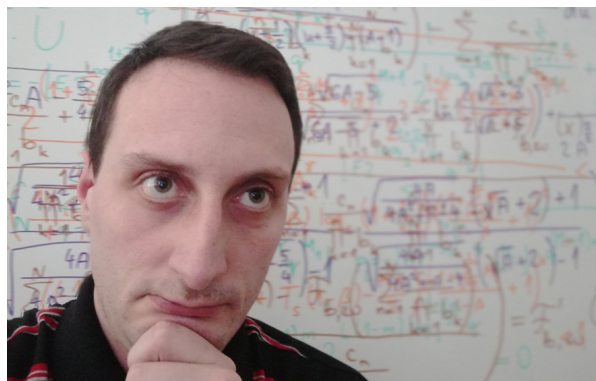
Bývá zvykem domácí úkoly dávat až během přednášky. Posluchači však dostanou domácí úkol již nyní. Vaším úkolem je najít nějaký zajímavý způsob zarovnání odstavce. Na přednášce zkusíme, zdali je možné tento způsob zarovnání nastavit v TeXu. Nejzajímavější neumělý způsob zarovnání bude oceněn.

GENEROVÁNÍ DOKUMENTOVANÉHO ZDROJOVÉHO SOUBORU PO BLOCÍCH

(Vyžiadaná prednáška)

Jan Šustek

V prednáške se zaměřím na dokumentované programování – kterak z jednoho vstupního souboru vytvořit zdrojový soubor programu a zároveň dokumentaci k němu. Na začátku ukážu, jak je možné snadnou úpravou maker OPmacu vytvořit makra pro dokumentované programování v TeXu. Dále ukážu, jak je možné generovat zdrojový soubor po blocích, které se pak vloží na jiné místo v souboru. Bloky je možné budovat po částech a je možné je do sebe libovolně vnořovat. To umožňuje autorovi vytvářet zdrojový soubor podle logických souvislostí a ne nutně řádek po řádku. To je pak také srozumitelnější pro čtenáře. Pomocí háčků pak ukážu, jak je možné v takto vygenerovaném souboru používat křížové odkazy. V závěru ukážu využití při generování zdrojového souboru v jazyce Basic, kde se to křížovými odkazy jenom hemží. Pomocí TeXu vytvořím program pro počítač Commodore 64 a jako bonus ukážu, jak je možné na tomto počítači použít české TeXové fonty.



Jan Šustek vyštudoval matematiku na Ostravské univerzitě, kde tiež v súčasnej dobe pracuje. Jeho odborným zameraním je teória čísel. Je organizátorom mnohých matematických konferencií a súťaží. S \TeX -om sa zoznámil v roku 2002. Je členom CSTUG, kde bol v rokoch 2010–2016 vo výbore a teraz je na pozícii šéfredaktora Zpravodaje. S obľubou používa \TeX a Metapost

na všetko možné a ešte radšej na všetko nemožné. Vo voľnom čase sa rád venuje cyklistike.

ZA OPONOU ÚSPEŠNÉHO OPEN SOURCE GIS PROJEKTU (Vyžiadaná prednáška)

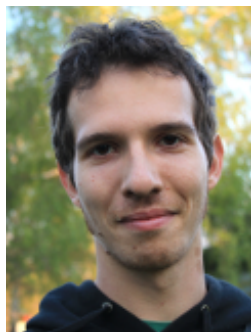
Martin Dobiaš

Bežný používateľ open source softvéru obvykle nesleduje dianie okolo projektu a často jediný kontakt s projektom je návšteva webovej stránky pre stiahnutie novej verzie. Pritom za každým projektom existuje väčšia či menšia komunita ľudí, ktorá zabezpečuje jeho chod a neustále posúva projekt ďalej. Nie je pritom väčšinou zrejmé, ako sa taká komunita sama organizuje a čím je motivovaná.

Ešte stále vo firmách aj vo verejnej správe pretrvávajú mýty o tom, že open source softvér je dobrý možno pre nadšencov, ale nie pre profesionálne nasadenie, že open source softvér nedokáže nahradiť komerčný softvér z hľadiska kvality, množstva podporovaných funkcií či kvôli chýbajúcej podpore.

V prednáške sa preto pozrieme do zákulisia projektu QGIS a skúsime si odpovedať na otázky ako:

- Kto tento softvér vytvára a prečo?
- Bude projekt naďalej existovať aj za päť rokov?
- Mám sa na koho obrátiť v prípade problémov?
- Môžem aj ja ovplyvniť smerovanie projektu?



Martin Dobiaš vyštudoval informatiku na matfyzě University Karlovy v Prahe a momentálne pracuje v anglickej firme Lutra Consulting, kde sa venuje predovšetkým vývoju Open Source nástrojov na prácu s geografickými dátami. Od roku 2005 aktívne prispieva do projektu QGIS. Keď nesedí pri počítači, tak rád cestuje do neznámych končín a športuje.

JAK JSEM UČIL FRAKTÁLY A GRAFY A VZORCE A TEXTY LÉTAT

(Vyžiadaná prednáška)

Pavel Stříž

Motto přednášky:

„... a ať mě matematici opravují ... a ať mě programátoři opravují ... a ať mě grafici opravují ... a ať mě ajťáci opravují ... a ať mě TeXisti opravují ...“

Dívám se před sebe na primitivní úkol pro ochotníky na divadle, a to vytvořit animaci do scény na televizi. Stačilo by přibližně pár vteřin stahování z YouTube přes youtube-dl v podstatě čehokoliv, vyhodit (ffmpeg s parametrem `-an`) či vypnout zvuk (mpv s parametrem `-mute`) a bylo by dílo hotové, ale ne, já zas musím jít po svých.

Došlo na nebožtíka Norberta Wienera, který, chudák, na řešení svých zpětno-vazebních problémů musel také sáhnout po více oborech. Já jsem možná nemusel, ale to by mohlo být téma do kuolárů konference: Jak na to jinak...

Představím vám nyní vizi, kterou mám v hlavě, a na konferenci budete moci vidět výsledek a vyslechnout si, kam jsem se až dostal.

- První vrstva. Animaci si představuji tak, že v pozadí poběží nějaké fraktály, ale nějak tak, aby nevypadaly jak přečizené z Microsoft PowerPointu, vytažené z vizualizací Microsoft Media Playeru či právě uvařené v Adobe Premiere. Nějak jinak.
- Druhá vrstva. Nad tím by mohlo být něco kulatého a rotujícího. Mars, Venuše či něco takového GISáckého. Ve smyslu loga či živých televizních přenosů s informačními panely.
- Třetí vrstva. Hluboce jsem se zadíval na bílou zeď a černý terminál a říkal jsem si, že by tu a tam mohl proběhnout graf či vzoreček. Něco, však víte, něco, co člověk často v TeXu nesází. Nainstalovat si poslední TeXLive, vyzkoušet funkčnost méně známých věcí ap.
- A poslední vrstva, kvůli které se to dělá, trochu toho textu. Na přání autora divadelní hry fráze: Móda live!

Pokusím se volně naznačit řešení, kde jsem se učil nové věci a víceméně záměrně experimentoval. Na fraktály jsem použil balík `flam3`. Od vytvoření jednoho plamenného fraktálu po jejich přechody. Následovaly mé pokusy o spojení několikavteřinových videí z výpočetního prostředí Electric Sheep, a to tak, aby přechody mezi fraktály byly spojitě, spisovně řečeno bezešvé. Narazil sem na NP-úplný problém hledání nejdelší cesty v neorientovaném grafu s cykly. Nějak jsem s tím zabojoval (Bash, Python, Lua).

Další vrstvy byl pro mne oříšek: jak vložit do existujícího videa objekty, a to tak, aby bylo zachováno průhledné pozadí. Zmíním své marné pokusy ve flowblade

a kdenlive. Poněvadž jsem víc zápasil s kvalitou videa než s problémem samotným, přešel jsem na ImageMagick a výpočetně nepohodlné vrstvení rastrových obrázků. Ale proč si to nezkusit, říkal jsem si.

Na vzorce a grafy jsem náhodně zvolil balíčky z TeXLive 2019, se kterými často nepracuji, trochu s černobílou vizí šílenství filmu *Pí* (1998). Ale ne zas graficky přehnaně moc, jako je vidět u začátků současných seriálů typu *Letecké katastrofy* (2003), *Vteřiny před katastrofou* (2004), *Humans* (2015), nedejbož čínského velko filmu *The Wandering Earth* (2019). Přišlo mi totiž zajímavé si výstupy vložit do programu jWildfire a zkusit si co místní matematické transformace umí.

Poněvadž celou dobu hovořím o vrstvách, je to trochu záměrně. Chtěl jsem, aby vizuálně něco bylo neustále v pozadí a něco předtím. Zaujal mě však jiný problém, jestli jsem schopn takto uměle vydefinované vrstvy zapínat a vypínat ve videu. Napadlo mě, že by každá vrstva mohla a nemusela být vidět (binární situace), při čtyřech vrstvách to je 2^4 možností. Nápad na řešení této šílenosti vzešel, když se člověk dívá na filmy a je tam další video/audio stopa, kterou si lze přepnout, podobně jako to považujeme za samozřejmé u titulků. Nešlo by tedy těchto 16 videí natlačit do jednoho? Není to situace ideální a pohodlná, ale co vše k tomu bude potřeba? Užít jedno video s 16 videostopami nebo 16 videí s jednou videostopou? Nebo na to jít úplně jinak? *Babylon.js* či *Three.js*? Blender s Pythonem, *EEVEE*, *Animation Nodes*, *Sverchokem* a *Text FX*. . . ? Asi skončím v 3D tvorbě jako jútúbák *Midge Sinnaeve* s jeho seriálem *Weird Sh!t*. . .

S těmito myšlenkami a otázkami za vámi do Žiliny zavítám. A pokud nedopadnu jako 3D vytištění představitelů seriálu *Westworld* (2016), tak se o zkušenosti a slepé uličky rád podělím.



Pavel Stríž už od útlého věku pořád něco s tím PéCéčkem zkouší. Tatick programátor a technik a máma umělecky nadaná své potomky k tomu nenásilně vedli. Nu, spíš ho rodiče odrazovali, ať u toho pořád nesedí. Někdy nad tou technikou vyhrává, jindy nikoliv. Dlouhé roky byl spokojen na *Microsoft Windows* se *CygWinem*, ale s příchodem *Microsoft Windows 10* přešel na *Linux* a občas ve virtuálním stroji si nostalgicky vzpomene na ty retro roky komerčního uzavřeného kódu.



S príchodom tretieho tisíciletia začal pokusy v $\text{T}_{\text{E}}\text{X}$ u a $\text{M}_{\text{E}}\text{T}_{\text{A}}\text{P}_{\text{O}}\text{S}\text{T}$ u. Pořád mu to nejde, ale ten boj nevzdáva. S príchodom Apache bojuje s PHP a MySQL, od svých začátků na OSSConf bojuje v Pythonu, Blenderu, GIS a s OpenData, baví ho boje s balíčky PSTricks, TikZ a PGFplots, s príchodem Lua $\text{T}_{\text{E}}\text{X}$ u bojuje v Lua, nu, a teď mu někdo na výpočetní stroj nainstaloval NodeJS, Nginx a takové. On to však nebyl, on za nic nemůže, to byl někdo jiný nebo to samo vlivem gravitace! A poté, co rozchodil wakeonlan/etherwake, tak si nemůže rozespálý při noční cestě na toaletu ani na chodbě rozsvítit, aby ta technika nebyla hned připravená k práci... Zrádná to čidla, Arduino i RaspberryPi! O OpenHardware mu ani nemluvte, buď ho ta elektřina zabije, a nebo z něj opravdu i něco užitečného vypadne.

OSS A JEHO VYUŽITIE PRI TVORBE CORINE LAND COVER 2018

Luboš Balážovič, Jakub Fuska

Cieľom príspevku je priblížiť využitie open source nástrojov v rôznych fázach prípravy slovenskej časti projektu mapovania európskej krajinej pokrývky Corine Land Cover. Prípravné práce sa realizovali prostredníctvom skriptov v programovacom jazyku Python a skriptovacom jazyku BASH využívajúcich knižnice GDAL/OGR. Práce, kde bol potrebný aktívny zásah užívateľa, sa realizovali s využitím softvéru QGIS. Tieto nástroje významne pomohli pri konečnej fáze kontroly, v ktorej bolo možné automatizovane identifikovať veľké množstvo aj starších chýb. Časť metodiky tak bola realizovaná v prostredí open source s výnimkou častí kde bolo potrebné využiť predpísaný ArcGIS modul, alebo špecializovaný softvér. Išlo o pilotné využitie tohto softvéru pri realizácii CLC na Slovensku.

LEPŠIA KARTOGRAFIA S QGISOM: PRÍPADOVÁ ŠTÚDIA GLOBAL SOLAR ATLAS

Juraj Beták, Šimon Štassel

Viac ako desaťročie poskytuje spoločnosť Solargis globálne meteorologické, klimatické a iné geodáta pre podporu solárnej energetiky. Jadrom našej práce je tvorba a operačná prevádzka rozsiahlej časopriestorovej databázy. Mapy sú vynikajúcim prostriedkom, ako takéto dáta vizualizovať, vysvetľovať rôzne fenomény, poskytnúť argumenty pre decíznu sféru, či podporiť edukačný segment v témach efektívnosti solárnej energetiky. V roku 2016 sme začali ambiciózný projekt, ktorý je v súčasnosti prezentovaný na doméne GlobalSolarAtlas.info. Pre sekciu Download sme pripravili a publikovali stovky regionálnych a štátnych tematických máp. Pre náš tím to bola technologická výzva. Na začiatku projektu sme po starostlivom vyhodnotení opustili komerčnú GIS platformu, vybrali sme QGIS a transformovali našu kartografickú produkciu. V príspevku sa chceme podeliť o našu 3-ročnú skúsenosť s QGISom, so zameraním na kvalitu kartografického výstupu, niektoré triky, ktoré máme radi, či možnosti automatického generovania máp:

- tematické mapy z rastrových vstupov;
- impakt farebnej legendy pre dobrú čitateľnosť a pochopenie mapy;
- korektná kombinácia dátovej vrstvy s tieňovaným reliéfom bez skreslenia tematickej farebnej legendy;
- triky v štýlovaní, využitie transparentie objektov;
- mapové popisky: automatizované versus manuálne umiestnenie;
- prepojenie s PyQGIS a limity automatickej tvorby máp; manuálne korekcie a úpravy;
- dizajn mapového layoutu: v jednoduchosti je krása (skúsenosti na základe konfrontácie s našimi koncovými užívateľmi);
- benefity Open source platformy pre náš tím a pre našich koncových užívateľov.

VÝUČBA PROGRAMOVANIA TVORBOU POČÍTAČOVÝCH HIER

Miroslav Biňas

Učiť sa programovať je ťažké. Učiť programovať niekoho iného je však ešte ťažšie. Čo tak však zvýšiť motiváciu študentov tým, že ich budeme učiť programovať spôsobom, kedy budú vytvárať vlastné počítačové hry? V tomto príspevku vám predstavím knižnicu Pygame Zero, ktorá je na tento účel ako stvorená. Tá umožňuje vytvárať hry v jazyku Python, ktorý sa stáva dominantným aj na Slovensku v prostredí stredných škôl. Predstavím vám teda nástroj, pomocou ktorého dokážete u študentov podporiť kreativitu. A nie len u nich.

AUTORSKÉ NÁSTROJE V JAZYKU MARKDOWN

Miroslav Biñas

Markdown je jednoduchý značkovací jazyk, ktorý bol vytvorený s cieľom umožniť používateľom písať dokumenty vo formáte, ktorý je jednoduchý na čítanie aj písanie s možnosťou ďalšieho exportu do jazyka HTML. Dnes to už nie je len HTML, ale z Markdown-u je možné exportovať dokumenty napr. aj do LaTeX-u, PDF alebo vytvárať knihy vo formáte ePub. V tomto príspevku vám predstavím nástroj Gitbook na tvorbu kompletných knižných publikácií a statický generátor stránok Jekyll, ktorý stránky umožňuje písať práve v tomto formáte.

ROBÍME IOT S ESP32 V JAZYKU MICROPYTHON

Miroslav Biñas

Arduino je v súčasnosti asi najznámejšia prototypovacia doska, ktorá sa používa aj vo výučbe či už na výučbu programovania alebo sa pomocou nej študenti zoznamujú so základmi elektroniky. Jedno z nevýhod tejto dosky je však absencia komunikačného modulu, pomocou ktorého by bolo možné Arduino pripojiť do internetu. Dokúpením vhodných shield-ov sa však cena takéhoto riešenia môže aj zdvojnásobiť. V rámci príspevku sa vám pokúsim predstaviť dosku ESP32, ktorá odstraňuje uvedené problémy a pridáva ďalšiu výhodu, ktorou je možnosť programovať riadenie v jazyku MicroPython.

GISQUICK: RYCHLÉ SDÍLENÍ MAP NA OPEN SOURCE ZÁKLADĚ

Jáchym Čepický

Gisquick je Open Source nástroj pro webovou prezentaci geografických dat. Při jeho tvorbě náš slovensko-český tým zúročil dlouholeté zkušenosti při tvorbě webových mapových aplikací pro obce a zkušenosti s organizací vývoje otevřeného software a při nasazování těchto aplikací do produkčního prostředí. Gisquick umožňuje publikaci mapového projektu z prostředí QGIS na web – doslova na několik kliknutí. Jelikož je na straně serveru využit mapový server QGIS, jsou data na webu vykreslena přesně tak, jak byla nastýlována v desktopovém prostředí. Součástí webové aplikace je pak vyhledávání prvků pomocí filtrů, měření, navigace, identifikace objektů a tisk mapy do PDF. Celé řešení je navrženo tak, aby nedošlo k přílišnému zatížení serveru a prohlížeče – data jsou ukládána do vyrovnávací paměti, dlaždicována, atd. Serverová část aplikace je pro jeho rychlé nasazení v domácí infrastruktuře zabalena do izolovaného kontejneru Docker. Celý projekt má tři části: zásuvný modul pro publikaci projektu v prostředí

QGIS-Desktop, serverová část aplikace na bázi QGIS-Server, webového frameworku Django. Prezentáční klientská část v jazyce JavaScript využívá OpenLayers a další knihovny pro tvorbu mapových aplikací. Komunikace mezi serverem a klientem je postavena na standardech OGC WMS, WMTS a WFS. Myslíme si, že Gisquick může být zajímavou alternativou k velkým proprietárním řešením pro veřejnou správu a samosprávu. Provozování platformy Gisquick je možné jak na vlastním hardware, tak na hostované infrastruktuře na adrese <http://gisquick.org>.

OPENGEOLABS: TECHNICKÁ PODPORA PRO OTEVŘENÝ SOFTWARE PRO GIS

Jáchym Čepický

Otevřený software nemá jednoho dodavatele, obvykle neexistuje jedno kontaktní místo pro uživatele software. Problém lze vyřešit na mnoha úrovních nebo prostřednictvím různých programátorských týmů. OpenGeoLabs s.r.o. nabízí placenou profesionální podporu všem, kteří takovou službu potřebují. Členové našeho týmu jsou součástí otevřených komunit, víme kde a jak se zeptat za vás. Umíme přizpůsobit zdrojový kód některých používaných programů nebo máme kontakty na vývojáře, kteří to umějí lépe. Ve světě otevřeného software pro GIS se pohybujeme od roku 2002, jsme jeho součástí a pomáháme ho budovat.

3D MAPY V QGIS

Martin Dobiaš

Vo verzii 3.0 pribudla v QGIS možnosť zobrazovať dáta v 3D pohľadoch a ďalšie verzie prinášajú ďalšie vylepšenia. Otvárajú sa tým úplne nové možnosti vizualizácie geografických dát v tomto nástroji. V prezentácii sa zameriame na schopnosti QGIS v tomto smere. Priblížime si, ako sa 3D zobrazenie ovláda, aké vstupné vektorové či rastrové dáta sa dajú použiť na vytvorenie 3D máp a aké sú momentálne možnosti nastavenia štýlu dát pre 3D zobrazenie. Ukážeme si tiež niektoré doplnkové nástroje na prácu s 3D dátami. Záverom sa pozrieme na to, aké ďalšie možnosti sú plánované do ďalších verzií.

PRIEMYSELNÉ ZARIADENIA ZALOŽENÉ NA ARDUINO A RASPBERRY PI. OSLOBODENIE PRIEMYSLU POMOCOU OSS

Richard Fabo

Prednáška pojednáva o rozdieloch medzi slobodným a neslobodným operačným systémom / softvérom, s prenesením tejto analógie do oblasti programovania

PLC. Krátko uvádza z toho vyplývajúce obmedzenia a rastúcu finančnú náročnosť pri obstarávaní softvérového vybavenia pre neslobodné zariadenia. V ďalšom hovorí o zariadeniach, ktoré, minimálne z pohľadu ich programovania, využívajú otvorené a slobodné vývojové prostredia so silným komunitným pozadím, pričom všetky ako riadiaci člen používajú procesory Atmega (hoci aj nie zákonite vývojové dosky Arduino).

Spomenutie najrozšírenejších zariadení je východiskom pre bližšie predstavenie PLC automatov (založených na Arduino Leonardo a Mega) od firmy Industrial Shields, u ktorých sa krátko vymenuje portfólio a najdôležitejšie vlastnosti. Ako zobrazovacie zariadenia budú predstavené dotykové panely založené na Raspberry Pi (resp. Banana Pi alebo TinkerBoard). Uvedú sa najbežnejšie možnosti (a vývojové prostredia) ich programovania, s dôrazom na najdôležitejšiu výhodu – jednoduché rozšírenie s využitím komunitou vytváraných knižníc.

NÁHODNÉ GENEROVÁNÍ PÍSEMEK Z DATABÁZÍ

Marian Genčev

Autor prednášky predstaví užití vlastních maker pro náhodné generování zkouškových písemek pomocí databází, v nichž jsou zapsány jednotlivé úlohy a případně jejich řešení. Důraz při užití maker koncovým uživatelem je kladen na jednoduchost a přehlednost. Systém maker poskytuje možnosti voleb pro vypnutí/zapnutí náhodného generování, resp. zobrazení řešení. Rovněž je podporována možnost generování dvou jazykových mutací z jedné databáze, kterou lze uplatnit v případě výuky studentů v cizím jazyce.

VYUŽITIE QGIS PRE SPRACOVANIE DÁT MIERY OHROZENIA ŽELEZNICE

Michal Kučera

Článok popisuje metodiku vyhodnotenia ohrozenia železničnej infraštruktúry príslahou vegetáciou.

ZOBRAZOVANIE METEOROLOGICKÝCH DÁT V QGIS 3.6

Peter Petrík

Väčšina objektov z praxe môže byť reprezentovaných v rastrovej alebo vektorovej vrstve. Avšak dáta z oceánografie, meteorológie alebo hydrológie sú zložené z viacerých parametrov (vietor, teplota, tlak), reprezentovaných na nepravidelnej štruktúrovanej sieti vo viacerých časoch (napríklad každú hodinu). Sieťová vrstva je nový typ QGISovej vrstvy, ktorá je práve navrhnutá pre jednoduché zaobchádzanie s týmito dátami. QGIS používa novú knižnicu MDAL, licencovanú

pod MIT, pre načítavanie týchto dát. V prezentácii použijeme rozšírenie QG-ribDownloader na stiahnutie verejne dostupných meteorologických dát. V QGIS 3.6 ich naštýľujeme, zobrazíme a pozrieme sa na rozdielne možnosti prehliadanie a analýzy. S rozšírením Crayfish sa pokúsime pracovať s interaktívnymi grafmi a s exportom animácii.

POUŽITIE MOBILNEJ APLIKÁCIE INPUT NA ZBER DÁT

Peter Petrík

Zber dát je podstatná časť všetkých užívateľov GIS systémov. Input, nová bezplatná open source mobilná aplikácia, bola vytvorená vďaka novovzniknutej knižnici QGIS Quick. S aplikáciou Input sa dajú jednoducho zaznamenať geoinformácie na tablete alebo mobilnom zariadení. V prezentácii sa pokúsime ukázať, ako sa táto nová aplikácia dá použiť pre zaznamenanie pár objektov. Najprv nastavíme jednoduchý projekt v desktopovej verzii QGISu. Potom s pomocou webovej služby Mergin nahráme projekt na mobil a zaznamenáme objekty. So službou Mergin znovu presunieme upravené vektorové vrstvy na desktopový QGIS. Počas prednášky si ukážeme aj ďalšie ovládacie prvky Inputu.

POUŽITIE FOSS GIS PRI TVORBE LETECKÝCH SYSTÉMOV A LETECKÝCH MÁP

Peter Piesecký, Andrea Wongreyová

FOSS GIS je veľmi rozvinutý a ponúka veľa možností uplatnenia nie len v univerzitnom prostredí a malých projektoch, ale aj v kritických systémoch pracujúcich v reálnom čase a veľkých kartografických projektoch. Knižnice ako GDAL, PROJ, GEOS a S2 umožňujú odbremeniť programátorov od nutnosti tvorby vlastnej nízko úrovňovej GIS logiky a poskytujú všetko nutné nato, aby sa programátori mohli sústrediť na hlavnú logiku ich konkrétneho riešenia. Nástroje ako QGIS, GRASS, PostGIS, GeoServer a mnohé iné pokrývajú celé spektrum práce GIS analytika a kartografa. Od digitalizácie zozbieraných údajov, cez tvorbu a spravovanie neobmedzene komplexných údajových modelov, analýzu nad týmito modelmi až po ich publikáciu na Internet-e v surovej podobe, alebo v podobe vysoko kvalitných kartografických výstupov. V prezentácii sa pokúsime priblížiť ako používame FOSS GIS na tvorbu kritických systémov pre letectvo a na tvorbu leteckých máp.

GEOWEB A GEOSERVER

Jan Růžička

Platforma GeoWeb nabízí řadu možností jak dnes vyvíjet geografické informační systémy, které pracují s aktuálními daty, umožňují sdílet znalosti a využívat výpočetní sílu technologií ze skupiny cloud computing. Jedním ze stavebních kamenů této platformy jsou služby standardizované OGC (Open Geospatial Consortium). GeoServer patří mezi hlavní hráče v oblasti implementace nových standardů nebo nových verzí existujících standardů. Pokud máte zájem se seznámit s tím co standardy nabízejí, je GeoServer volbou velmi vhodnou. V rámci příspěvku se podíváme na platformu GeoWeb, včetně využití služeb na bázi cloud computing a ukážeme si jakou roli zde hraje GeoServer. Dále si představíme GeoServer a ukážeme si jeho základní konfiguraci. V rámci diskuse se může evěnovat otázkám pokročilé konfigurace a napojení do infrastruktury.

MALÉ KRŮČKY V CONTEXTU ANEB PŘÍPRAVA VELKÉ PŘEDNÁŠKY A MLUVENÍ O KALENDÁŘI

Pavel Stríž

Mám toho na TeXovém srdci víc a musí to ze mne ven. . . Chci příchozím představit některé (hlavně české) TeXové osobnosti a připomenout současný vývoj v TeXovém světě. Poněvadž mi už kape na maják, řekl jsem si, že je čas vyzkoušet ConTeXt Minimals a že slajdíky by mohly být to pravé ořechové na sazbu, relativně nenáročnou ve fázi studia nového prostředí. Když by mi to šlo, bylo by to bezvadné, když ne, mám čas přednášku napsat v něčem jiném. A poněvadž v září 2019 bude představen LuaMetaTeX 2.0 (LMTX), tak jsem si ho rovnou nainstaloval také. Zkusíme si v ConTeXtu "Hello World!", nainstalujeme nový modul a vyzkoušíme si místní filtr. Ukážu, jak snadné je zahodit LaTeX(3) s celým TeXLivem, a když už jsem byl u toho, tak jsem zahodil i můj oblíbený TeXworks a nastavil jsem si editor SciTE s náhledem PDF v programu XpdfReader. Podíváme se na trendy u variabilních a barevných písem. Zavzpomínám krátce na kaligrafii, Manfreda Kleina přes Edwarda R. Tufteho až po vize Hanse Hageny a Taca Hoekwatera. Zamyslím se nad sazbou uvnitř kresby znaků (úzce spjaté s wordcloudy) i nad málo používanou sazbou mimo kresbu znaků se zajištěním průhlednosti. Hra na černou kočku a bílou myš se zakrývající barvou skončila! Je čas renesance Metafontu v podobě MFLua a na oprášení FontForge? Když už jsem "čistil" disk, tak jsem hodil do koše i TikZ a zacílil se na Metapost. Poněvadž je konference v duchu 3D, organizátoři předají dárek či dají alespoň nahlédnout na dvouletý plánovací kalendář s decentním prostorovým efektem. Zamyslím se jaké další 3D efekty by šly vymyslet a jak (ne)výhodně na ně pravděpodobně jít. Příprava kalendářů, když už na nic jiného, je výborné duševní cvičení a ohlednutí se vždy za uplynulým školním/ akademickým/ pracovním/ sazebním rokem. Nu,

a jdu si obnovit programy z Koše, třeba se budou ještě hodit... Sakra! Já si ten Koš už vysypal...

OTVORENÉ IT NÁSTROJE VO VYUČOVANÍ FYZIKY

Martin Šechný

Bádatelské vyučovanie fyziky môže účelne využiť rôzne IT nástroje. Otvorené IT nástroje prinášajú viaceré výhody. Otvorenosť vyjadrená vhodnou verejnou licenciou dáva používateľovi právo použiť nástroj na akýkoľvek účel a skúmať, ako nástroj funguje. To je sloboda bádania. Efektívne a bezpečné používanie IT nástrojov je obsahom digitálnej gramotnosti pre školu aj prax. Do prehľadu otvorených IT nástrojov pre vyučovanie fyziky možno zaradiť mikroprocesorové stavebnice s paletou elektronických súčiastok, veľké množstvo aplikačného softvéru, simulačný softvér, programovacie jazyky a vývojové prostredia, webové nástroje, virtuálne laboratóriá, katalógy dát, hypertextové dokumenty a iné pomocné zdroje. Zaujímajú nás hlavne také nástroje, ktoré sú dostatočne univerzálne, užitočné a pre žiaka alebo študenta zrozumiteľné, jednoduché. Vhodne zvolené nástroje budú použiteľné vo fyzike, matematike, informatike a v iných predmetoch.

POVINNÉ ZVEREJŇOVANIE ZMLÚV, OBJEDNÁVOK A FAKTÚR

Ondrej Vrábek

Povinnosť zverejňovať zmluvy, objednávky a faktúry sa dá splniť aj efektívnejšie. Projekt Zverejniť.sk ponúka jednoduchú, pohodlnú a efektívnu možnosť zverejňovania dokumentov pre povinné osoby. Všetky nahrané záznamy sú automaticky sprístupnené ako Open data, takže ich je možné jednoducho analyzovať a vyhodnocovať. Projekt zároveň zvyšuje dostupnosť týchto dát pre ľudí s hendikepmi – výstupy sú optimalizované pre čítačky obrazoviek.

Vydanie tohto zborníka bolo podporené grantom slovenskej kultúrno-edukačnej agentúry KEGA č. 041ŽU-4/2017 „Experimentálna matematika - prístupná pre všetkých“.

OTVORENÝ SOFTVÉR VO VZDELÁVANÍ, VÝSKUME A V IT RIEŠENIACH
Zborník príspevkov medzinárodných konferencií
OSSConf 2018 2.–4. júla 2018 v Žiline,
OSSConf 2019 2.–4. júla 2019 v Žiline

Prvé vydanie 2019
Elektronická sadzba programom pdf \LaTeX Rudolf Blaško
Vydala Žilinská univerzita v Žiline
Tlač EDIS – vydavateľstvo Žilinskej univerzity v Žiline
Náklad 100 ks

ISBN 978-80-554-1627-4

ISBN 978-80-554-1627-4



9 788055 416274