

Spoločnosť pre otvorené informačné technológie

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



**Zborník príspevkov medzinárodnej konferencie
OSSConf 2013**

**2.–4. júla 2013
Žilina, Slovensko**

SPOLUORGANIZÁTOR KONFERENCIE



FAKULTA RIADENIA
A INFORMATIKY
ŽILINSKÁ UNIVERZITA

SPONZORI KONFERENCIE



PARTNERI KONFERENCIE



Spoločnosť pre otvorené informačné technológie

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



**Zborník príspevkov medzinárodnej konferencie
OSSConf 2013**

**2.–4. júla 2013
Žilina, Slovensko**

Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach

2.–4. júla 2013, Žilina, Slovensko

Vedeckí garanti konferencie:

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

prof. Ing. Miloš Šrámek, PhD., Austrian Academy of Sciences, Wien (AUT); SOIT, Bratislava

Vedecký a programový výbor:

prof. Ing. Miloš Šrámek, PhD., Austrian Academy of Sciences, Wien (AUT) – **predseda**

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

doc. Ing. Karol Grondžák, PhD., Žilinská univerzita, SOIT, Žilina

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

RNDr. Ján Buša, CSc., FEI, Technická Univerzita Košice, SOIT, Košice

Mgr. Peter Czimmermann, PhD., Žilinská univerzita, SOIT, Žilina

Mgr. Michal Kaukič, CSc., Žilinská univerzita, SOIT, Žilina

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

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

Ing. Tatiana Šrámková, FEI, Slovenská technická univerzita, SOIT, Bratislava

Organizačný výbor:

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

Rudolf Blaško, Žilinská univerzita, SOIT, Žilina

Roman Hajtmanek, SOIT, Žilina

Michal Kaukič, Žilinská univerzita, SOIT, Žilina

Tomáš Majer, Žilinská univerzita, SOIT, Žilina

Ľubica Micháľková, Žilinská univerzita, Žilina

Peter Mráz, DRU, SOIT, Kremnica

Miloslav Ofúkaný, GeoCommunity, SOIT, Bratislava

Peter Štrba, Gymnázium Mikuláša Galandu, SOIT, Turčianske Teplice

Vydavateľ: Spoločnosť pre otvorené informačné technológie, Bratislava

ISBN 978-80-970457-3-9

Copyright © 2013 autori príspevkov

Ktokoľvek má dovolenie 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.

Recenzenti:

Blaško, Rudolf, RNDr., PhD.
Fodrek, Peter, Ing., PhD.
Grondžák, Karol, doc., Ing., PhD.
Kaukič, Michal, Mgr., CSc.
Kozubík, Aleš, RNDr., PhD.
Kovalík, Štefan, RNDr., PhD.
Levashenko, Vitaly, doc., Ing., PhD.
Ofúkaný, Miloslav, Mgr.
Peško, Štefan, doc., RNDr., PhD.
Petrovič, Pavel, Mgr., PhD.
Rybička Jiří, doc., Ing., Dr.
Stříž, Pavel, Ing., Ph.D.
Ševčovič, Ladislav, RNDr., PhD.
Šrámek, Miloš, prof., Ing., PhD.
Zaitseva, Elena, doc., Ing., PhD.

Editori:

Rudolf Blaško
Aleš Kozubík
Pavel Stříž
Ladislav Ševčovič

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

Obsah

Úvod	7
Cena SOIT a Liberixu za najlepšie študentské práce	9
Pozvaní prednášatelia	13
Robert Mařík PdfL ^A T _E Xa tvorba nestatických elektronických dokumentů	17
Otvorený softvér vo vede a výskume	25
Ján Boháčik, C. Kambhampati, Darryl N. Davis, Miroslav Benedikovič Risk Estimation of Heart Failure Patients Using Weka	27
Roman Hajtmanek Heuristické algoritmy pre maticové rovnomerné rozvrhy	33
Michal Chovanec, Jaroslav Múčka Akcelerometrické meranie výstrelu z luku	39
Jozef Kostolny Logging Tool — The Log4j	47
Marian Kováč Redis — Non-Relation Database System	55
Miroslav Kvaššay Algorithm for Finding All Minimal Cut Sets in Reliability Analysis Based on Logic Differential Calculus	63
Pavel Stríž Stahování přes python-selenium	71

Martin Šechný Operačný systém GNU/Linux vo vyučovaní na strednej škole	77
TEX a jeho priatelia	85
Rudolf Blaško Cykly vo formáte L ^A TEX	87
Kristýnka Hůlková, Pavel Stříž Týdenní dovolená v zemi tangramů	95
Tereзка Hůlková, Pavel Stříž Představení programu musescore	103
Aleš Kozubík Chemická sazba s balíčkem chemfig	107
Tibor Menyhért, Pavel Stříž Zkušenosti s pluginem WP-quickL ^A TEX	115
Jiří Rybička Sazba not v systému MusixT _E X	123
Pavel Stříž Jak se T _E Xista mezi divadelními ochotníky neztratí	131
Abstrakty nerecenzovaných ukážok a prezentácií	139

Vážení čitatelia,

dostáva sa vám do rúk zborník, ktorý je výstupom v poradí už piatej samostatnej konferencie, venovanej slobodnému a otvorenému softvéru a jeho využitiu vo vzdelávaní, vede a ostatných oblastiach, ktoré si vyžadujú IT riešenia. Hoc v poradí piata, ostáva naša konferencia stále prvou a jedinou konferenciou na Slovensku, venovanou OSS.

Ako som už spomenul, naša tohtoročná konferencia je už v poradí piata. Dovoľte mi teda pri príležitosti tohto nášho malého jubilea trochu bilancovať. Prvé riadky dejín našej konferencie sa začali písať už v rokoch 2007 a 2008, kedy sekcia o slobodnom otvorenom softvéri fungovala ako sekcia pod strechou medzinárodnej konferencie *Aplimat* a taktiež sme sa stretávali pri príležitosti *Víkendov s Linuxom*. Tak uzrela svetlo sveta myšlienka oddelenia tejto sekcie do podoby samostatnej konferencie venovanej OSS, ktorá sa dožila svojej prvej realizácie v roku 2009. Od vtedy sa stalo už tradíciou, že sa pravidelne na začiatku júla stretávame v priestoroch Fakulty Riadenia a Informatiky Žilinskej Univerzity. Za štyri roky tu odznelo takmer 200 prednášok a ukážok. Z pôvodne jedinej sekcie Aplimatu sa náš OSSConf rozrástol do podoby, v ktorej má pravidelne päť a viac samostatných sekcií. Pravidelnou súčasťou sa stal aj seminár pre učiteľov informatiky. Tradíciou sa stala taktiež cena SOIT a Liberixu udeľovaná najlepším študentským prácam zaoberajúcim sa problematikou OSS.

A kam sa za týchto, pre niekoho dlhých, pre niekoho krátkych päť rokov posunul otvorený softvér? Nenáleží mi právo súdiť, či môžeme byť s výsledkom spokojní alebo nie, no nemôžem si nevšimnúť, že aj na tomto poli nastal významný pokrok. Dnes už nie sme považovaní za akýchsi exotov, ktorí sa zaoberajú čímisi, čo nikto iný nepoužíva, či za čudákov z akademického prostredia. Nik sa už nepozastavuje nad tým, ak povie, že bežíte po Linuxom. Takmer nestretnete užívateľa, ktorý by nepoznal Firefox, bežne používame smartfóny s Androidom a ani záverečné práce sádzané v $\text{T}_{\text{E}}\text{X}$ -u už nie sú výnimočné. Žiaľ, len v oblasti verejnej a štátnej správy stále zaostávame za svetovým trendom a sporadické výskyty OSS vo verejnom živote sú stále len ojedinelými lastovičkami, ktoré jar nerobia. Ostáva nám veriť, že aj predchádzajúce ročníky nášho OSSConfu napomohli k rozšíreniu otvorených technológií a nádejať sa, že v budúcnosti napomôže aj k prelomeniu ľadov v tej poslednej menovanej oblasti.

Ak už sme sa pustili do bilancovania, patrí sa venovať aj cieľom a víziám. Pravidelní účastníci konferencie určite postrehli niekoľko zmien v zložení tímu organizátorov. Aké sú teda naše nové vízie a plány? Prvou podstatnou zmenou je, že sme sa rozhodli z programu konferencie vypustiť sekciu OSS vo verejnej správe, ktorú chceme realizovať ako samostatné stretnutie k tejto téme v Bratislave. Od tejto zmeny si sľubujeme vyššiu návštevnosť kompetentných predstaviteľov štátnej a verejnej správy, ktorých sídla sa sústreďujú v hlavnom meste. Druhou zmenou, ktorú sa nám však nepodarilo naplniť, malo byť naštartovanie novej sekcie venovanej vývoju pre OS Android a mobilným zariadeniam. Ale niekoľko prvých príspevkov sa v rámci vývojárskej sekcie objavilo a tak sme presvedčení, že budú zárodkom takejto sekcie v budúcnosti. Treťou, a poslednou významnou inováciou v budúcnosti je naša ambícia premeniť učiteľské semináre z podoby „seminárov pre učiteľov informatiky“ na „semináre pre učiteľov *nielen* informatiky“. Teda do podoby, kde každého učiteľa chápeme ako užívateľa IT technológií, ovplyvňujúceho práve svojou aktivitou a používaním OSS a otvorených dát povedomie svojich

žiakov či študentov a upriamiť tak ich pozornosť týmto smerom. Taktiež budeme radi, ak sa nám zo seminára, ktorý bol „čisto slovenskou záležitosťou“ podarí urobiť podujatie s nadnárodným dosahom, aspoň v rámci jazykovo blízkych krajín bývalej federácie.

Už som sa dotkol faktu, že niektorí z tradičných aktívnych organizátorov sa rozhodli postupne opustiť naše rady či participovať na menej exponovaných pozíciách. Ich dôvody boli rôzne. U niekoho osobné, u iných to je únava a u ďalších ambícia skúsiť niečo nové. Všetkým sa im však treba poďakovať za ich nemalé úsilie a vynaloženú energiu, za hodiny práce pre dobrú vec. Do budúca im popriať, aby sa im podarilo ich plány naplniť, problémy všetkých druhov čo najskôr prekonať a vysloviť nádej, že sa do nášho kruhu, kde sme si už na nich na všetkých zvykli, vrátia. Dostáva sa Vám teda do rúk zborník tak trochu novej a trochu starej konferencie OSSConf. Toho nového sem sa už dotkli a čo ostáva staré? Staré ostáva to, že konferencia je stále jednou z aktivít SOIT, zameranou na propagáciu otvoreného softvéru. Starým dobrým zvykom ostáva aj to, že príspevky a prezentácie či videozáznamy z konferencie nájdete na webových stránkach konferencie <http://ossconf.soit.sk>. Starou ostáva aj obsahová skladba zborníka, v ktorom nájdete jednak plné znenie recenzovaných príspevkov a abstrakty nerecenzovaných vyštúpení.

Naša konferencia, tento zborník a ostatné materiály konferencie aj samotný fakt, že sa toto podujatie vôbec mohlo uskutočniť je kolektívnym dielom všetkých organizátorov, editorov, autorov príspevkov a prezentácií, recenzentov a samozrejme aj všetkých účastníkov. Nemohlo by však prebehnúť bez podpory sponzorov a partnerov, ktorí naše podujatie podporili alebo dlhodobo podporovali v minulosti. Chceme sa v tejto súvislosti poďakovať Slovenskej Infromatickej Spoločnosti, ktorá každoročne poskytuje finančné prostriedky na granty pre učiteľov prichádzajúcich na seminár, spoločnostiam Red Hat, HP Slovensko, Gista a DRU, za ich dlhodobú podporu konferencie a partnerom CSTUG, Liberix, Geoinformatika, freemap.sk, LinuxOn, LinuxOS.sk, nakladateľstvi Martin Stríž, GISportal.cz, promospravy.sk, SOŠe Poprad-Matejovce, Linux Vacation Eastern Europe.

Osobitné poďakovanie patrí vedeniu Fakulty Riadenia a Informatiky ŽU zastúpenému osobou jej dekana pána prof. Ing. Karola Matiaška, PhD., za každoročnú záštitu nad konferenciou a bezodplatné poskytnutie konferenčných miestností a laboratórií v priestoroch fakulty.

Dostávam sa tak k samotnému záveru nevďačnej úlohy napísať úvod ku zborníku. Úvod, o ktorom jedni hovoria, že ho aj tak nikto nečíta a druhí hovoria, že ho číta každý, aj ten, kto už ďalej nečíta nič. Tak teda verím, že tí prví o nič neprišli a tých druhých som neodradil od ďalšieho čítania . . .

Všetkých účastníkov konferencie vítam na jej piatom ročníku a želám im veľa poučenia a príjemných zážitkov.

Za organizačný výbor OSSConf 2013

Aleš Kozubík
predseda

CENA SOIT A LIBERIXU ZA NAJLEPŠIE ŠTUDENTSKÉ PRÁCE

Spoločnosť pre otvorené informačné technológie SOIT a Obecně prospěšná společnost zabývající se propagací a podporou svobodných informačních technologií Liberix vyhlásili v roku 2013 už 4. ročník soutěže o nejlepšíu bakalářsku a diplomovou práci súvisiacu s otvoreným softvérom a ďalšími otvorenými technológiami. Cieľom súťaže je stimulovať záujem o používanie OIT na školách, pretože sú to práve školy, na ktorých je propagácia progresívnych technológií najúčinnnejšia. Vypísaním Ceny chcú organizátori vysloviť uznanie pedagógom, ktorí sa touto formou venujú propagácii OIT.

SOIT (SK) a Liberix (CZ)

Spoločnosť pre otvorené informačné technológie a spoločnosť Liberix vyhlásili súťaž o najlepšiu bakalársku a diplomovú prácu súvisiacu s otvoreným softvérom a ďalšími otvorenými technológiami. Cieľom súťaže bolo stimulovať záujem o používanie OIT na školách, pretože sú to práve školy, na ktorých je propagácia progresívnych technológií najúčinnnejšia. Vypísaním Ceny chcú organizátori súčasne vysloviť uznanie pedagógom, ktorí sa touto formou venujú propagácii OIT.

Cieľovou skupinou Ceny boli slovenské a české univerzity so zameraním najmä na informačné technológie, pričom udeľovanie Ceny SOIT a Liberixu je súčasťou konferencie Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach. Posudzovateľmi prác zaslaných v roku 2013 boli *Slavko Fedorik, Karol Grondžák, Michal Kaukič, Pavol Lajčiak, Tomáš Majer, Peter Mráz, Matúš Chochlík* a *Miloš Šrámek*.

Laureáti Ceny

Do súťaže za rok 2013 bolo zaslaných 17 prác. Posudzovatelia rozhodli o prvých troch miestach a laureátoch Ceny takto:

1. cena

Filip Kosík, Fakulta informatiky Masarykovy univerzity v Brně:

Testování uživatelského prožitku instalátoru linuxové distribuce

Diplomová práca, vedúci práce: Marek Grác

Abstrakt. Ve své diplomové práci jsem se zabýval problematikou uživatelského prožitku (UX) a použitelnosti softwarových produktů. Po nastudování problematiky jsem začal s přípravou testování použitelnosti reálného softwarového produktu. Tímto produktem byl přepracovaný instalátor Anaconda (pro linuxovou distribuci Fedora 18). Měl jsem připravit veškeré potřebné podklady (včetně testovacích scénářů) a zajistit záznam průběhu sezení. Pořízená data z testování jsem zpracoval do podrobné zprávy, popisu zachycených problémů a doporučení na jejich opravu. Nakonec jsem ještě zanalyzoval beta verzi Fedory 19, která některé úpravy instalátoru již zohlednila.

2. cena

Marianna Takáčová, Univerzita Komenského v Bratislave:

Odborný preklad v kontexte lokalizácie softvéru

Diplomová práca, vedúci práce: Miloš Šrámek

Abstrakt. Diplomová práca sa zaoberá prekladom softvéru v rámci lokalizácie softvéru so zameraním na otvorený softvér. Zaoberá sa postupmi a problémami pri preklade, prináša prehľad dostupných otvorených nástrojov na preklad softvéru a porovnáva ich. Praktickou časťou práce je anglicko-slovenský slovník odborných informatických pojmov dostupný vo formátoch PO a TMX pod licenciou GFDL.

3. cena

Michal Chovanec, Fakulta riadenia a informatiky, Žilinská univerzita:

Operačný systém pre jednočipové mikropočítače s jadrom Cortex-M3

Diplomová práca, vedúci práce: Ján Kapitulík

Abstrakt. V práci je predstavená realizácia operačného systému pre jednočipové mikropočítače rodiny ARM Cortex-M3 a Cortex-M4. Celý systém bol napísaný a vyvinutý použitím open source nástrojov v prostredí Ubuntu Linux. Operačný systém realizuje preemptívny multitasking, správu kritických sekcií, systém posielania správ a základné knižnice pre spoluprácu s hardvérom. Výsledné riešenie bolo implementované a odladené na doske STM32 Discovery kit, vlastnej doske s STM32 a vývojovom kite Stellaris Launchpad. K jadru systému je pripojených niekoľko príkladov, ktoré demonštrujú funkcie systému a uľahčujú vývoj vlastnej aplikačnej časti.

Ostatné prihlásené práce

Okrem ocenených prác boli do súťaže prihlásené tieto práce (bez poradia):

Lóránt Oroszlány:

Octave System Sound Processing Library

Fakulta informatiky Masarykova Univerzita Brno,

vedúci práce: Luděk Bártek

študenta: Petra Mojžišová:

Využívanie bezplatných informačných a komunikačných technológií v malých a stredných podnikoch

Fakulta ekonomiky a manažmentu, SPU Nitra,

vedúci práce: Marcela Hallová

Lukáš Lamper:

Modulové, inteligentné programové prostredie na efektívne spravovanie servera s voliteľnými parameterami

Fakulta elektrotechniky a informatiky, STU Bratislava,

vedúci práce: Alexander Hambalík

Michal Toman:

LLVM IR Service for Fedora

Masarykova Univerzita, Brno, Fakulta informatiky,

vedúci práce: Jan Strejček

Roman Došek:

OS Linux na platformě ARM

Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky,

vedúci práce: Tomáš Dulík

Miroslav Košík:

Experimental study of progressive multi-factor authentication methods and apparatus in network environment

Technická univerzita v Košiciach,
vedúci práce: František Jakab

Jakub Hadvig:

Virtualization and repository management in academic environment

Masarykova Univerzita, Fakulta informatiky,
vedúci práce: Ivan Nečas

Vladimír Ruman:

Inovatívny platformovo nezávislý VPN klient

Fakulta informatiky a informačných technológií, STU v Bratislave,
vedúci práce: Peter Vilhan

Lukáš Melega:

Testovanie web stránok s formulármi

Technická univerzita v Košiciach,
vedúci práce: Csaba Szabó

Tomáš Oberle:

Možnosti aplikácie knižnice SQLite

Technická univerzita v Košiciach,
vedúci práce: Csaba Szabó

Ondřej Kozina:

Rozhraní pro správu virtuálních blokových zařízení z pohledu víceúrovňové administrace

Fakulta informatiky, Masarykova univerzita, Brno,
vedúci práce: Marek Grác

Peter Mačkay:

Monitoring aplikací běžících v JVM pomocí JVM Tool Interface (JVMTI)

Fakulta informatiky, Masarykova univerzita, Brno,
vedúci práce: Marek Grác

Michal Novák:

Dokumentační nástroj pro aplikační rozhraní REST

Fakulta informatiky, Masarykova univerzita, Brno,
vedúci práce: Marek Grác

POZVANÍ PREDNÁŠATELIA

Ako zabezpečiť súkromie na smartphone telefónoch

Abstrakt: Cieľom prezentácie je popísať možnosti ochrany a zabezpečenie súkromia používateľov mobilných zariadení pri komunikácii hlasom, mailami, či interaktívnymi textovými správami využitím protokolov SIP/TLS ZRTP, PG-P/SMIME, či OTR. Súčasne v prezentácii budú uvedené možnosti ochrany voči neoprávneného úniku dát, či forenznej analýzy využitím plného šifrovania (tzv. "full disk encryption") zo samotného mobilného zariadenia. Ako ochranu voči life-loggingu sociálnych sietí, monitorovaniu známych vyhľadávačov, či trackovaniu všetkých platobných transakcií vašimi platobnými spoločnosťami, autor uvedie možnosti alternatívnych identít, anonymizačných sietí (Tor, I2P) a používanie decentralizovanej kryptomeny Bitcoin. V prednáške odznejú konkrétne implementácie bezpečnostných a anonymizačných aplikácií a nástrojov pre najrozšírenejšiu mobilnú platformu Android vrátane bitcoinovej peňaženky.



Pavol Lupták je certifikovaným bezpečnostným špecialistom a výkonným riaditeľom firmy Nethemba. Vyštudoval FEI-STU v Bratislave a FEL-ČVUT v Prahe odbor informatika s diplomovou prácou zameranou na ultrabezpečné systémy. Je držiteľom prestížnych bezpečnostných certifikácií CISSP a CEH, vedie slovenskú OWASP pobočku. Je tiež spoluzakladateľom organizácií Progressbar a SOIT, kde vedie sekciu pre IT bezpečnosť. Máva pravidelné prezentácie na rôznych svetových bezpečnostných konferenciách (Holandsku, Luxemburgsku, Berlíne, Varšave, Krakove, Prahe).

V minulosti demonštroval možnosť zneužitia SMS lístkov vo všetkých veľkých mestách Európy, spolu s kolegom Norbertom Szeteiom demonštroval masívne prelomenie čipových kariet Mifare Classic. Má 14-ročné skúsenosti v oblasti IT bezpečnosti a penetračného testovania a tvorby najrozličnejších bezpečnostných auditov vrátane sociálneho inžinierstva, či forenznej analýzy.

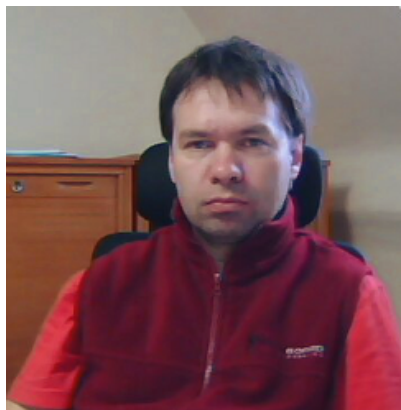
Je spoluautor testovacej príručky OWASP Testing Guide v3, detailne ovláda OSSTMM, ISO17799/27001, má dlhoročné skúsenosti s manuálnym vyhľadávaním zraniteľností a rôznymi bezpečnostnými nástrojmi. Ovláda množstvo programovacích jazykov (ASM, C, C++, XSLT, Perl, Java, PLSQL, Lisp, Prolog, skriptovacie jazyky) a operačných systémov, venuje sa tiež VoIP a zaujímavému výskumu v oblasti IT bezpečnosti.

PDF \LaTeX a tvorba nestatických elektronických dokumentů

Abstrakt: Elektronické publikování v nejrůznějších formách zažívá díky vývoji technologií dlouhodobý a setrvalý rychlý vývoj. Elektronické publikace mohou nabídnout oproti publikacím statickým celou řadu doplňujících funkcí.

Velké procento uživatelů zaměřených na technické a matematické obory používá k přípravě publikací a e-publikací volně šiřitelný systém TeX (LaTeX). TeX se díky svému výstupu do PDF formátu s elektronickým publikováním vhodně doplňuje. Přítomnost programovacího jazyka v TeXu navíc umožňuje rozšiřovat možnosti výstupů a plně využívat všech dostupných vlastností PDF souborů. V příspěvku shrneme, jaké interaktivní objekty jsou v PDF formátu dostupné, jak tyto vlastnosti využívají tvůrci stávajících LaTeXových rozšíření (pdfcomment, ocg-p, ocgtools, fancytooltips, AcroTeX a další) a jak co nejlépe a nejefektivněji tyto funkce jako autor dokumentu využít.

Nedílnou součástí každé publikace je kromě obsahové stránky i design. Ukážeme si, jak pomocí TeXu odstranit některá omezení formátu PDF a jak TeX použít k tvorbě uživatelsky přívětivých dokumentů.



Robert Mařík vyučuje na Mendelově univerzitě matematiku. Po celou svou profesní kariéru se kromě teorie diferenciálních rovnic zabývá publikováním v systému LaTeX. Aktivně se podílí na vývoji maker a balíčků užitečných pro tvorbu elektronických prezentací a učebních materiálů. Je autorem několika softwarových řešení založených na využití OpenSource programů. Nejznámější z nich je aplikace Mathematical Assistant on Web pro automatické řešení vybraných úloh vysokoškolské matematiky se zobrazením postupu výpočtu.

PDF \LaTeX A TVORBA NESTATICKÝCH ELEKTRONICKÝCH DOKUMENTŮ

ROBERT MAŘÍK (CZ)

Abstrakt. V příspěvku shrneme současné dostupné možnosti a nástroje zařazení interaktivních prvků do PDF dokumentů. Vhodné používání těchto prvků vede k tvorbě atraktivnějších a obsahově bohatších souborů. Důraz bude kladen na tvorbu učebních materiálů pomocí pdf \LaTeX u.

Klíčová slova. \LaTeX , PDF.

PDF \LaTeX IN CREATION OF NONSTATIC ELECTRONIC PUBLICATIONS

Abstract. We summarize existing tools and methods for inserting interactive objects into PDF documents. The main stress is given on tools and techniques suitable and capable to create eye candy and easy to use teaching materials with pdf \LaTeX .

Key words and phrases. \LaTeX , PDF.

Úvod

Prakticky všechny dokumenty vytvářené vědeckými a pedagogickými pracovníky na vysokých školách jsou dokumenty nadčasové, s dlouhou životností. Pokud obsahují nějaké komplikovanější formátování než běžné kancelářské dokumenty je vhodné pořizovat jinak, než v sice běžných, ale často aktualizovaných programech a kancelářských balících s problematickou zpětnou kompatibilitou. Pokud se navíc jedná o texty matematického nebo technického charakteru, jsou \TeX nebo \LaTeX téměř jasnou volbou. Vzhledem k početnosti uživatelů se zaměříme spíše na \LaTeX . Je vhodné si uvědomit, že \LaTeX je mnohem více než systém pro snadnou sazbu a archivaci matematických a dalších textů. \LaTeX je programovací jazyk, který umožňuje provádět s textem operace, které u WYSIWYG programů nemají obdobu.

V tomto příspěvku si ukážeme některé dostupné nástroje, které umožní naše texty obohatit o nestatický obsah a vytvořit tak nestatický dokument. Nestatickým dokumentem budeme v tomto příspěvku rozumět dokument obsahující prvky reagující na činnost uživatele. To si vyžaduje zobrazení dokumentu v prohlížeči, který tyto interaktivní prvky podporuje. Pokud se jedná o PDF dokument

pracující s Javascripty, je nutné použít pro práci s takovým dokumentem zpravidla pouze program Adobe Reader, částečná podpora Javascriptů je i v programu FoxitReader.

Příspěvek je členěn následovně. V úvodní části zrekapitulujeme některé L^AT_EX-ové balíčky umožňující tvořit nestatické PDF dokumenty. Poté si ujasníme, z jakých stavebních prvků může být tento nestatický dokument sestaven a na závěr odkážeme na nástroje, metody a zdroje informací pro práci s těmito stavebními prvky.

1. Ukázky nestatických PDF dokumentů

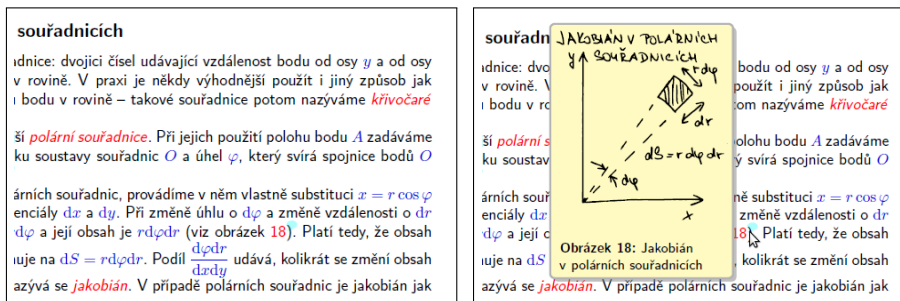
1.1. Balíčky `ocg-p` a `ocgtools`

Balíčky `ocg-p` a `ocgtools` pracují s vrstvami v PDF dokumentech (viz dále). Balíček `ocg-p` umožňuje vkládat do PDF dokumentu materiál, který je viditelný nebo tisknutelný pouze za určitých podmínek. Přímo v balíčku `ocg-p` jsou například makra pro sazbu tabulek, které se samy seřadí podle sloupce na jehož záhlaví klikne čtenář myši. Trik spočívá v tom, že se tabulka vysází v několika různě seřazených vrstvách na jedno místo a pro zobrazení a tisk se použije vždy jen jedna z těchto vrstev.

Balíček `ocgtools` obsahuje makra pro vkládání vrstev a aktivních oblastí dokumentu. Aktivní oblasti při kliknutí nebo najetí myši přepnou u odpovídajících vrstev viditelnost. Příkladem použití je vložení náhledu obrázku, který se při kliknutí zvětší na celou stranu PDF.

1.2. Balíček `fancytooltips` a skript `fancy-preview`

Balíček `fancytooltips` funguje z hlediska čtenáře podobně jako `ocgtools` – v dokumentu máme k dispozici aktivní oblasti, které umožňují zobrazit dodatečný materiál. Na rozdíl od balíčku `ocgtools` se však nejedná o kus našeho dokumentu, ale o stránku externího PDF souboru. S balíčkem `fancytooltips` spolupracuje skript `fancy-preview`. Tento skript nejprve projde náš dokument a extrahuje do pomocného dokumentu prostředí, která nás zajímají, například číslované rovnice, obrázky, tabulky a odkazy na literaturu. Poté skript provede kompilaci pdfL^AT_EXem s tím, že předefinuje příkazy `\ref` a `\eqref` tak, aby kromě čísla odkazovaného objektu a hypertextového odkazu vkládaly i tlačítko, které odkazovaný objekt zobrazí vložený do aktuální strany na místě ve kterém je kurzor. Možné aplikace zahrnují tvorbu učebních textů a vědeckých článků, kde se při najetí za odkaz na rovnici, obrázek nebo položku seznamu literatury zobrazí tento odkazovaný objekt v aktuální stránce. K odskoku na stránku s objektem dojde až při kliknutí. Výhodou je, že pro rychlé zorientování stačí náhled a odpadá tím nutnost návratu na stranu, ze které vedl odkaz.



Obrázek 1. Fancy-preview: čistý text a text s aktivovaným náhledem odkazovaného obrázku.

1.3. Balíček dps – párovací hra Das Puzzle Spiel

Balíček `dps` z programového balíku AcroT_EX slouží pro tvorbu výukových her ve formě dokumentů obsahujících zaškrťovací políčka rozdělená do dvou skupin, například otázky a odpovědi. Jedna odpověď patří vždy k jedné otázce. Úkolem čtenáře je spárovat otázky a odpovědi. Při každém vytvoření správného páru se objeví políčko tajenky. Hra je vhodná například pro oživení vyučovacích hodin. Snímek obrazovky na obrázku 2 je příkladem takové hry. Jedná se o dokument vytvořený pro projekt Matematika s radostí ([1], <http://msr.vsb.cz>) s použitím vlastní modifikace balíčku `dps`. Tato modifikace zahrnuje vložení tajenky do vrstev místo do textového pole (umožní mít v tajence nejenom čistý text, ale libovolný objekt vysázený T_EXem) a náhodné permutování otázek a odpovědí při otevření PDF dokumentu (s využitím stejného triku jako u výše popsanych tabulek balíčku `ocg-p`, které se řadí podle různých kritérií).

1.4. Systém AcroT_EX

Systém AcroT_EX (<http://www.acrotex.net>) je systém pro tvorbu elektronických testů. V jeho možnostech je tvorba testů s testovými otázkami, které jsou uzavřené (volba jedné nebo několika odpovědí z nabízených variant) nebo otevřené (odpověď je textový nebo matematický výraz, v případě matematického výrazu je správnost odpovědi vyhodnocována numerickým výpočtem a procedury pro vyhodnocení správnosti si poradí i s případy, kdy je správná odpověď dána jednoznačně až na aditivní nebo multiplikatívni konstantu). Správnost odpovědi může být vyhodnocena buď okamžitě nebo až po zodpovězení všech otázek. Podle nastavení může být řešitel testu informován buď jenom o počtu správných odpovědí a počtu získaných bodů, nebo mohou být do testu i vyznačeny opravy a správné odpovědi. Pro projekt Matematika s radostí jsme styly AcroT_EXu rozšířili tak, aby při otevření PDF souboru došlo k náhodnému permutování variant jednotlivých odpovědí podobně, jako je tomu u párovacích her.

Rečník ██████████ posluchače. (Winston Churchill (1874–1965), britský politik)

Každou lineární funkci je možné zapsat rovnicí ve tvaru $y = ax + b$, kde a , b jsou reálná čísla. Ke každému grafu vyberte odpovídající vlastnosti koeficientů a a b .

Grafy

Parametry

- a $b < 0$
 $a = 1$
- b $b > 0$
 $a \in (1; +\infty)$
- c $b > 0$
 $a \in (-1; 0)$
- d $b < 0$
 $a \in (1; +\infty)$
- e $b > 0$
 $a = 0$
- f $b < 0$
 $a \in (-1; 0)$
- g $b > 0$
 $a \in (-\infty; -1)$
- h $b > 0$
 $a \in (0; 1)$

Obrázek 2. Párovací hra.

2. Typy nestatických objektů v PDF

V této podkapitole si sesumarizujeme, které nejdůležitější objekty zajišťují interaktivitu v PDF souborech.

2.1. Javascripty a formulářové prvky

PDF dokumenty mohou obsahovat formulářové prvky, jako jsou zaškrťávací políčka, rozbalovací menu, textová pole, tlačítka a přepínače. Všechny tyto prvky mohou být viditelné nebo skryté, editovatelné (a klikací) nebo pouze ke čtení. Možnosti formátování nejsou příliš bohaté (je možné nastavení barvy objektu, barvy okraje, tloušťky okraje, jsou omezeny na písmo pro popisek a obdélníkový tvar hranice). Pokud však v praxi nedostačují, je možné omezení plynoucí z dostupných možností PDF formátu obejít tak, že objekt vysázíme s nulovými rozměry průhledný a bez popisku a pod něj vysázíme \TeX em cokoli potřebujeme. K těmto objektům mohou být připojeny Javascripty spouštěné při různých událostech (onblur, mouseover, mouseexit, enter atd.). Platí i částečný opak, že vlastnosti těchto objektů mohou být měněny pomocí Javascriptů. Je tedy možné programově kterýkoliv z těchto objektů učinit skrytým nebo needitovatelným, vložit do něj určitý popisek nebo symbol, nastavit barvu textu, symbolu nebo okraje a podobně.

Další Javascripty je možné spouštět při otevření PDF souboru (do této části zpravidla také zapisujeme počáteční hodnoty společných proměnných a definice funkcí) nebo při otevření určité strany.

2.2. Optional Content Groups (vrstvy)

Části PDF dokumentu mohou být označeny jako Optional Content Groups (OCG, zkráceně layers, tj. vrstvy). U těchto vrstev je možno nastavit, zda se budou či nebudou zobrazovat a zda se budou či nebudou tisknout. Viditelnost vrstev je podobně jako u formulářových prvků možné při smluvených akcích a událostech přepínat pomocí Javascriptů. Ve vrstvě může být libovolný text nebo objekt vysázený \TeX em.

Využití vrstev jsme viděli v balíčcích `ocg-p` a `ocgtools`. Další využití zahrnuje například zatím experimentální možnost balíčku `hyperref` vkládat hypertextové odkazy do PDF dokumentu tak, že se na obrazovce zobrazují barevně a jsou výrazné, ale tisknou se černě a neruší okolní text na papíře barvou nebo šedým odstínem.

2.3. Formulářové prvky podruhé

Pomocí vrstev je možné nechat na stránce objevit nebo zmizet jakýkoliv text, který je na této stránce zapsán. Existuje ještě jeden mechanismus, umožňující zobrazit ve stránce nějaký text či objekt. Tato možnost je založena na tom, že tlačítka kromě popisku mohou obsahovat i ikony – stránky jiného PDF souboru. Tyto ikony je možné pomocí Javascriptů měnit a je také možné nastavit jak se mají ikony chovat a kam umisťovat, pokud velikost ikony neodpovídá velikosti tlačítka. Je tedy možné kamkoliv do hlavního dokumentu vložit neviditelná tlačítka obsahující jako ikony stránky pomocného PDF dokumentu a tlačítka, na která budeme tyto ikony umisťovat dle potřeby. Takto je možno Javascriptem kteroukoliv ikonu zkopírovat ze skrytého tlačítka a zobrazit ji na kterémkoliv jiném tlačítku. Pokud je toto tlačítko průhledné, bez okraje a jen ke čtení, je efekt stejný jako kdybychom v tomto místě odkryli skrytou vrstvu s textem. Výhoda je v tom, že informace v tomto místě nemusela být zapsána autorem dokumentu, ale přenesli jsme ji na cílové místo z úložiště ve formě skrytého tlačítka. Tato technika byla pravděpodobně poprvé použita v balíčku `animfig` pro vkládání animací do PDF souborů, je popsána v [2] a je použita v balíčku `fancytooltips`.

3. Metody vkládání nestatických objektů do PDF

Tato kapitola slouží jako jistý rozcestník pro autory, kteří mají zájem vkládat do PDF dokumentů tvořených pdf \LaTeX em interaktivní prvky. Do PDF souboru je možné vkládat příslušné příkazy přímo pomocí primitivů programu pdf \LaTeX , častější a pohodlnější je však využití balíčků připravených pro tuto práci. Přesné možnosti, názvy a syntaxe příkazů jsou zpravidla součástí dokumentace, proto je

uvádět nebudeme. Místo toho se pokusíme upozornit na některá úskalí čekající na uživatele.

Není-li řečeno jinak, jsou stabilní verze balíčků buď přímo v použitém systému (nejčastěji MikTeX a TeX Live) nebo v archivu CTAN (<http://www.ctan.org>).

Při výměně verzí PDF souboru se spoluautory nebo při výměně rukopisů a opravených rukopisů s redakcí časopisů je běžné používat anotace. Tyto anotace se používají při čtení hotového PDF souboru v příslušném GUI. Kromě toho je možné vkládat anotace již při tvorbě PDF souboru, což umožňuje balíček `pdfcomment`.

Nejužitečnějšími balíčky jsou zřejmě balíčky systému AcroTeX. Tyto balíčky nejsou na TeX Live a je nutné si je v případě použití s TeX Live nainstalovat ručně. Primárním zdrojem k šíření není CTAN, ale autorovy vlastní stránky <http://www.acrotex.net> a <http://www.math.uakron.edu/~dpstory/webeq.html>, verze na CTAN nemusí být aktuální. V praxi se mi dobře osvědčil volně šiřitelný AcroTeX eDucation Bundle, díky podpoře pdfLaTeXu. Na výběr jsou i balíčky AcroTeX eDucation Bundle Pro a AcroTeX Presentation Bundle, které obsahují dodatečné funkce nebo šablony prezentací a vyžadují finální zpracování PDF dokumentu v programu Adobe Professional. V AcroTeXu jsou přítomny například balíček `insdljs` pro vkládání Javascriptů do PDF dokumentů, `eforms` pro vkládání formulářových prvků, `exerquiz` pro tvorbu interaktivních testů. Balíčky `eforms` a `insdljs` jsou vyžadovány i při použití `ocgtools` a `fancytoltips`.

Na dvě úskalí při použití balíčku AcroTeX je vhodné upozornit. První z nich se týká pouze slovenských uživatelů a spočívá v tom, že jeden z regulárních výrazů použitých v Javascriptech dokumentu je v konfliktu s aktivními znaky balíku `babel` při použití jazyka `slovak`. Často pomůže načíst nejdříve balíčky AcroTeXu a teprve poté `babel` a slovenštinu. Další úskalí na tvůrce interaktivních testů spočívá v tom, že aby byly testy schopny vyhodnocovat matematické výrazy, je v Javascriptech těchto dokumentů použita funkce `eval`. Kvůli této funkci označuje Google testy tvořené AcroTeXem jako nebezpečné a chová se k nim jako k virům – neumožní je posílat jako přílohy a nedoručí je na poštovní účet služby Gmail. Jedná se o nepříjemnost, která je známa a postihuje více projektů. Není jasné, zda je možné doufat, že se v tomto směru něco změní, proto je vhodné mít tuto skutečnost buď na paměti nebo v ideálním případě vhodně ošetřenu, abychom nevyklučovali velmi početnou skupinu uživatelů služby Gmail. Například v testech projektu Matematika s radostí [1] tato nepříjemnost odpadá, protože jsme díky nepoužívání otevřených otázek v testech mohli funkci `eval` vymazat.

Z pera stejného autora a ze stejného zdroje jako AcroTeX jsou balíčky `dps` a `jj_game` pro tvorbu párovacích her a hry Jeopardy. Kromě toho pro tvorbu hry Jeopardy existuje na CTAN ještě i balíček `jeopardy`.

Vkládání vrstev do PDF souboru řeší například balíčky `ocg` a `ocg-p`. Balíček `ocg` je starší neudržovaný, je omezen pouze na pdfLaTeX a vykazuje nepříjemné chování při použití velkého množství vrstev (projeví se však až u řádově stovky

vrstev). Proto je vhodnější využívat `ocg-p`. Balíček `ocgtools` pro schovávání materiálu do vrstev a vyvolávání tohoto materiálu pomocí tlačítek byl navržen pro starší verzi `ocg`, měl by však pracovat i s `ocg-p`.

Pokud máme na interaktivní dokument vlastní požadavky, které nejsou řešeny přímo některým \LaTeX ovským stylem, je vhodné si uvědomit, že možnosti ladění Javascriptů v PDF dokumentu jsou (zejména bez použití programu Adobe Professional) velmi omezené. Proto je velmi vhodné naplánovat práci dokumentu jako sled jednotlivých malých kroků, každý krok u kterého to je možné si odladit samostatně a teprve poté vše spojit do jednoho dokumentu. Výhodou tohoto postupu je, že při řešení malých kroků, které se vzpírají našemu \TeX ařskému úsilí je relativně snadné zformulovat prosbu o pomoc na některém z diskusních fór věnovaných \LaTeX u a doufat v pomoc komunity.

Bohatou studnicí nápadů a zdrojem odpovědí na naše otázky je server <http://tex.stackexchange.com>. Zde jsou k dispozici například makra, která dokáží ohraničit úsek textu tak, že při kliknutí na tento text se změní zoom při prohlížení dokumentu tak, aby byl zobrazen právě tento označený úsek¹. V projektu Matematika s radostí jsme například z tohoto serveru využili nápad s makrem, které umí rozřezat obrázek na kousky². Stačilo tyto části vkládat do vrstev odkrývaných při odhalování tajenky párovací hry a rázem jsme měli k dispozici možnost tvořit párovací hry, kde tajenkou není text, ale postupně odkrývaný obrázek.

Při tvorbě interaktivních PDF dokumentů kombinujících použití několika výše uvedených balíčků je možno očekávat problémy, proto je nutno vše nejprve odladit na krátkém dokumentu. Zejména mohou nastat problémy v dokumentech, kde je použit registr `pdfpageattr`, který není zvykem předefinovávat a proto s jeho nenulovým obsahem nemuseli autoři balíčku počítat. Někteří uživatelé jej však předefinovávají, aby předešli zkreslení barev pod Linuxem³.

4. Závěrem

Vzhledem k rozvoji HTML5 a úpadku některých technologií vytvořených firmou Adobe se někdy věští i brzká stagnace a zánik formátu PDF. Přirozenou otázkou tedy je, zda tvorba PDF dokumentů není slepou uličkou.

Dle mého názoru podpora PDF dokumentů v operačních systémech ohrožena není, protože PDF je v podstatě jediný formát používaný pro tvorbu a archivaci vědeckých článků, nepočítáme-li export do html, což je zatím vzhledem ke kvalitě zobrazování matematiky možné brát spíše jenom jako orientační náhled.

Mé přesvědčení, že se od formátu PDF a programu pdf \LaTeX neodvrátí běžní uživatelé typu autor vysokoškolských učebních textů se upevňuje pokaždé, když

¹<http://tex.stackexchange.com/questions/12290/>

²<http://tex.stackexchange.com/questions/70458>

³viz například <http://tex.stackexchange.com/questions/35868> a <http://tug.org/pipermail/pdfptex/2007-December/007482.html>

kolegové píšící matematické texty v programu Microsoft Word upgradují na novou verzi a musí ve svých dokumentech kontrolovat a přepisovat matematické výrazy.

Uživatel pdfL^AT_EXu se však nemusí cítit ohrožen ani v případě, že by úpadek formátu PDF opravdu nastal. Stačí dodržovat jistá pravidla při pořizování dokumentů, zejména dbát na oddělení obsahu dokumentu od definice vizuálního stylu. Například v projektu Matematika s radostí [1] sestavuje tým přibližně třiceti pracovníků interaktivní testy a hry v PDF formátu. Bylo by jistě dobré mít tuto práci uchovanou tak, aby se dala využít i v budoucnu a aby nebyla ohrožena podporou či nepodporou programů pro práci s PDF. Proto k zápisu otázek nepoužíváme přímo příkazy AcroT_EXu, jak je dokumentováno v manuálu AcroT_EXu nebo v [3], ale vytváříme mezi naším dokumentem a AcroT_EXem další vrstvu. Ve stylu máme pro zapsání otázky definován příkaz s jedním nepovinným a dvěma povinnými parametry. Nepovinným parametrem může být klíčové slovo nebo slova ovlivňující formátování otázky, prvním povinným parametrem je text otázky a druhým povinným parametrem jsou varianty odpovědí, kde odpovědi jsou odděleny čárkou a správná odpověď je buď uvedena jako první, nebo začíná hvězdičkou. V otázkách mohou být použita vlastní makra definovaná ve speciální sekci v hlavičce dokumentu. Tento formát byl zpočátku navržen pouze pro snadný a rychlý zápis otázek a odpovědí. Časem se však ukázala vhodnost této volby, když od grafika přišly poznámky ke grafickému vzhledu (například vizuální oddělení otázek a odpovědí) nebo když došlo na rozšíření funkcionality (na rozdíl od běžných testů tvořených AcroT_EXem, testy projektu Matematika s radostí automaticky permutují varianty odpovědí). Předefinováním použitých příkazů dokážeme vyexportovat otázky a odpovědi ve zkompileovatelných tvarech, ať už jako úryvek T_EXovského dokumentu nebo PNG náhled otázky náhledy jednotlivých odpovědí. V případě, že bude dostupná vhodná nová technologie, neměl by pro zkušeného programátora být velký problém přepnout testy na tuto technologii a vdechnout tak AcroT_EXovým testům v PDF nový život.

Reference

- [1] *Matematika s radostí*, projekt OPVK, reg. č. CZ.1.07/1.1.00/26.0042, <http://msr.vsb.cz>.
- [2] MAŘÍK, R.: *Vkládání JavaScriptů pdfL^AT_EXem prakticky*, Zpravodaj ČSTUG. 2007. sv. 2, č. 2, s. 72–83. ISSN 1211-6661.
- [3] MAŘÍK, R. – PLCH, R. – ŠARMANOVÁ, P.: *Tvorba interaktivních testů pomocí systému AcroT_EX*, Zpravodaj ČSTUG. 2010., sv. 20, č. 4, s. 266–299. ISSN 1211-6661.

Kontaktní adresa

doc. Mgr. Robert Mařík, PhD., Ústav matematiky, Fakulta lesnická dřevařská, Mendelova univerzita, Zemědělská 1, 613 00 Brno, Česká republika,
E-mailová adresa: marik@mendelu.cz, <http://user.mendelu.cz/marik>

OTVORENÝ SOFTVÉR VO VEDE A VÝSKUME

Otvorený softvér je často považovaný len za jednu z ciest ako sa dostať k pomerne kvalitnému alternatívnemu softvéru zadarmo. Je to tak aj vo vede a výskume? Moje doterajšie skúsenosti pri vedení diplomantov a doktorandov ale aj vlastnom výskume hovoria, že otvorený softvér ponúka viac.

Mladým začínajúcim vedeckým pracovníkom umožňuje dostať sa k takým softvérovým nástrojom, ktoré sú pre nich inak nedostupné. Ale nie je to úplne zadarmo. Čaká ich nejedno rozhodnutie, od výsledku ktorého závisí, či budú s výberom spokojní. Nehovoriac už o prekážkach, ktoré treba prekonať pri inštalácii a overovaní kvality takýchto produktov. Skúsenosť so zdolávaním takýchto prekážok však prispieva k ich odbornému rast. Otvorený kód je aj príležitosťou pre zvedavých programátorov nahliadnúť do sveta tvorcov kvalitných programov, zoznámiť sa s ich technikami, pokúsiť sa opravu prípadných „bugov“ či tvorbu vlastných doplnkov. K nezaplateniu je aj rýchly neformálny kontakt s „prirodzene otvorenými“ autormi programov.

Čo ponúka nám skôr narodeným, viac či menej poučených z nenaplnených očakávaní? Pre mňa je to možnosť vyskúšať v pomerne krátkom čase nástroje s novými optimalizačnými solvéri. V prípade recenzií a expertných prác tak mám príležitosť ponúknuť alternatívu k jednostrannej orientácii na drahé komerčné riešenia. Pri vlastnom výskume sa zase môžem rýchlejšie rozhodnúť, či sa oplatí vyvíjať nové modely a metódy riešenia alebo postačuje vhodná kombinácia dostupných OSS riešení. Oceňujem tiež užívateľské pohodlie pri nastavovaní parametrov väčšiny OSS nástrojov na požadovanú mieru. Iste nie na poslednom mieste je to i studnica nápadov „ako na to“.

Rád by som upozornil na niekoľko tohoročných príspevkov, ktoré ma poučili.

Dozvedel som sa ako použiť nástroj Weka pre výpočet odhadu rizika u pacientov so srdcovým zlyhaním, ako sa možno pohrať s jazykom Ruby a miniatúrnym akcelerometrom pri meraní dynamických vlastností šípky i ako netradične využiť logický diferenciálny počet v teórii spoľahlivosti. Ostáva mi zaželať, aby sa príspevky v tejto sekcii stali podnetom pre ďalších autorov, ktorí by sa radi podelili nielen o svoje pozitívne ale i negatívne skúsenosti s otvoreným softvérom a jeho tvorbou pri riešení svojich vedeckých projektov.

Štefan Peško

RISK ESTIMATION OF HEART FAILURE PATIENTS USING WEKA

JÁN BOHÁČIK (UK, SK), C. KAMBHAMPATI (UK), DARRYL N. DAVIS (UK)
AND MIROSLAV BENEDIKOVIČ (SK)

Abstract. Two to three percent of the adult population suffer from heart failure and half of all patients with this diagnosis die within four years. To minimize life-threatening situations and costs of treatment, it is interesting to predict if a patient with known heart failure could die soon. For these reasons risk estimation of heart failure patients on the basis of collected data is described in this paper. The risk estimation makes use of some data mining techniques which are implemented in open source software tool Weka - Waikato Environment for Knowledge Analysis. A description of used risk estimation models based on data mining techniques and experimental results showing the performance of these models are also given.

Key words and phrases. Data mining, risk estimation, heart failure.

ODHADOVANIE RIZIKA PRI PACIENTOCH SO SRDCOVÝM ZLYHANÍM POMOCOU NÁSTROJA WEKA

Abstrakt. Dva až tri percentá dospeljej populácie trpí zlyhaním srdca a polovica všetkých pacientov, ktorí majú túto diagnózu zomrú do štyroch rokov. Za účelom minimalizácie životu ohrozujúcich situácií a nákladov na liečbu je zaujímavé odhadovať či by pacient so známym zlyhaním srdca mohol zakrátko zomrieť. Práve v tomto článku je popísané odhadovanie riziku u pacientov so zlyhaním srdca. Pri odhadovaní riziku sa využívajú techniky dolovania z údajov, ktoré sú implementované v open-source softvérovom nástroji Weka — Waikato Environment for Knowledge Analysis. K dispozícii je daný aj popis použitých modelov na odhadovanie riziku založených na technikách pre dolovanie z údajov a experimentálne výsledky ukazujúce výkonnosť jednotlivých modelov.

Kľúčové slová. Dolovanie z údajov, odhadovanie riziku, zlyhanie srdca.

Introduction

According to [5], heart failure makes up an important medical, social, and economic problem. Patients with heart failure suffer disabling symptoms, the most common of which are fatigue and dyspnea, while in terms of disability, the end stage of the disease is comparable to the end stage of terminal cancer. Although there are problems with reliable estimates in many countries, the prevalence of heart failure is about 2%–3% of the adult population and it increases with age. The cost of medical care for heart failure is measured in billions of euros per year.

Within the costs, the most powerful contributing factor is repeated hospitalization. The prevalence of heart failure progressively increased from the early 1950s. It is likely a new increase will be observed in the future mainly because of the aging of the population and because of the trend showing an increasing prevalence of major heart risk factors, including obesity and diabetes.

The long-term prognosis associated with heart failure is not optimistic at all. Half of all patients diagnosed with heart failure die within four years. It is very interesting to predict if a patient dies soon so that an effective prevention can be employed. The goal is to minimize life-threatening situations and to minimize the costs of treatment. Due to massive explosion of data collected about patients in hospitals and home telemonitoring systems, data mining techniques such as a Bayes network classifier and a decision tree classifier can be used. In this paper we concentrate on data mining techniques implemented in open source software tool Weka - Waikato Environment for Knowledge Analysis [4] and we employ it on our heart failure dataset from Hull LifeLab [2].

The paper is organized as follows. Our data about patients with heart failure are described in Section 1. In Section 2 open source software tool Weka and used data mining techniques are discussed. The performance of the techniques on our data is given in Section 3. Section 4 concludes this paper.

1. Heart Failure Dataset

As a heart failure dataset, a group of 2032 patients (\mathbf{V}) classified into two levels of patient status (C) and described by 9 attributes (\mathbf{A}) as queries about clinical findings and physiological measurements is used. Patients and data about them are from Hull LifeLab which is large, epidemiologically representative, information-rich clinical data [2]. Its purpose is studying patients with heart failure so that its definition and diagnosis, its natural history, its mechanisms and markers of progression, the associated costs to health services and society, and the delivery of proven treatment to patients are improved.

The description of attributes and their summary can be seen in Table 1. Describing attributes (\mathbf{A}) are defined as $\mathbf{A} = \{A_1; \dots; A_k; \dots; A_9\}$. If A_k is a categorical attribute, $A_k = \{a_{k,1}; \dots; a_{k,l}, \dots; a_{k,l_k}\}$. Class attribute C is used to classify patients into two possible categorical values c_1 and c_2 (*alive* and *dead*). It expresses if a patient is alive or dead within a period of time after the risk estimation. In the heart failure dataset there are 1512 patients (74.4094%) classified as *alive* and 520 patients (25.5906%) classified as *dead*.

2. Weka and Data Mining Techniques

Weka (Waikato Environment for Knowledge Learning) is an open source data mining system implemented in Java [4]. Releasing Weka as an open source software

Attribute	Data type	Value range
<i>Age</i> (A_1)	Numerical	27 – 96
<i>Blood Creatinine Level</i> (A_2)	Numerical	37 – 1262
<i>Blood Sodium Level</i> (A_3)	Numerical	123 – 148
<i>Blood Uric Acid Level</i> (A_4)	Numerical	0.11 – 1.06
<i>Height</i> (A_5)	Numerical	1.2 – 1.96
<i>NT-proBNP Level</i> (A_6)	Numerical	0.89 – 18236
<i>Pulse Rate</i> (A_7)	Numerical	38 – 150
<i>Sex</i> (A_8)	Categorical	<i>female</i> ($a_{8,1}$) <i>male</i> ($a_{8,2}$)
<i>Weight</i> (A_9)	Numerical	29.8 – 193.8
<i>Patient Status</i> (C)	Categorical	<i>alive</i> (c_1) <i>dead</i> (c_2)

Table 1. Heart failure dataset.

and implementing it in Java are two factors which ensure that it remains maintainable and modifiable irrespective of the commitment or health of any particular institution or company.

Its aim is to provide a comprehensive collection of machine learning algorithms and data preprocessing tools to researchers and practitioners alike. It can cope with preprocessing and data analysis, classification models, association models, and evaluation metrics. There are three modes of Weka operation:

- a) GUI,
- b) command-line,
- c) Java API.

Java API allows to make computer programs for solving classification tasks. In the `weka.classifiers` package, the most important class is *Classifier*. It is a general scheme for any classification model in Weka. *Classifier* contains two significant methods, *buildClassifier()* and *classifyInstance()*. The former is for building the classification model, the latter is for determining the value of class attribute C when the values of all attributes in \mathbf{A} are known (i.e. classification).

We used two principally different data mining techniques for classification: a Bayes network classifier (BNC) and a decision tree classifier (DTC). They can be described as follows. A BNC is based on a Bayesian network which represents a joint probability distribution over a set of categorical attributes [1]. Since it is

a distribution over a set of categorical attributes, numerical attributes in \mathbf{A} are discretized and transformed into categorical.

It consists of $\langle G; \Theta \rangle$, a directed acyclic graph G consisting of nodes and arcs and conditional probability tables $\Theta = (\theta_{A_1}; \dots; \theta_{A_k})$. The nodes represent attributes in \mathbf{A} and attribute C whereas the arcs indicate direct dependencies. The Bayesian network allows the computation of the (joint) posterior probability distribution of any subset of unobserved assignments of values to attributes in \mathbf{A} , which makes it possible to use for determination of $c_j \in C$. A DTC consists of a decision tree which is generated on the basis of instances in \mathbf{V} [3].

There are two types of nodes in the decision tree:

- a) the root and internal nodes (associated with an attribute $A_k \in \mathbf{A}$);
- b) leaf nodes (associated with a $c_j \in C$).

Basically, each non-leaf node has an outgoing branch for each possible value $a_{k,l} \in A_k$, $A_k \in \mathbf{A}$ is an attribute associated with the node. Numerical attributes $A_k \in \mathbf{A}$ are discretized. Value $c_j \in C$ is determined for a new instance using a decision tree, beginning with the root, successive internal nodes are visited until a leaf node is reached. At the root node and at each internal node, a test is applied. The outcome of the test determines the branch traversed, and the next node visited. Value $c_j \in C$ for the instance is simply c_j of the final leaf node.

3. Experimental Results

The main purpose of the experiments is to compare the performance of a Bayes network classifier and a decision tree classifier on our heart failure dataset. The classifiers are implemented in Weka [4]. The performance of the classifiers is measured with sensitivity $\frac{tp}{tp + fn}$ and specificity $\frac{tn}{tn + fp}$.

In the formulas, $tp/fp/fn/tn$ is the number of true positives/false positives/false negatives/true negatives. “*C is alive*” is considered negative, “*C is dead*” is considered positive. Values tp , fp , fn and tn are computed during 10-fold cross-validation. In 10-fold cross-validation, the (fuzzified) data is partitioned into 10 folds of patients. The partition is random, but all folds contain roughly the same proportions of alive and dead patients. Of the 10 folds, a single fold is retained as the testing data for evaluation of discovered knowledge, and the remaining 9 folds are used as the learning data. The learning data is analyzed by the algorithm for the purpose of discovering the knowledge. The cross-validation process is repeated 10 times, with each of the 10 folds used exactly once as the testing data.

The results of our experiments are given in Table 2. Bayes denotes a Bayesian network classifier implemented in Weka as class BayesNet. DTC is a decision tree classifier implemented in Weka as class J48. Sensitivity is sensitivity in percentages, Specificity is specificity in percentages and Sum is the sum of Sensitivity

Classifier	Sensitivity	Specificity	Sum
BNC	24.038	95.37	119.408
DTC	40.96	87.90	128.86

Table 2. Experimental results.

and Specificity. It is very important to avoid classification of dead patients as alive as it would lead to life-threatening situations. Many alive patients classified as dead ones increase the costs considerably. The classifier based on decision tree (DTC) with sensitivity 40.96 and specificity 87.90 outperforms the other one.

4. Conclusions

Open source machine-learning Java software tool Weka was described with focus on classifiers such as a Bayes network classifier and a decision tree classifier. It was employed on our heart failure dataset from Hull LifeLab. For evaluation purposes, two different criteria were considered:

- a) minimization of life-threatening costs;
- b) minimization of treatment costs.

Life-threatening situations appear when patients who are at risk of death soon are considered alive, which is measured by sensitivity. These situations should be minimized and so sensitivity should be maximized. Costs are increased when patients with a low risk of death are treated as if they could die soon, which is measured by specificity. For the costs to be minimized, specificity should be maximized. For our Hull LifeLab data about 2032 patients, a decision tree classifier with sensitivity 40.96% and specificity 87.90% was found, which is significantly better than the other tested classifier (a Bayesian network classifier). In the future it can be useful to consider other classifiers such as a group of fuzzy rules since it would allow us to address vagueness and ambiguity present in the dataset.

Acknowledgment. This work was supported by a HEIF-5 funded project. The authors would like to thank the University of Hull, UK for its support.

References

- [1] BAESENS, B., EGMONT-PETERSEN, M., CASTELO, R., VANTHIENEN, J.: *Learning Bayesian network classifiers for credit scoring using Markov Chain Monte Carlo search*, Proceedings of International Congress on Pattern Recognition, Montreal, Canada, IEEE Computer Society, 2010, pp. 49-52,
- [2] CLINICAL EFFECTIVENESS AND EVALUATION UNIT OF THE ROYAL COLLEGE OF PHYSICIANS: *Managing Chronic Heart Failure: Learning from Best Practice*. The Lavenham Press Ltd, Sudbury, Suffolk, UK, 2005,

- [3] GAROFALAKIS, M., HYUN, D., RASTOGI, R., SHIM, K.: *Building decision trees with constraints*, Data Mining and Knowledge Discovery, Volume: 7, Number: 2, Pages: 187-214, Year: 2003, ISSN: 1384-5810,
- [4] HALL, M., FRANK, E., HOLMES, G., PFAHRINGER, B., REUTEMANN, P., WITTEN, I. H.: *The WEKA data mining software: An update*, ACM SIGKDD Explorations Newsletter, Volume: 11, Issue: 1, Year: 2009, Pages: 10-18, ISSN: 1931-0145,
- [5] LÓPEZ-SENDÓN, J.: *The heart failure epidemic*, Medicographia, Volume: 33, Number: 4, 2011, pp. 363-369,

Contact addresses

Ing. Ján Boháčik, PhD., EUR ING, Department of Computer Science, Faculty of Science, University of Hull, Cottingham Road, Hull, HU6 7RX, United Kingdom and Department of Informatics, Faculty of Management Science and Informatics, Univerzita 8215/1, 010 26 Žilina, Slovak Republic,

E-mail address: J.Bohacik@hull.ac.uk

Dr. C. Kambhampati, Department of Computer Science, Faculty of Science, University of Hull, Cottingham Road, Hull, HU6 7RX, United Kingdom,

E-mail address: C.Kambhampati@hull.ac.uk

Dr. Darryl N. Davis, Department of Computer Science, Faculty of Science, University of Hull, Cottingham Road, Hull, HU6 7RX, United Kingdom,

E-mail address: D.N.Davis@hull.ac.uk

RNDr. Miroslav Benedikovič, Department of Informatics, Faculty of Management Science and Informatics, Univerzita 8215/1, 010 26 Žilina, Slovak Republic,

E-mail address: Miroslav.Benedikovic@fri.uniza.sk

HEURISTICKÉ ALGORITMY PRE MATICOVÉ ROVNOMERNÉ ROZVRHY

ROMAN HAJTMANEK (SK)

Abstrakt. Problém permutovania prvkov v stĺpcoch matice vznikol pri tvorbe rovnomerných rozvrhov prác, keď danému počtu vodičov bolo treba priradiť čo najrovnomernejšie týždenný rozvrh prác. Cieľom tak je dosiahnuť minimálnu mieru nerovnomernosti riadkových súčtov stĺpcovo permutovanej matice. Tento NP-ťažký problém je polynomicky riešiteľný pre dvojstĺpcové reálne matice. Ukážeme, že takýto problém v tomto prípade je polynomicky riešiteľný. Budeme prezentovať dva heuristické algoritmy pre všeobecný prípad problému. Prvý je založený na exaktnom riešení agregovaných dvojstĺpcových úloh. Druhý vychádza z pažravého prístupu priradovania prác. Obe heuristiky majú spoločný náhodný mechanizmus výberu prvkov v nerozhodných situáciách.

Kľúčové slová. rovnomerné rozvrhy, stĺpcovo permutčná matica, stochastický algoritmus.

HEURISTIC ALGORITHMS FOR UNIFORM SCHEDULES OF MATRIX

Abstract. The problem of permuting the columns of matrix elements arose in straight work schedules forming, when was needed to assign weekly work schedule to given number of drivers as evenly as possible. The goal is to achieve a minimum non-uniformity level of column permuted matrix row-sums. This NP-hard problem is polynomial solved for real two-column matrix. We show that such a problem in this case is polynomial solved in this case. We will present two heuristic algorithms for the general case of the problem. The first is based on exact solutions of aggregated columns tasks. The second comes out of greedy attitude of assigning of tasks. Both heuristics have a common random mechanism of elements selection in indecisive situations.

Úvod

Najskôr, na malom ilustračnom príklade popíšeme takéto rozdeľovania práce vodičov vo vozovom parku.

Príklad 0.1. Máme m vozidiel (napr. smetiarske autá) a n pracovných dní. V každý deň je potrebné spraviť m výkonov. Nech sú prvky a_{ij} danej matice A doby výkonov i -teho vodiča v j -ty deň. Poznamenajme, že i -ty vodič jazdí s i -tym vozidlom. Súčet výkonov vodiča i -tého vozidla je rovný $s_i = \sum_{j=1}^n a_{ij}$. Ak je veľký rozdiel medzi celkovými výkonmi jednotlivých vodičov vozidiel s_1, s_2, \dots, s_m , vodiči budú protestovať. Preto zamestnávateľ permutuje priradenie príslušných výkonov v jednotlivých stĺpcoch matice A tak, aby dosiahol čo najmenšie rozdiely medzi nimi.

Tento problém prvý formuloval Peško a časopisecky vyšiel v [2]. Rovnomerné rozvrhy na grafoch boli analyzované v práci [1]. V prípade neistých výkonov bol riešený takýto model v intervalovej aritmetike [3]. Model našiel odozvu aj pri manažérskom rozhodovaní [4].

1. Problém permutácie matíc

Ďalej sa budeme zaoberať nasledujúcim problémom: Pre maticu $A = (a_{ij} \in \mathbb{R})$ typu $m \times n$ a n -ticu $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ permutácií množiny $\{1, 2, \dots, m\}$ nech A^π označuje maticu, ktorá má prvok $a_{\pi_j(i),j}$ v i -tom riadku a j -tom stĺpci. Nech $S^\pi = (s_i^\pi \in \mathbb{R})$ určuje m vektor, ktorého i -ty prvok je i -ta súčet riadku matice A^π :

$$A^\pi = \begin{pmatrix} a_{\pi_1(1),1} & a_{\pi_2(1),2} & \cdots & a_{\pi_n(1),n} \\ a_{\pi_1(2),1} & a_{\pi_2(2),2} & \cdots & a_{\pi_n(2),n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{\pi_1(m),1} & a_{\pi_2(m),2} & \cdots & a_{\pi_n(m),n} \end{pmatrix}, \quad S^\pi = \begin{pmatrix} \sum_{j=1}^n a_{\pi_j(1),j} \\ \vdots \\ \sum_{j=1}^n a_{\pi_j(i),j} \\ \vdots \\ \sum_{j=1}^n a_{\pi_j(m),j} \end{pmatrix}.$$

Problémom permutácie matíc tak rozumieme nasledujúci kombinatorický problém: Je daná matica $A \in \mathbb{R}^{m \times n}$ a reálna funkcia $f : \mathbb{R}^m \rightarrow \mathbb{R}$

$$\min \left\{ f(s_1^\pi, s_2^\pi, \dots, s_m^\pi) : \pi = (\pi_1, \pi_2, \dots, \pi_n) \in \Pi, s_i^\pi = \sum_{j=1}^n a_{\pi_j(i),j} \right\},$$

kde f je nejaká miera nerovnomernosti vektora S^π a Π je množina všetkých permutácií množiny $\{1, \dots, n\}$.

Nech množina $I = \{1, 2, \dots, m\}$ označuje riadkové a $J = \{1, 2, \dots, n\}$ stĺpcové indexy matice. Ďalej sa budeme zaoberať len tromi praktickými mierami nerovnomernosti vektorov:

$$f_{dif}(x_1, \dots, x_m) = \max_{i \in I} \{x_i\} - \min_{i \in I} \{x_i\},$$

$$f_{abs}(x_1, \dots, x_m) = \max_{i \in I} \{|x_i - s|\}, \quad f_{sqr}(x_1, \dots, x_m) = \sum_{i \in I} (x_i - s)^2,$$

kde $s = \frac{1}{m} \sum_{i \in I} \sum_{j \in J} a_{ij}$ je priemerný (želaný) riadkový súčet. Jednoduchá miera f_{dif} je hrubá a tak bola zovšeobecnená na citlivejšiu mieru f_{abs} . Ešte citlivejšia je však miera nerovnomernosti f_{sqr} . My sa budeme zaoberať len najjednoduchšou mierou f_{dif} .

2. Dvojtĺpcový prípad

Uvažujme dvojtĺpcovú maticu A typu $m \times 2$. Naším cieľom je nájsť takú permutáciu π riadkov matice:

$$A^\pi = \begin{pmatrix} a_{1,1} & a_{\pi(1),2} \\ a_{2,1} & a_{\pi(2),2} \\ \vdots & \vdots \\ a_{m,1} & a_{\pi(m),2} \end{pmatrix},$$

ktorá minimalizuje hodnotu funkcie f_{dif} :

$$f_{dif}(s_1^\pi, \dots, s_m^\pi) = \max_{i \in I} \{s_i^\pi\} - \min_{i \in I} \{s_i^\pi\}, \quad s_i^\pi = \sum_{j=1}^n a_{\pi_j(i),j}, \text{ pre } i \in I.$$

Poznamenajme, že prvý stĺpec matice nemusíme permutovať. Bude výhodné ho mať v A utriedený vzostupne. Potom totiž stačí utriediť prvky druhého stĺpca zostupne a máme optimálne riešenie v polynomiálnom čase $O(m \log_2 m)$ [2].

Vyššie uvedý postup demoštrujeme na príklade.

Príklad 2.1. Uvažujme matice:

$$A = \begin{pmatrix} 1 & 6 \\ 2 & 8 \\ 3 & 7 \\ 4 & 9 \end{pmatrix}, \quad S = \begin{pmatrix} 7 \\ 9 \\ 11 \\ 13 \end{pmatrix},$$

kde S je matica riadkových súčtov matice A s hodnotou $f_{dif}(S) = 5$.

Vytvoríme permutáciu $\pi = (4, 2, 3, 1)$ druhého stĺpca matice tak, aby jej hodnoty boli utriedené zostupne. A tak dostaneme optimálne permutovanú maticu A^π s minimálnou hodnotou $f_{dif}(S^\pi) = 10$:

$$A^\pi = \begin{pmatrix} 1 & 9 \\ 2 & 8 \\ 3 & 7 \\ 4 & 6 \end{pmatrix}, \quad S = \begin{pmatrix} 10 \\ 10 \\ 10 \\ 10 \end{pmatrix}.$$

3. Agregáčn  heuristika

Agregačná heuristika sa používa na riešenie inštancií s väčším počtom riadkov stĺpcov ($n \geq 2$) pomocou nasledovnej dvojtĺpcovej heuristiky:

Krok 1: Začneme s $m \times n$ maticou A a n -ticou $\pi = (\pi_1, \pi_2, \dots, \pi_n)$ identických permutácií.

Krok 2: Náhodne vyberieme neprázdnu podmnožinu množiny stĺpcových indexov J , $\emptyset \subset J \subset \{1, 2, \dots, n\}$ a definujeme jej doplnok $K = \{1, 2, \dots, n\} - J$.

Krok 3: Vyriešime dvojstĺpcovú inštanciu úlohy s $m \times 2$ maticou $B = (b_{ij})$, pričom:

$$b_{i1} = \sum_{j \in J} a_{\pi_j(i)j}, \quad b_{i2} = \sum_{j \in K} a_{\pi_j(i)j}$$

Riešenie získame tak, že prvý stĺpec matice B utriedime vzostupne a druhý zostupne. Príslušne permutujeme zodpovedajúce množiny stĺpcov J a K , čím dostaneme riešenie, ktoré nie je horšie než východiskové.

Krok 4: Ak nejaké kritérium ukončenia je splnené (max. počet krokov, doba výpočtu, atď) STOP inak GOTO **Krok 2**.

V prípade kroku **3** sa v nerozhodných situáciách riadky náhodne zamiešajú. Výpočtové experimenty so známymi optimálnymi riešeniami ukazujú, že táto zlepšujúca heuristika je prekvapujúco kvalitná.

4. Pažravá heuristika

Tento postup zrovnomenenia riadkových súčtov matice spočíva v pažravom (greedy) princípe, keď jednotlivé hodnoty matice A postupne premiestňujeme do matice A^π od najväčších hodnôt po najmenšie. Budeme sa riadiť pravidlom, že pridávame vždy do toho riadku, ktorý má minimálny aktuálny riadkový súčet. V prípade nejednoznačnosti vyberieme riadok náhodne.

Túto tvoriacu heuristiku môžeme formalizovať nasledovne:

Krok 1: Vytvoríme dvojstĺpcovú maticu T typu $m \cdot n \times 2$ z matice $A = (a_{ij})$:

$$t_{k1} = a_{ij}, \quad t_{k2} = j, \quad k = (j - 1) \cdot n + i.$$

Krok 2: Vynulujeme maticu $B = (b_{ij})$, $b_{ij} = 0$ typu $m \times n$.

Krok 3: for ($k = 1, 1, m \cdot n$) begin

Nájdí index $i^* \in I$ pre $j = t_{k,2}$ taký, že $s_{i^*} = \min_{i \in I} \{s_i : b_{i,j} = 0\}$,
 polož $b_{i^*,j} = t_{k,1}$ end.

Krok 4: Polož $A^\pi = B$, STOP.

5. Záver

Oba prezentované prístupy k riešeniu problému tvorby rovnomerného maticového rozvrhu sa ukazujú byť životaschopné aj napriek tomu, že ide o riešenie NP-ťažkého problému. Navyše umožňujú doplnenie o ďalšie praktické obmedzujúce podmienky napr. deň pracovného voľna, pravidelné prestávky na údržbu vozidiel, ap.

Vytvorenú pažravú heuristiku možno použiť ako štartovacie riešenie pre agregáciu heuristiku. Neprekvapil by nás ešte častejší výskyt optimálneho riešenia.

V ďalšej práci sa chceme sústrediť na modelovanie rozvrhov pri neurčitosti veľkosti pracovných výkonov. Cesta sa črtá v stochastickom programovaní, kde by

bolo možné získať z reálnych dát príslušné charakteristiky štatistickými metódami, prípadne s využitím fuzzy resp. drsných množín.

PodĎakovanie

Práca vznikla s podporou grantu VEGA 1/0374/11: Modelovanie a optimalizácia mobility a infraštruktúry v logistických sieťach, (2011-2013)

Literatúra

- [1] Czimmermann, P. and Peško, Š.: *The regular permutation scheduling on graph*, Journal of information control and management system, Vol. 1, pp. 15–22, 2000.
- [2] Černý, J.: *How to minimize non-uniformity* [in Slovak], PMFA XXX, 145–150, 1985.
- [3] Peško, Š.: *The Matrix Permutation Problem in Interval Arithmetic*, Journal of Information, Control and Management Systems, Faculty of Management Science and Informatics, University of Žilina, Vol.4, No.1, pp 35-40, ISSN 1336–1716, 2006.
- [4] Peško, Š. Černý, J.: *Uniform Splitting in Managerial Decision Making*, E+M, Economics and Management, 4/2006, IX, pp. 67–71, Technická univerzita v Liberci, ISSN 1212-3609, 2006.

Kontaktná adresa

Ing. Roman Hajtmanek, Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská Republika,

E-mailová adresa: `roman.hajtmanek@fri.uniza.sk`

AKCELEROMETRICKÉ MERANIE VÝSTRELU Z LUKU

MICHAL CHOVANEC (SK) A JAROSLAV MŮČKA (SK)

Abstrakt. Článok popisuje využitie miniatúrneho akcelerometra na meranie dynamickej vlastností šípu. Meranie umožňuje stanoviť rýchlosť, zrýchlenie a dobu výstrelu šípu. Uvedené údaje je možné použiť pri správnom výbere šípu, prípadne presné stanovenie kinetickej energie pre lovcov s lukom. Riešenie je implementované v mikrokontroléry msp430, naprogramovanom s využitím msp430-gcc kompilátora. Aplikácia spracúvajúca údaje v PC je napísaná v jazyku Ruby.

Kľúčové slová. akcelerometer, šíp, msp430, ruby.

ACCELEROMETRICS MEASURING BOW SHOT

Abstract. This article shows an application of a miniature accelerometer for the measurement of dynamic arrow properties used in archery. The measurements allows the determination of the arrow's acceleration, velocity and flight trajectory. The recieved measurements can be used when choosing the correct arrows, or for an accurate designation of the arrow's kinetic energy, when the bow is used for hunting. The design uses msp430 microcontrolers programmed using the msp430-gcc compiler. The application for data processing is written in Ruby.

Keywords. accelerometer, arrow, msp430, ruby.

Úvod

V modernej lukostreľbe je výberu šípu venovaná veľká pozornosť. Základným faktorom, ktorý ovplyvňuje správanie sa šípu, je spine. Spine je veľkosť ohybu šípu dlhého 28 palcov (71.12 cm), pri zaťažení váhou 1.94 lb (880 g). Vypustenie šípu je doprevádzané sklznutím prstov z tetivy. Tento šikmý pohyb rozkmitá tetivu v osi kolmej na jej smer pohybu [1]. Je zrejmé, že kontakt šípu s lukom nepriaznivo ovplyvňuje jeho dráhu. Voľba správneho spinu teda znamená vybrať takú tuhosť, ktorá optimálne v oblúku minie luk. Príliš malá tuhosť šípu (veľký spine) môžu viesť ku zlomeniu šípu v luku, poškodeniu luku alebo vážnemu zraneniu. Naopak, príliš tuhý šíp (malý spine) príde do kontaktu s lukom a dôjde k výraznému zhoršeniu presnosti.

Meranie zrýchlenia umožňuje detailne stanoviť priebeh výstrelu a všetky jeho fázy. Za týmto účelom bol zhotovený miniatúrny modul trojosého akcelerometra. Dáta spracúva nízkošpotrebný mikrokontrolér msp430. Ako úložisko dát bola zvolená FRAM pamäť.

1. Modul akcelerometra

Modul sa skladá z troch častí:

akcelerometer: ADXL377 (rozsah 200 g) výrobca Analog Devices,

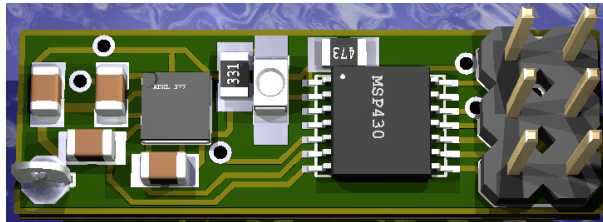
mikrokontrolér: MSP430G2452 (16 bit, 16 MHz, 8 kB flash) výrobca Texas Instruments,

pamäť: FM24CL64 (64 kbit, 1 MHz) výrobca Ramtron.

Na všetky tri prvky sú kladené prísne požiadavky. Akcelerometer musí mať požadovaný rozsah. Výkon mikrokontroléra musí byť dostatočný na digitálnu filtráciu, ukladanie a komunikáciu s PC. Od pamäte sa očakáva adresovateľnosť po bytoch a možnosť okamžitého zápisu.

Všetky elementy musia byť čo najmenšie a vystačiť s napájaním 3,3 V, pri čo najmenšom odbere prúdu. Napájanie zabezpečuje miniatúrna li-pol batéria. Zohľadnením týchto požiadavok sa výber súčiastok obmedzí na minimum.

Celkový pohľad na rozloženie prvkov na doske znázorňuje obrázok. Pamäť FRAM je umiestnená zospodu.



Obr. 1. 3D model akcelerometra

Na konektore je k dispozícii napájanie, programovacie rozhrania UART komunikácia s počítačom. Firmware je tak možné dodatočne upravovať. Zmenou vzorkovacej periódy je možné merať celkovú dráhu letu šípu, alebo len okamih tesne po výstrele. Akcelerometer umožňuje vzorkovať maximálne s frekvenciou 1000 Hz, čo zodpovedá 1ms intervalom. Kapacita pamäte je 8192 bytov. To vystačí na 2048 záznamov.

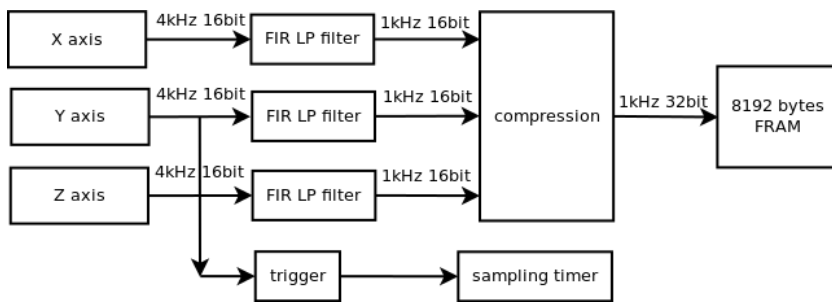
Doska spolu s li-pol akumulátorom boli umiestnené v plastovej trubičke. Lovecké šípy umožňujú osadiť šrobovací hrot. Celý modul bol preto uchytený práve týmto spôsobom – šíp bez hrotu musí byť strieľaný na veľmi mäkký terč (ťažký záves).

2. Programové vybavenie

Firmware

Firmware pre mikrokontrolér je skompilovaný pomocou msp430-gcc kompilátora. Na nahratie programu do flash pamäte slúži doska Texas Instruments Launchpad [2].

Samotný program vykonáva niekoľko dielčích funkcií. Jednotlivé funkčné bloky znázorňuje obrázok.



Obr. 2. Znázornenie funkčných blokov firmware

Po reštarte a inicializácii hardvéru, program čaká na spustenie náhlym zrýchlením. Časovač sa nastaví na periódu 4 kHz (3906,25 Hz). Vždy pri výskyte prerušenia, sa postupne pomocou 10 bitového AD prevodníka prečítajú všetky 3 osy akcelerometra.

Nasleduje banka FIR filtrov 4. rádu. Tie sú tu z dôvodu vyhladenia dát pred prípadnými chybami.

Diferenčná rovnica popisujúca filtre má tvar:

$$y(n) = 0,25(x_n + x_{n-1} + x_{n-2} + x_{n-3}).$$

Je to jednoduchý plávajúci priemer. Pre optimalizáciu rýchlosti sú v programe namiesto násobení použité bitové posuny. Výstup z filtrov je 16 bitové číslo. Z toho ale len dolných 10 bitov je platných. Zvyšné sú nastavené na nulu. Vhodným premapovaním 3×10 bitov (3×2 byty) do 30 bitov (4 byty) je možné ušetriť miesto vo FRAM pamäti. Celková úspora miesta je potom 33%. Do pamäte tak vojde 2048 záznamov – asi 2 sekundy letu. Ukladá sa každá štvrtá vzorka, tak aby frekvencia bola 976,56 Hz.

Po ukončení merania (vyprázdnení pamäte) akcelerometer čaká na príjem ľubovoľného znaku z UARTu. Po jeho príjme začne vysielat dáta z pamäte po sériovej linke.

PC aplikácia

Program vyhodnocujúci dáta na strane PC je napísaný v jazyku Ruby. Transformuje hexadecimálny vstup na napätia akcelerometra. Taktiež sa overuje CRC. Prípadnú nezhodu program zaznamená.

Práca so sériovým portom je v Ruby veľmi jednoduchá:

```
@sp      = SerialPort.new "/dev/ttyUSB0", 9600; #inicializácia
c        = sp.getc.ord; #príjem znaku
@sp.write s;          #vyslanie znaku
```

Z dôvodu nedostupnosti sériového portu v dnešných počítačoch, bol použitý konvertor USB->UART s obvodom FT232 [3]. V systéme Linux však nie je potrebné inštalovať žiadne dodatočné ovládače.

Z akcelerometrického modulu počítač získava dáta v surovej podobe. Najprv je potrebné vypočítať napätie:

$$U_{acc} = U_{ref} \frac{N}{1024}.$$

Z napätia a z údajov v katalógovom liste [4] je možné odvodiť zrýchlenie:

$$a = 9,81 \frac{U_{acc} - U_{ofs}}{U_{sens}} \text{ ms}^{-2}.$$

Kde U_{ofs} je offset akcelerometra v kľudovom stave, obvykle polovica napájacieho napätia. U_{sens} je citlivosť akcelerometra, výrobca udáva 6,6 mV/g pre os Y.

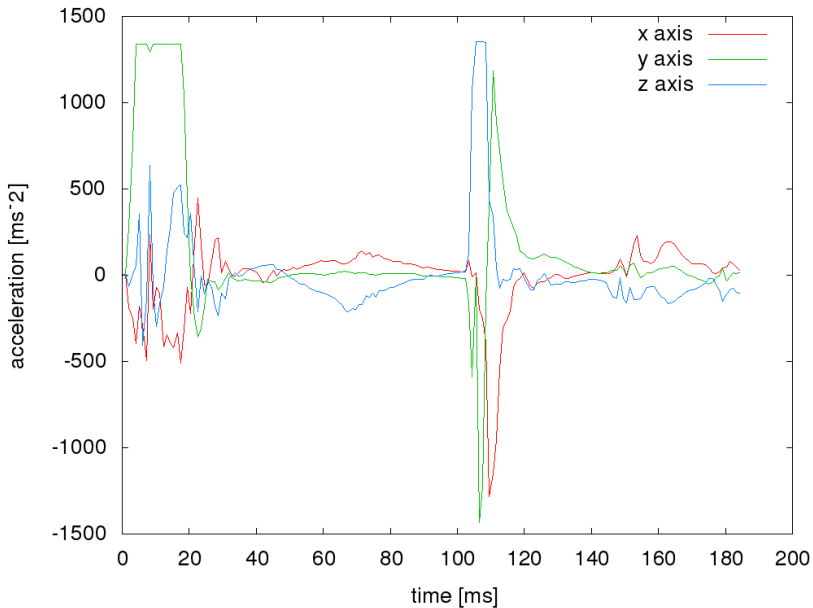
3. Merania

Meranie bolo uskutočnené s lukom so stredom Hoyt Horizon s karbónovými ramenami Samick Vision o sile 42 lb. Šíp bol zvolený Easton Powerflight, spine 500 [5]. Dráha letu bola 4,9 m. Os Y akcelerometra zodpovedá smeru letu šípu. Os Z je kolmá na smer letu a znázorňuje horizontálne pôsobenie síl.

Na grafe je viditeľný priebeh výstrelu. Samotný let šípu a jeho náraz do voľne zaveseného ťažkého závesu. Jeden dielik na časovej osi zodpovedá času 1 ms. Je zrejmé, že došlo k prekročeniu rozsahu akcelerometra v smere osi Y. Maximálne namerané zrýchlenie bolo $1337,94 \text{ ms}^{-2}$, čo zodpovedá 136,4 g. Orežanie dát teda spôsobil rozsah AD prevodníka. Napätie viac ako $U_{ref} = 2,5 \text{ V}$ už nedokázal spracovať. Pre maximálne preťaženie 200 g vychádza výstupné napätie 2,9186 V. Taktiež dáta v okamihu nárazu sú značne skreslené, najmä kvôli nedostatočnej frekvencii vzorkovania a meraciemu rozsahu. Treba poznamenať, že z dostupných MEMS akcelerometrov bol vybraný jeden z najlepších dostupných.

Z doby letu a známej vzdialenosti je možné určiť rýchlosť šípu. Skutočný čas dostaneme prepočítaním vzhľadom na vzorkovaciu frekvenciu:

$$T = \frac{X}{976,25 \text{ Hz}}.$$



Obr. 3. Namierané zrýchlenie v jednotlivých osiach

Čas a rýchlosť letu:

$$t = (T_{\text{impact}} - T_{\text{start}}), \quad v = \frac{s}{t}.$$

Po dosadení dostaneme rýchlosť šípu

$$v = 54,35 \text{ms}^{-1}.$$

Z grafu je ďalej možné odčítať dobu trvania samotného výstrelu. Vychádza 20 ms. Šípu teda muselo byť udelené priemerné zýchlenie:

$$\bar{a} = \frac{54,35 \text{ms}^{-1}}{g \times 20 \times 10^{-3} \text{s}} = 282 \text{g}.$$

Z hmotnosti šípu 26 g vychádza kinetická energia 38 J.

Veľmi hrubo je možné odhadnúť energiu uloženú v napnutom luku za predpokladu lineárneho nárastu sily:

$$E = 0,5Fs.$$

Po dosadení:

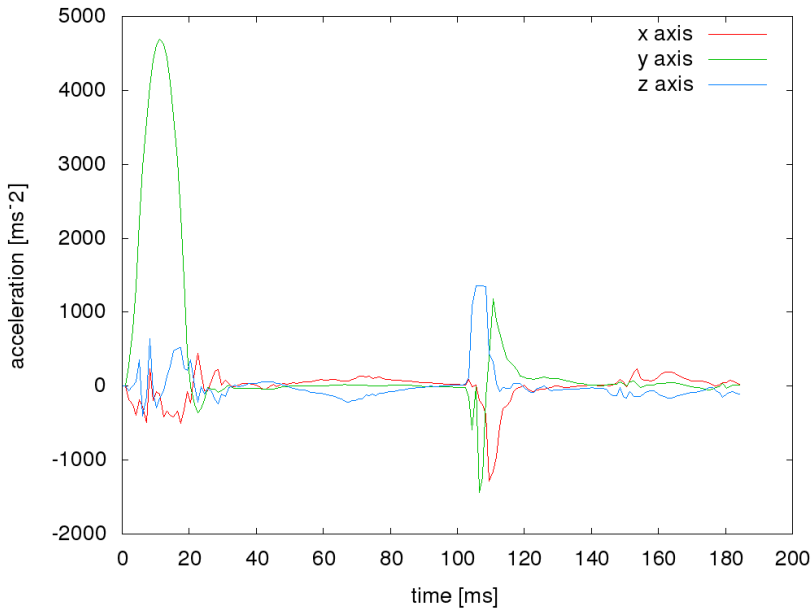
$$E = 0,5 \times 190 \text{ N} \times 0,525 \text{ m} = 49,875 \text{ J}.$$

Kde s je dĺžka pracovného náťahu luku a F je sila potrebná na napnutie. Treba poznamenať, že určitá časť energie sa spotrebuje na samotný pohyb ramien. Po jeho ukončení sa lukom ešte šíria zvyškové vibrácie, ktoré stlmí stabilizátor.



Obr. 4. Testovaný šíp Easton Powerflight

Z dôvodu ďalšej analýzy, je možné oblasť prekročeného rozsahu akcelerometra aproximovať preložením hladkej krivky. Princiipiálne nezáleží na jej voľbe, preto bola zvolená parabola. Z bodov začiatku a konca aproximačného intervalu a známej plochy pod touto krivkou – zodpovedá výstrelnej rýchlosti je možné určiť predpis paraboly.



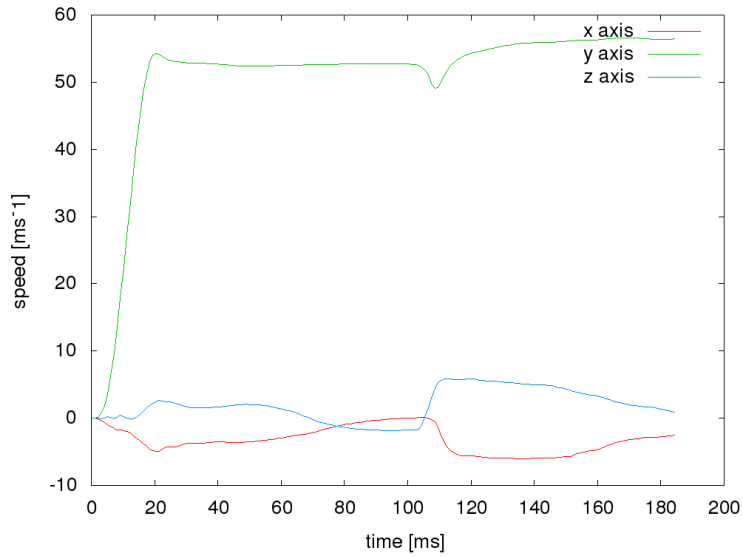
Obr. 5. Aproximácia zrýchlenia

Jednoduchou integráciou je možné vyjadriť rýchlosť. Táto skutočnosť vyplýva z druhého Newtonovho zákona. Dvojitou integráciou sa dá overiť prejdená trajektória. Funkcie sú dané diskretnými bodmi. Na aproximáciu integrácie bola použitá lichobežníková metóda integrovania

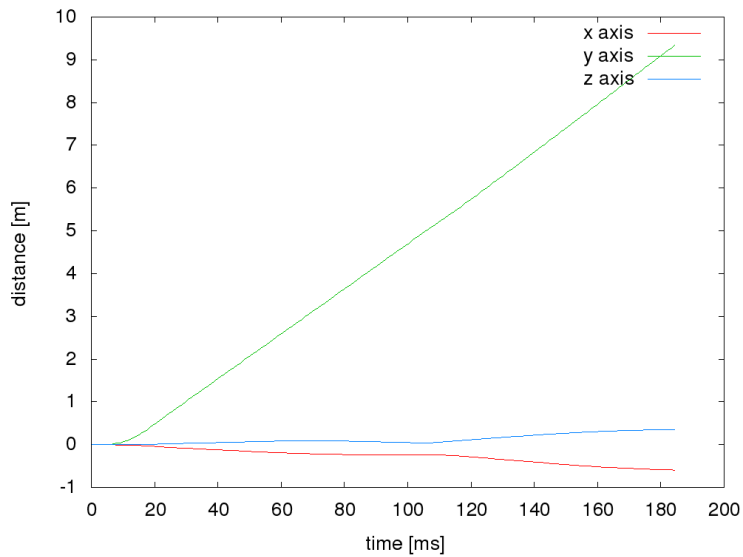
$$v(t) = \int_0^t a(u) du \approx \sum_{i=0}^n \frac{a(t_i) + a(t_{i-1})}{2},$$

$$s(t) = \int_0^t v(u) du \approx \sum_{i=0}^n \frac{v(t_i) + v(t_{i-1})}{2}.$$

Kde du zodpovedá vzorkovacej perióde, $v(t)$ je rýchlosť v čase t , $s(t)$ je prejdená dráha v čase t .



Obr. 6. Dosaiahnutá rýchlosť



Obr. 7. Prejdená dráha

4. Záver

Merania ukázali pôsobenie síl na šíp. Podľa očakávaní, dosialo zrýchlenie špičkovy stovky g . Rozsah použitého akcelerometra ($\pm 200 g$) sa ukázal ako nedostačujúci. Z dostupných MEMS akcelerometrov sa však zatiaľ nevyrába akcelerometer, ktorý by tomuto účelu vyhovoval. Najlepší sa javí ADXL180, ktorý poskytuje rozsah až $500 g$. Má však len jednu os a vzorkovanie iba 800 Hz . Zvolený typ ADXL377 sa preto v najbližšom horizonte javí ako najlepší kompromis: rozsah $200 g$, 3 osi, vzorkovanie $1,6 \text{ kHz}$.

Ako riešenie do budúcnosti, je najlepšie zvýšiť merací rozsah AD prevodníka a jeho bitové rozlíšenie. Zaujímavý mikrokontrolér vyhovujúci tomuto predpokladu je MSP430AFE253. Obsahuje tri 24bitové sigma-delta prevodníky, primárne určený pre trojfázové elektromery. S využitím tak vysokej rozlišovacej schopnosti by bolo možné monitorovať celú dráhu letu, vrátane vybráčení šípu.

Namerané údaje poskytujú obraz o dobe trvania výstrelu a rýchlosti šípu. Z množstva kinetickej energie je možné stanoviť vhodnosť použitia luku na lov podľa [6]. Výsledky meraní vrátane zdrojových súborov programov je možné stiahnuť na adrese [7].

Literatúra

- [1] Vypustenie šípu: <http://www.youtube.com/watch?v=wGNslUNBrEM>.
- [2] Vývojová doska launchpad: <http://www.ti.com/tool/msp-exp430g2>.
- [3] Obvod FT232: <http://www.ftdichip.com/Products/ICs/FT232BM.htm>.
- [4] Akcelerometer ADXL377: <http://www.analog.com/en/mems-sensors/mems-accelerometers/adxl377/products/product.html>.
- [5] Šíp powerflight <http://eastonarchery.com/arrows/carbon-powerflight/0/2>.
- [6] Energia šípu http://eastonarchery.com/store/kinetic_calculator.
- [7] Výsledky meraní <http://sourceforge.net/projects/msp430files/>.

Kontaktné adresy

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

Jaroslav Múčka, Katedra technickej kybernetiky, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská Republika,
E-mailová adresa: olololwtfomg@gmail.com

LOGGING TOOL — THE LOG4J

JOZEF KOSTOLNY (SK)

Abstract. Logging and tracing is the essential part of debugging of applications. The article deals with advanced Log and Trace system **Log4j**. There are listed basic logging and tracing operations troubles and main reasons for usage of the advanced logging systems. The rest of the article describes the **Log4j** system itself, its components, features, basic commands and their parameters with a description, the configuration file and examples of using this system.

Key words and phrases. Log4J, log, trace.

LOGOVACI NASTROJ LOG4J

Abstrakt. Logovanie a sledovanie činností aplikácií patrí medzi neodmysliteľné súčasti debugovania. Článok sa venuje pokročilému systému pre logovanie a sledovanie **Log4j**. Sú tu uvedené základné úskalia logovania činností aplikácií a od nich odvodené dôvody potreby pre používanie pokročilých logovacích systémov. Zvyšok článku popisuje samotný **Log4j** systém, jeho súčasti, vlastnosti, základné príkazy a ich parametre s popisom, konfiguračný súbor systému **Log4j** a príklady použitia tohto systému.

Kľúčové slová. Log4J, Logovanie, sledovanie.

What is “log” and “trace”?

It is a mechanism that allows writing the informations in the course of program in order to allow later analysis of events that occurred during run time, to allow determination of the impact of events arising in the operations and possibly to allow determination of the variables. The vast majority of us after writing the program “Hello world!”, but before acquainting with debugger, most often use drawing text into console for debugging their programs. After acquaintance with such powerful instrument as a debugger is, most of us immediately forget what is the function `printf()` and recording errors. As will be shown, this way is wrong and it allows only continuous monitoring of program’s behavior it doesn’t allow long-term analysis and debugging.

What is the difference between the logging and tracing? Logging is designed to record the program’s events and the tracing has the task of writing the program course and values of variables. This distinction is purely logical and for both activities is used a library that meets both purposes simultaneously, physically writing to management.

What is Log and Trace good for us?

As we mentioned, as quickly as we enjoy recording and monitoring the program, as quickly we forget about the existence of a debugger for use with continuous error detection. That way in many cases removes the possibility of later finding serious errors in program design. For example, during the development of software for embeded systems operating in other than standard PC architecture, it often happens that we have no way of debugging. Therefore, the projects in which part of the test is conducted on the client side, where the testers do not see the source code, well-written set of logs is more valuable than any description of the problem made by tester. Due to the complexity of current applications, errors which are not repeated regularly occur but only upon the occurrence of certain, unknown circumstances. If attempts to call such errors again are unsuccessful, well-designed set of logs from the first occurrence of the error is the only way to find it and fix it.

How can we use recording and monitoring the program course?

We propose to record all changes in the application or module (e.g. start of operation, initialization, ready state or deinitialization). As a result, we can easily prevent that our module will not respond to external commands, if we have checked the information that it has not been initialized by the user. We should also record all the critical errors, which make impossible to normal continuation of the program. Such information allows us to immediately focus on the problem with no matter what its other effects are described by the user (see Figure 1).

We can also write down all important events in the module, such as it is call its interfaces, which will help us to determine whether the module was responsible for system malfunction or even if it was not used.

Many more problems are found during the program tracking. On one side is good to have the track of all input and output functions and registration of parameters and the values of their internal decision conditions (if-else), but on the other side is a lot of functions called hundreds to thousands times while running the program, which would in recording cause a reduction of speed and it could lead to the fact, that the log file would be unreadable in the analysis. Therefore, the functions should be induced to write just such a track, which is unusual in its implementation and may indicate the occurrence of the problem.

INTRODUCTION TO LOG4J USE

We have written already about the philosophy of the use of logging and tracking progress of the program and now we try to explain how Log4j library works in practice. Log4j has three parts: Logger, Appender and Layout (see Figure 2). The objects of these three classes are necessary to create a tie among them to make

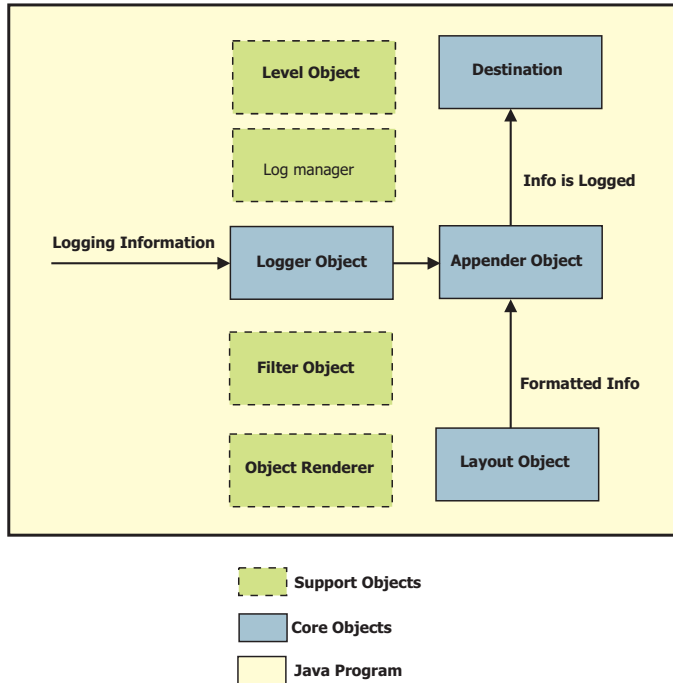


Figure 1. Logging method.

full use of services Log and Trace. Logger class object we use for report program events or theirs tracks, Layout class object we use to adjust the formatting of these reports and Appender class object we use to create a repository for writing reports. Objects of described classes we can freely connect, such the report from one Logger can be redirected into different Appender class objects and in every case it will be formatted differently. Moreover, properties of those objects can be adjusted in the program's source code or by configuration files also in the program course. This simple but enormous flexibility resulting from the modular architecture of the library in conjunction with the availability of the Apache license 2.0, leads to the fact that it is essentially an ideal solution. Let's take a closer look at each of the three described components.

LOGGER

Logger class object, which allows writing the reports, can be obtained by inducing a static method of Logger class by sending an object name that we need. The name can be prefixed with names consisting of ancestors, separated by dots:

```
Logger mylogger = Logger.getLogger('TestApp.class');
```

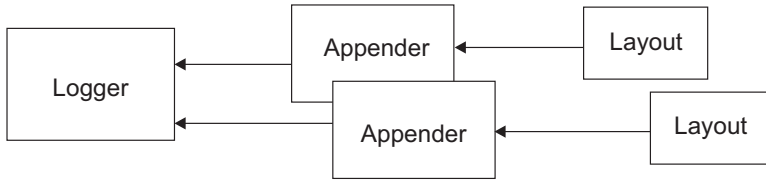


Figure 2. Log4j components.

In the example we created a Logger class object called “TestApp.clas1”, which will inherit all the settings from the object named “TestApp”, and again this Logger object will inherit settings of the main Logger object, which is obtained by calling the function:

```
Logger mylogger = Logger.getRootLogger();
```

Each object will inherit the settings of their predecessors, if we do not set it otherwise. Each Logger enables a lot of features, which allow logging of messages with different priorities: TRACE, DEBUG, INFO, WARN, ERROR, FATAL (see Figure 3). Writing messages e.g. with priority INFO will be redirected to an Appender object only in case when the Logger has set the logging level with priority INFO or higher. This means WARN, ERROR or FATAL. We can also set the level ALL or OFF, which means logging of all messages (priority ALL) or don’t log any messages (priority OFF).

This simple mechanism and the ability to set priorities for all inheriting levels of Logger objects from a configuration file is an important attribute of a system for logging and tracing. Moreover, the function **getLogger()** does not create a new object, but only returns an object with a certain name, so we can easily get access to the same object in different places of the source code. In addition, by updating the configuration file we can change settings of an object, which the program has not have created yet, but for which there is reference in memory and thus it can then be obtained by its name.

LAYOUT

Layout class object is responsible for formatting the output message. These objects are attributed to each Appender object, which allows special formatting of the output. There are many implementations of the abstract Layout class. Some convert message into HTML table format (HTML-Layout), other record logs into XML format (XML-Layout). The most useful implementation is the Enhanced-Pattern layout, which allows its own definition of a formula by which messages are formatted (see Figure 4). The Layout we create by the following manner:

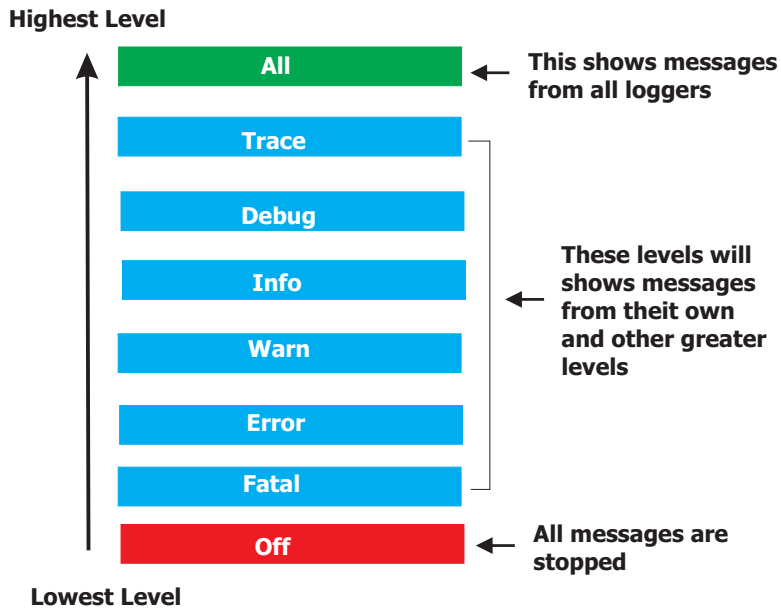


Figure 3. Log4j Levels – Preferences.

```
Layout myLayout = EnhancedPatternLayout(conversionString);
```

As `conversionString` we enter sequence of characters that defines the conversion. For example: “`\\%r \\%l [\\%t]: \\%-5p \\%m\\%n`”.

The characters mean:

- `\\%r` – Time in milliseconds that has elapsed since the creation of an object of Layout class to creating record for this report.
- `\\%l` – Name of the function, with the full path of classes, name of the file and line number where the report was generated.
- `\\%p` – Message priority. Modifier “5” means alignment to the right for five characters minimally.
- “_” – Means a change of alignment to the left.
- `\\%m` – Message is forwarded to the logging function.
- `\\%n` – Newline sign.

Report formatted by that Layout that can look like this:

```
0 logandtracetest.Main.main(Main.java:39)
[main]: INFO MessageText 1
110 logandtracetest.Main.main(Main.java:43)
[main]: INFO MessageText2
```

More conversion characters of formula and their modifiers can be found in the JavaDoc documentation, which is in the official Log4j website.

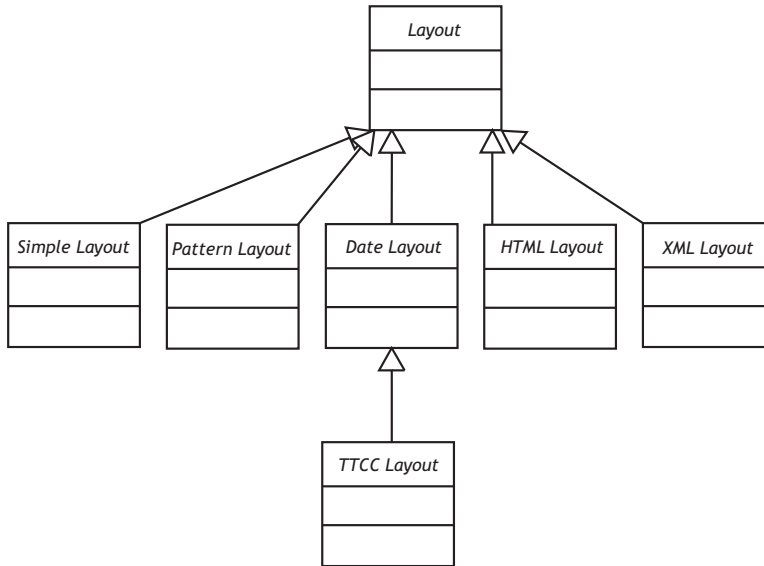


Figure 4. Layout in Log4j.

APPENDER

The last element is the Appender, which is the target object of sending messages. This means that this class has several different versions. Accessible are Appenders that record messages into a file (see Figure 5), into different databases, by messages to the console or send over the network to any port or by using different protocols (e.g. telnet). There is also an Appender, which allows sending messages via email! Details of the Appender configuration are in JavaDoc paperwork at Log4j project website. The most commonly used Appender is probably FileAppender, which can be created by the following manner:

```

FileAppender myFileAppender;
myFileAppender = new FileAppender( myLayout,
//Reporting format for given appender
'log.txt', //Path to the file
true,
//Parameter of the appender, if value is set to true,
//other logs will be wrote to he end of existing file
true,
//Parameter Buffered-IO, if value is set to true,

```

```
//write to the file is buffered
256); //Buffer size
```

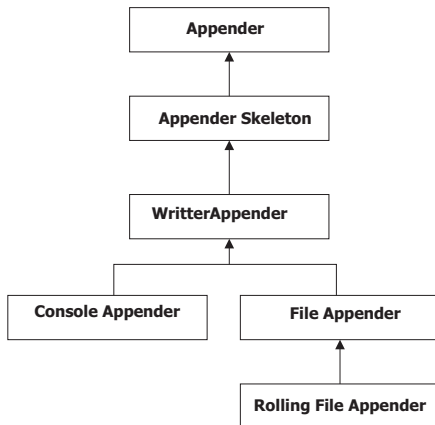


Figure 5. Appender.

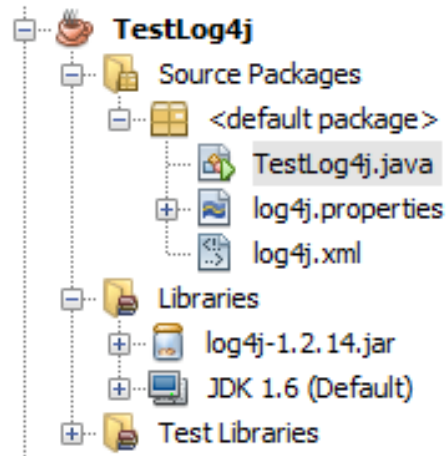


Figure 6. Structure of the Log4j project.

CONFIGURATION FILE

As we see, with the right combination of objects of classes `Logger`, `Appender` and `Layout` we can in any way and anywhere write any information (see Figure 6). Unfortunately, it really does not help us, if the information amount is so great that we will not be able to effectively analyze them. Therefore, the right balance between informations that we collect and informations that we write with `Log` and `Trace` is important. The solution to this problem is a dynamic configuration, which can be adapted during the program course, depending on our needs. For example, if the program works normally, we write only messages with priority `ERROR` and if there such message occurs (with priority `ERROR`), change the configuration of logging messages in this module so we have access to any information that will enable the removal of errors. This can be done by calling such function inside the code:

```
PropertyConfigurator.configure('configFile.txt');
```

By calling the above introduced function, all of the objects described in the configuration file change their properties according to the description. The exact description of the commands used in the configuration file is located at the project `Log4j` webpage. Commands are very intuitive, which shows the example below:

```
# We set the level of rootLogger to DEBUG a assign to it Appender A1
log4j.rootLogger=DEBUG, A1
```

```
# We set Appender A1 as ConsoleAppender
log4j.appender.A1=org.apache.log4j.ConsoleAppen -der
# We set Layout for Appender A1 to Pattern-Layout
log4j.appender.A1.layout=org.apache.log4j.Pat -ternLayout
# We set the conversion formula
log4j.appender.A1.layout.ConversionPattern=
%-4r[%t] %-5p %c %x - %m%n
```

Conclusions

If we have all three elements and we know what our goal is, we can create a very effective tool for debugging and maintenance of our applications. By employing the configuration file, we can be sure that we get as much information as is needed and yet not reduce application performance. We hope that these few brief information allows you to save a lot of hours searching hidden errors in your code. We refer interested readers to the project site Log4j [1].

References

[1] Log4J project <http://logging.apache.org/log4j/1.2>,

Contact address

Ing. Jozef Kostolny, Department of Informatics, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic,

E-mail address: Jozef.Kostolny@fri.uniza.sk

REDIS — NON-RELATION DATABASE SYSTEM

MARIAN KOVÁČ (SK)

Abstract. The article describes widely used non-relational „key-value” type database system Redis. There is provided a brief history of the supported data structures, described how to install Redis on Linux and Windows, how to run Redis on these systems and the configuration options using different parameters. Next part of the article is devoted to Redis command line in detail and to basic commands, where is always given a brief description of each command. The end of the article summarizes the advantages and disadvantages of nosql solution Redis.

Key words and phrases. Redis, database system, nosql.

REDIS — NERELAČNÝ DATABÁZOVÝ SYSTÉM

Abstrakt. Tento článok popisuje najpožívanejší nerelačný databázový systém typu „kľúč-hodnota” Redis. Je tu uvedená jeho stručná história a podporované štruktúry údajov. Ďalej článok uvádza spôsoby inštalácie systému Redis na systémoch Linux a Windows, spustenie Redisu na týchto systémoch a možnosti konfigurácie pomocou rôznych parametrov. Ďalšia časť článku sa podrobnejšie venuje príkazovému riadku na ovládanie Redisu a jednotlivým základným príkazom, pri ktorých je vždy uvedený ich stručný popis. V závere článku sú zhrnuté výhody a nevýhody nosql riešenia Redis.

Kľúčové slová. Redis, databázový systém, nosql.

1. History

Redis project (home page: <http://www.redis.io>) was established in late 2009 by an Italian programmer Salvatore Sanfilippo. The initial target was to improve performance of data analysis in the real time (LLOOGG), on which Sanfilippo has been working. After several months of development was Redis stable enough and it has so many features that it was put into the project LLOOGG. The consequence was that in a production environment MySQL database has been replaced by a Redis database. Other months have brought an immediate increase of popularity thanks to the community, which began to develop this database system. In 2010 arrived probably the most important event in the short history of the project: VMware hired Salvatore Sanfilippo, so he could devote to Redis project fully. Most importantly, the project is still provided under the BSD license [http://sk.wikipedia.org/wiki/BSD_licencia].

Redis is a **key-value** type database. Simplified, these databases are like one table containing two columns of text: **key** and **value**. The most important information regarding the **key**:

- Redis distinguish letter case in the name of the **key**.
- In the key name should not appear a gap.
- Common convention (of course not set by server) providing the names of the keys is as follows: **ObjectType: id: field**.

The best Redis description is that Redis is the server of sorted structural data (data structure server). Redis supports the following data structures: Strings, Sets, Sorted Sets, Lists, Hashes (Hash tables).

2. Installing Redis

Linux

Installation of Redis server (along with the command line) is for Linux very easy even for inexperienced users:

```
wget http://redis.googlecode.com/files/redis-2.4.14.tar.gz
tar xzf redis-2.4.14.tar.gz
cd redis-2.4.14
make
src/redis-server
src/redis-cli
```

Windows

There is no official support for Windows (what can be considered to be serious problem), but for testing purposes we can use solution accessible at [<https://github.com/MSOpenTech/redis>]. It is a small archive containing Redis for Windows. It is not recommended to use this solution in a production environment. We can only hope that there will be official version of Redis for Windows too in the near future, which will broaden the field of potential users.

Try.Redis-DB.com

If we want to become familiar with Redis database command line quickly, we can visit the site Redis-DB.com [<http://try.redis-db.com/>]. We use it to get know the Redis system commands quickly without installing it (there is a short interactive tutorial on the site too). We believe that every database system should have access to online tools to get acquainted with the database environment just like Redis. Pattern for Try.Redis-DB.com was Try.mongoDB.org page that offers similar functionality to the system MongoDB.

3. Server startup and configuration

Server startup

The server is started by `redis-server` command. To run the server with a predetermined configuration, it is necessary to enter `redis-server` command with path to the configuration file as a parameter. If the server was run with no specified configuration file, the server uses the default settings.

Server configuration (file `redis.conf`)

An example of configuration is shown in the file `redis.conf`. The most important configuration parameters are shown below:

- `<port_number> port` – sets the port on which Redis will work (default port is 6379).
- `loglevel <debug/verbose/notice/warning>` – sets how much information will be shown in the server console. The default is `verbose`, but it should be changed to `notice`.
- `save <seconds> <changes>` – determines how often the content of the database is recorded to the hard disk. The value `seconds` specifies the interval of writing data to the disk in seconds and value `changes` determines how many keys have to be changed to start writing data to the disk. For example, parameter `save 300 10` means write to disk every 300 seconds, if 10 keys at least was changed. If the `save` parameter is not specified in the configuration, the content of the database will not be written to the disk (but by commands `bgsave` or `save` we can force content of the database to be written to the disk).
- `dbfilename <filename>` – path to the file containing the database contents. Default setting is `Dump.rdb`.
- `appendonly <yes/no>` – default Redis setting is to write data to disk by asynchronous method (by setting of the parameter `save`). This may cause that in case of system failure last modified data would not be saved. To avoid this situation, we can set `append only` method. Setting up this method will cause that any operation will be saved into a file determined by parameter `appendfilename` (default `appendonly.aof`). In case of server startup (for example after its failure), the database state will be reconstructed by the file `appendonly.aof`. The default setting is `no` (what means that `append only` method is inactive).
- `appendfilename <filename>` – path to the file, which `append only` method uses. The default value of this parameter is `appendonly.aof`.
- `requirepass <password>` - password to the database (default setting is no password). If the Redis enables remote access and it requires a password,

we recommend to use very strong password (Redis is able to test over 150 000 passwords per second).

- `slaveof <masterip> <masterport>` – is responsible for setting replication between the servers (determines from which server the data will be downloaded). Redis provides replication by using master-slave model (generally said: changes of `master` instance are transmitted to slave instance, changes of `slave` instance are not transmitted to `master` instance).

4. Database command line

Start of command line command is caused by `redis-cli` command. If no parameters are provided to the command, the command line starts with the default settings, what mean it connects to the server with the address 127.0.0.1 (localhost) on port 6379. To connect to another database, there must be specified following parameters (complete list of parameters is obtained by typing `redis-cli-help` command):

```
-h <hostname>,          -p <port>.
```

Example: `redis-cli-h 192.168.1.3 -p 6378`.

5. Commands

The most important commands of Redis are listed below. A complete list of commands can be found at [<http://redis.io/commands>]. Redis is being developed constantly, thus the number of commands is increased in each new version. Commands in Redis are case insensitive. If we use a key name in the command, it must be the full key name (if description of the command is not different), i.e. without the use of universal symbols “*” or “?”.

Commands associated with keys

- `DEL key [key ...]` - removes `key` (`keys`). Other keys should be separated by a space. Returned value of this command is the number of deleted records. Command: `DEL *` does not removed all keys, but only the key “*” (if such exists).
- `EXISTS key` - returns value `1` if `key` exists, otherwise returns value `0`.
- `KEYS pattern` - find all the keys matching the `pattern`. It is possible to use the symbol “?” (substituting one character) or the symbol “*” (substituting more characters). Command `KEYS *` returns all keys in the database.
- `RANDOMKEY` - returns a random key from the database.
- `RENAME key newkey` - changes the key name. The `key` must exists in the database.
- `RENAMENX key newkey` - changes the name of the `key` only when the `newkey` is not in the database.

- **TYPE key** - returns the type of data stored under **key** (string, set, sorted set, hash, list). If the key does not exist in the database, returned value is **none**.

Temporary keys. There are two types of keys in Redis:

- **Ordinary keys** – they are accessible after their creation normally.
- **Temporary keys** – They are accessible only a certain number of seconds after their creation. Temporary keys may appear as very strange and impractical element, but in practice there is need to store data only for a limited period often (e.g. information about the user's session). The commands:
 - **EXPIRE key seconds** - determines how many **seconds** will be **key** accessible.
 - **PERSIST key** - removes the "temporality" of the **key**. After the command execution, the temporary key becomes an ordinary key.
 - **TTL key** - returns information how many seconds will be **key** accessible yet. For ordinary keys or keys that do not exist in the database it returns value **-1**.

Commands associated with strings

String is the simplest type of data stored in Redis.

- **SET key value** - sets the **value** of **key**
- **MSET key value [key value ...]** - sets the **value** of multiple **keys** simultaneously
- **SETNX key value** - sets **value** of the **key** only if **key** is not in the database.
- **GET key** - returns **value** of the **key**.
- **MGET key [key ...]** - returns **values** of multiple **keys** simultaneously.
- **APPEND key value** – appends the **value** to **value** of the **key**.
- **STRLEN key** - returns the length of the string **value** under the **key**.

Commands associated with sets

The elements of the set can not be repeated, sequence of elements insertion into the set is irrelevant (or not recorded) and we can only determine whether a given element belongs to the set or not. These rules correspond to set interface in Java language.

- **SADD key member** – adds element **member** to the set under the **key**.
- **SREM key member** – removes element **member** from the set under the **key**.
- **SRANDMEMBER key** – returns a random element from the set under the **key**.
- **SPOP key** – returns and removes the random element of the set under the **key** (a combination of commands **SRANDMEMBER** and **SREM**).

- SCARD **key** – returns the total number of elements in the set under the **key**.
- SISMEMBER **key member** – returns value **1** if the **member** is an element in the set under the **key**, otherwise it returns value **0**.
- SMEMBERS **key** – returns all elements in the set under the **key**.
- SUNION **key [key ...]** – returns union of sets under given **keys**.
- SUNIONSTORE **destination key [key ...]** – returns union of sets under given **keys** and outcome inserts into set under the **destination key**.
- SDIFF **key [key ...]** - returns difference of sets under given **keys**.
- SDIFFSTORE **destination key [key ...]** – returns difference of sets under given **keys** and outcome inserts into set under the **destination key**.
- SINTER **key [key ...]** – returns intersection of specified sets (only the common elements of the sets).
- SINTERSTORE **destination key [key ...]** – returns intersection of specified sets and outcome inserts into set under the **destination key**.
- SMOVE **source destination member** – move element **member** from set under the **source key** to set under the **destination key**.

Commands associated with lists

The elements in the list can be repeated and the insertion sequence of elements into the list matters. These rules correspond to the list interface in Java language. Indexes of elements in the list: 0 means the first element, 1 second element, -1 last element, -2 penultimate element, etc.

- LPUSH **key value** – inserts element **value** into the list under the **key** (element **value** is inserted before the first element of the list). If the list does not exist, it is created.
- LPUSHX **key value** - works like LPUSH, but the list must exist.
- RPUSH **key value** – inserts element **value** into the list under the **key** (element **value** is inserted after the last element of the list). If the list does not exist, it is created.
- RPUSHX **key value** - works like RPUSH, but the list must exist.
- LPOP **key** – returns and then remove the first element of the list under the **key** (the element with index 0).
- RPOP **key** – returns and then remove the last element of the list under the **key** (the element with index -1).
- LSET **key index value** – sets the **value** of the element with **index** in the list under the **key** to **value**.
- LLEN **key** – returns the count of elements in list under the **key**.
- LINDEX **key index** – returns the element located at **index** position in the list under the **key**.

- `LRANGE key start stop` – returns elements located at positions with indexes in range `start-stop`.
- `LTRIM key start stop` – leaves in the list under the key only elements with indexes in the range of `start-stop` and others are removed.

Commands associated with Hashes (hash tables)

Elements of hash tables are accessed by using the name of field, what corresponds to Map interface in Java language.

- `HSET key field value` – sets `field`'s value in hash table under the `key`.
- `HMSET key field value [field value ...]` – sets multiple values of fields in hash table under the `key`.
- `HSETNX key field value` – sets `field`'s value in hash table under the `key` only if the `field` does not exist in the hash table.
- `HGET key field` – returns value of `field` in hash table under the `key`.
- `HMGET key field [field ...]` – returns multiple values of given fields in hash table under the `key`.
- `HKEYS key` – returns all fields in hash table under the `key`.
- `HVALS key` – returns all values in hash table under the `key`.
- `HLEN key` – returns count of all fields in hash table under the `key`.
- `HDEL key field` – removes `field` from hash table under the `key`.
- `HEXISTS key field` – returns value `1` if in hash table under the `key` exists `field`, otherwise returns value `0`.

6. Jedis

Redis became very popular during a few years and therefore there raised a lot of libraries that allow to connect to Redis. Jedis [<https://github.com/xetorthio/jedis>] is a library that developers Redis recommend for use in Java language. Its use is very intuitive and it consists only in using methods that correspond to the above described commands.

In addition to `jedis-2.0.0.jar`, `commons-pool-1.6.jar` file is needed too [http://commons.apache.org/pool/download_pool.cgi].

7. Summary

Advantages

- Performance. High performance is a main advantage of Redis. It should be noted that for each command is listed its time complexity at the project documentation, for example $O(\log(N) + M)$.
- Easy configuration and efficient replication.

- Ideal for deposition of a large amount (order of millions) of information that is composed by small amounts of data (such as records of registered users' sessions).

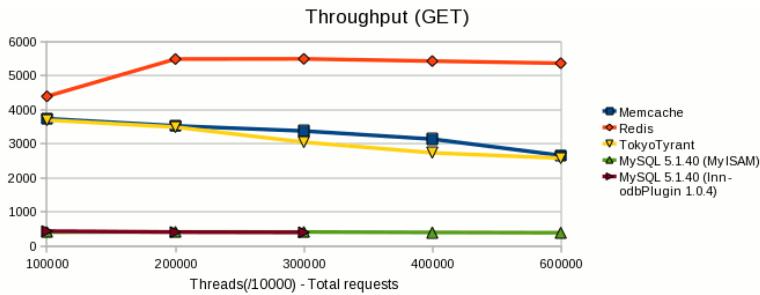


Figure 1. Performance comparison of various database systems.

Disadvantages

- A simple data model. This model requires greater amount of programmer's time and increased knowledge and experiences in implementing various elements.
- Working with data at a low level of abstraction.
- The entire database is loaded into server memory (so the server must be configured properly).

Conclusion

Redis is fancied especially for its high performance. The consequence is that this data model that can appear too easy in comparison to more complex systems. On the other hand, Redis provides a simplicity and data structures, what is very intuitive for programmers. It is difficult to assess the environment Redis from these contradictory properties. Nevertheless, we recommend becoming familiar with this environment.

Contact address

RNDr. Marian Kováč, Department of Mathematics Methods, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic,
E-mail address: marian.kovac@fri.uniza.sk

ALGORITHM FOR FINDING ALL MINIMAL CUT SETS IN RELIABILITY ANALYSIS BASED ON LOGIC DIFFERENTIAL CALCULUS

MIROSLAV KVAŠŠAY (SK)

Abstract. Reliability is one of the important characteristics of many real systems. Most of the systems consist of more than one component. Individual components have different influence on system reliability. Therefore, one of the principal step in reliability analysis is identification of those components that have the most influence on the system reliability. Some of techniques, which are used for this task, are based on minimal cut sets. Minimal cut sets have to be identified before using these techniques. In this paper, a simple algorithm for this problem is presented. The algorithm has been implemented in C++ programming language with using Qt library.

Key words and phrases. Reliability, structure function, min cut set, min cut vector.

ALGORITMUS PRE NÁJDENIE VŠETKÝCH MINIMÁLNYCH REZOV V TEÓRII SPOĽAHLIVOSTI, ZALOŽENÝ NA LOGICKOM DIFERENCIÁLNOU POČTE

Abstrakt. Jednou zo základných charakteristík mnohých systémov je spoľahlivosť. Reálne systémy sú obyčajne tvorené viacerými komponentmi, pričom obyčajne je vplyv jednotlivých komponentov na spoľahlivosť systému rôzny. Preto je jednou zo základných úloh teórie spoľahlivosti identifikácia tých komponentov, ktoré majú najväčší vplyv na funkčnosť systému. Za týmto účelom sa používa niekoľko metód, z ktorých sú niektoré založené na tzv. minimálnych rezoch. Avšak, nevyhnutným predpokladom týchto techník je znalosť jednotlivých minimálnych rezov. Predmetom tohto článku je popis jednoduchého algoritmu pre nájdenie všetkých minimálnych rezov systému. Uvedený algoritmus bol implementovaný v jazyku C++ s využitím knižnice Qt.

Kľúčové slová. Spoľahlivosť, štruktúrna funkcia, minimálny rez, minimálny rezný vektor.

Introduction

From the point of view of reliability engineering, system can be in two difference states — functional or failed. The reliability is defined as probability that the system is functional. However, most of the systems consist of more than one component. Every component can be in one of two states — operational and failed. The correlation between system state and states of its components is defined by the structure function $\phi(\mathbf{x})$:

$$\phi(\mathbf{x}) = \phi(x_1, x_2, \dots, x_n) : \{0, 1\}^n \rightarrow \{0, 1\}, \quad (1)$$

where

- n is a number of system components,
- x_i is a state of the i -th component, for $i = 1, 2, \dots, n$ (0 – the component is failed; 1 – the component is functional),
- $\mathbf{x} = (x_1, x_2, \dots, x_n)$ is a state vector.

Every component is characterized by probabilities of functional and failed state:

$$p_i = \Pr\{x_i = 1\}, \quad q_i = \Pr\{x_i = 0\}, \quad p_i + q_i = 1, \quad i = 1, 2, \dots, n. \quad (2)$$

Using (1) and (2), the system reliability and unreliability are defined as follows:

$$R = \Pr\{\phi(\mathbf{x}) = 1\}, \quad U = \Pr\{\phi(\mathbf{x}) = 0\}, \quad R + U = 1. \quad (3)$$

Equations (3) can be used to determine the overall reliability or unreliability of the system. In reliability analysis, there exist also importance measures, which are used for identification of system components that are the most important for system working. Some of them (e.g. Fussell-Vesely importance [2]) are based on minimal cut sets.

In what follows, we assume that system is coherent, which means every component is relevant and the structure function (1) is non-decreasing [2]. Also, using the convention that $\mathbf{x} > \mathbf{y}$, where \mathbf{x} and \mathbf{y} are two different states vectors, means that $x_i \geq y_i$ (for $i = 1, 2, \dots, n$) and there exists at least one i such that $x_i > y_i$.

1. Minimal Cut Set and Minimal Cut Vector

There exists more definitions of a minimal cut set (MCS) [2]. For our purpose, the best one is the following [1]:

Definition 1.1. *A cut set is a set of components of a system whose simultaneous failure leads into the failure of the system (if the system has been operational). A cut set is minimal, if no component can be removed from it without losing its status as a cut set.*

The similar term, which is very close to a MCS, is a minimal cut vector (MCV). A MCV is defined as follows [3]:

Definition 1.2. *A state vector \mathbf{x} is a cut vector if $\phi(\mathbf{x}) = 0$. A cut vector \mathbf{x} is minimal if $\phi(\mathbf{y}) = 1$ for any $\mathbf{y} > \mathbf{x}$.*

There exists one-to-one correspondence between (minimal) cut sets and (minimal) cut vectors [1]. The relation between a (minimal) cut set and the corresponding (minimal) cut vector is as follows [1]:

- if a state vector \mathbf{x} is a (minimal) cut vector, then the corresponding (minimal) cut set consists of components, which are in 0-state,
- if there is a (minimal) cut set, then the corresponding (minimal) cut vector has 0-value in the positions of all components from the (minimal) cut set and 1-value in other.

It means that if we need to find MCSs, we can compute MCVs and then, using the previous rules, transform them into the corresponding MCSs.

2. Basic Algorithm for Finding All Minimal Cut Vectors

Very simple algorithm for finding all MCVs has been presented and analyzed in the paper [1]. If we denote the set of all MCVs as MCVS and we use the following notation:

$$\mathbf{x}^{(i+)} = (x_1, x_2, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n),$$

$$\mathbf{x}++ = \text{assignment of the lexicographic successor of state vector } \mathbf{x} \text{ into } \mathbf{x},$$

then the algorithm is as follows:

Alg_1:

0. MCVS = \emptyset

1. $\mathbf{x} = (0, 0, \dots, 0)$

2. for $i = 1, 2, \dots, n$:

if $\mathbf{x}^{(i+)} \neq \mathbf{x}$ AND $\phi(\mathbf{x}^{(i+)}) = \phi(\mathbf{x})$ then
go to step 4

3. MCVS = MCVS \cup $\{\mathbf{x}\}$

4. $\mathbf{x}++$

5. if $\mathbf{x} < (1, 1, \dots, 1)$ then
go to step 2

else

stop

The previous algorithm is based on the following statement [1]:

$$\mathbf{x} \text{ is a MCV} \iff \phi(\mathbf{x}) = 0 \wedge (\forall i \in \{1, 2, \dots, n\}; \mathbf{x}^{(i+)} \neq \mathbf{x} : \phi(\mathbf{x}^{(i+)}) = 1). \quad (4)$$

In the paper [1], there has been also shown that the complexity of this algorithm is $O(nf(n)2^n)$ in the worst case, where $f(n)$ represents the average number of operations for calculation of the value of the structure function for a given state vector. This complexity is caused by three factors [1]:

1. The algorithm iterate over 2^n state vectors.
2. On the average, every state vector can be compared with $n/2$ other ones.
3. On the average, there are performed $f(n)$ operations to compute the system state for a given state vector.

3. Minimal Cut Vectors and Logic Differential Calculus

Logic differential calculus is a special tool that can be used for analysis of dynamic properties of logic functions. The part of this technique is a direct partial logic derivative (DPLD) that can be used to identify situations, in which the change of

a logic variable results into the change of the logic function. The DPLD is defined as follows [4]:

$$\frac{\partial\phi(j \rightarrow \bar{j})}{\partial x_i(s \rightarrow \bar{s})} = \begin{cases} 1, & \text{if } \phi(s_i, \mathbf{x}) = j \text{ and } \phi(\bar{s}_i, \mathbf{x}) = \bar{j}, \\ 0, & \text{else.} \end{cases} \quad (5)$$

In the paper [4], there has been shown that this technique can be also used in reliability analysis, because the formal definition of the structure function (1) is same as the logic function definition. However, in the case of a coherent system only two DPLDs can have nonzero values:

$$\frac{\partial\phi(1 \rightarrow 0)}{\partial x_i(1 \rightarrow 0)} = \frac{\partial\phi(0 \rightarrow 1)}{\partial x_i(0 \rightarrow 1)}. \quad (6)$$

The first DPLD identifies situations, in which the failure of component i results into the failure of the system, and the second one identifies those cases, when the i -th component repair leads into the system repair. It is clear that these two cases are symmetric.

If the last element of the second DPLD in (6) is nonzero, then it means that repair of the i -th component results into the system repair in the case when all other components are functional, i.e.:

$$\begin{aligned} (1_1, 1_2, \dots, 1_{i-1}, 0_i, 1_{i+1}, \dots, 1_n) &\rightarrow (1_1, 1_2, \dots, 1_{i-1}, 1_i, 1_{i+1}, \dots, 1_n) \\ &\downarrow \\ 0 &\rightarrow 1. \end{aligned}$$

This is very similar to the statement (4). Therefore, the last element of DPLD $\partial\phi(0 \rightarrow 1)/\partial x_i(0 \rightarrow 1)$ can be used for identification those MCVs, which have only one zero element, i.e. MCSs, which have only one component.

There also exist DPLDs of higher dimensions [5]:

$$\frac{\partial\phi(j \rightarrow \bar{j})}{\partial x_{i_1}(s_1 \rightarrow \bar{s}_1) \dots \partial x_{i_k}(s_k \rightarrow \bar{s}_k)} = \begin{cases} 1, & \text{if } \phi(s_1 1_{i_1}, \dots, s_k 1_{i_k}, \mathbf{x}) = j \\ & \text{and } \phi(\bar{s}_1 \bar{1}_{i_1}, \dots, \bar{s}_k \bar{1}_{i_k}, \mathbf{x}) = \bar{j}, \\ 0, & \text{else.} \end{cases} \quad (7)$$

that identifies situations, in which the simultaneous change of k variables leads into the change of the logic function. Since the DPLD (7) models influence of the simultaneous change of k variables on the logic function, then its values do not depend on the variables order, i.e. the following equations hold:

$$\begin{aligned} \frac{\partial\phi(j \rightarrow \bar{j})}{\partial x_{i_1}(s_1 \rightarrow \bar{s}_1) \dots \partial x_{i_k}(s_k \rightarrow \bar{s}_k)} &= \\ &= \frac{\partial\phi(j \rightarrow \bar{j})}{\partial x_{\pi(i_1)}(s\pi(1) \rightarrow \bar{s}\pi(1)) \dots \partial x_{\pi(i_k)}(s\pi(k) \rightarrow \bar{s}\pi(k))}, \quad (8) \end{aligned}$$

where π is any permutation of integers $1, 2, \dots, k$.

DPLDs (7) can be used in reliability analysis to identify situations, in which the simultaneous change of states of k components results into the change of the system state. There exist many DPLDs of k dimensions that can be used for analyzing of coherent systems, but we consider only the following ones:

$$\frac{\partial\phi(1 \rightarrow 0)}{\partial x_{i_1}(1 \rightarrow 0) \dots \partial x_{i_k}(1 \rightarrow 0)} = \frac{\partial\phi(0 \rightarrow 1)}{\partial x_{i_1}(0 \rightarrow 1) \dots \partial x_{i_k}(0 \rightarrow 1)}. \quad (9)$$

The first DPLD identifies those cases, in which the simultaneous failure of components i_1, i_2, \dots, i_k leads into the failure of the whole system, and the second one determines those situations, when the simultaneous repair of components i_1, i_2, \dots, i_k results into the system repair. Also, it is clear that these two cases are symmetric.

Again, if the last element of the DPLD $\partial\phi(0 \rightarrow 1)/\partial x_{i_1}(0 \rightarrow 1) \dots \partial x_{i_k}(0 \rightarrow 1)$ is nonzero, then it means that the simultaneous repair of components i_1, i_2, \dots, i_k results into the system repair in the case when all other components are functional, i.e.:

$$\begin{array}{c} (1_1, 1_2, \dots, 1_{i_1-1}, 0_{i_1}, 1_{i_1+1}, \dots, 1_{i_2-1}, 0_{i_2}, 1_{i_2+1}, \dots, 1_{i_k-1}, 0_{i_k}, 1_{i_k+1}, \dots, 1_n) \\ \downarrow \\ (1_1, 1_2, \dots, 1_{i_1-1}, 1_{i_1}, 1_{i_1+1}, \dots, 1_{i_2-1}, 1_{i_2}, 1_{i_2+1}, \dots, 1_{i_k-1}, 1_{i_k}, 1_{i_k+1}, \dots, 1_n) \\ \downarrow \\ 0 \rightarrow 1. \end{array}$$

From the cut set definition 1.1 and the relation between cut sets and cut vectors, we can see that the last element of the second DPLD in (9) can be used to identify cut vectors. From the definition of a MCS 1.1, it is clear that a cut set is minimal if no component can be removed from it. In the terms of DPLDs it means that DPLD (9) identifies a MCS if and only if its last element is nonzero and all DPLDs that have smaller dimensions than the current DPLD and that contain only those variables, which are in the current DPLD, have zero value in the last position.

4. Algorithm for Finding All Minimal Cut Vectors Based on Direct Partial Logic Derivatives

The statements presented in the previous section, leads into the following algorithm for computation of all MCVs:

1. Compute all DPLDs according to one variable.
2. Identify MCSs (MCVs) in those DPLDs, which have nonzero value in the last position.
3. Select components (variables), which DPLDs do not contain nonzero value in the last position, and compute the second order DPLDs according to all variable combinations, created from selected components.
4. Repeat steps 2 and 3 for DPLDs of higher dimensions until there does not exist any DPLD that have zero value in the last position.

Note that we do not need to compute all elements of individual DPLDs but only the last element. Therefore, the computational complexity of DPLD calculation in our case is only $O(f(n))$ (according to (7), the value of the structure function has to be computed only for two state vectors). Next, there exist n different DPLDs according to one variable, which have to be computed. Then, there have to be computed at most $\binom{n}{2}$ DPLDs according to two variables, $\binom{n}{3}$ DPLDs according to three variables, \dots , and finally, $\binom{n}{n}$ DPLD according to n variable. So, in the worst case, there have to be calculated:

$$\sum_{k=1}^n \binom{n}{k} = 2^n - 1 \approx 2^n.$$

DPLDs to find all MCVs (MCSs). It means that the worst case time complexity of the proposed algorithm is $O(f(n)2^n)$, which is better than the algorithm presented in paper [1].

4.1. Implementation of the Algorithm

DPLDs have to be computed in a special order to implement the aforementioned algorithm. Consider a system with 3 components. The DPLDs are in Figure 1.

In Figure 1, the arrows show the precedence of DPLDs (i.e. which DPLDs have to be computed if the current DPLD has zero value in the last position). From the proposed algorithm, it is clear that the tree, depicted in the Figure 1, have to be traversed in level-order. Therefore, a queue is needed to implement this algorithm. This is also the main disadvantage of the proposed algorithm, because its memory demands are not constant with compare to the first algorithm.

Using the notation:

$$\mathbf{x}^{(i-)} = (x_1, x_2, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n)$$

the algorithm for finding all MCVs, based on the DPLDs, can be implemented as follows:

Alg_2:

0. $\text{MCVS} = \emptyset$
1. $\mathbf{x} = (1, 1, \dots, 1)$, $\mathbf{found} = (n + 1, n + 1, \dots, n + 1)$, $level = 1$
2. $\text{s_queue.push}(\mathbf{x})$, $\text{s_queue.push}((2, 0, \dots, 0))$, $\text{start_queue.push}(1)$
3. while NOT $\text{s_queue.isEmpty}()$
 - $\mathbf{x} = \text{s_queue.pop}()$
 - if $\mathbf{x}[1] < 2$ then
 - $\text{start} = \text{start_queue.pop}()$
 - for $i = \text{start}, \text{start} + 1, \dots, n$:
 - if $\mathbf{x}[i] > 0$ then
 - if $\phi(\mathbf{x}^{(i-)}) < \phi(\mathbf{x})$ and $\mathbf{found}[i] \geq level$ then
 $\text{MCVS} = \text{MCVS} \cup \{\mathbf{x}\}$, $\mathbf{found}[i] = level$

```

if  $\phi(\mathbf{x}^{(i-)}) > 0$  then
    s_queue.push( $\mathbf{x}^{(i-)}$ ), start_queue.push( $i + 1$ )
else if NOT s_queue.isEmpty() then
    level = level + 1, s_queue.push( $(2, 0, \dots, 0)$ )

```

The final version of the proposed algorithm uses some special variables. Firstly, it uses variable *level* that identifies the current level in the DPLDs tree (Figure 1). The end of every level is defined by inserting of dummy state vector $(2, 0, \dots, 0)$ into the queue of state vectors (or DPLDs). Secondly, array **found** is used to store information about computed DPLDs to avoid to calculation of needless DPLDs.

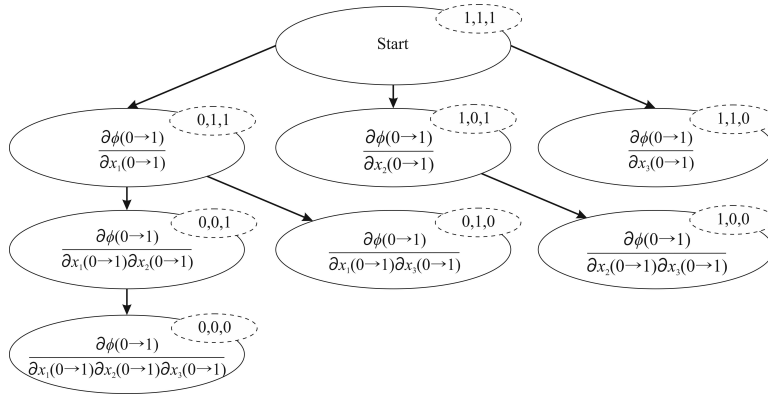


Figure 1. The tree of DPLDs and corresponding state vectors, that can be identify as MCVs for system with 3 components.

5. Experiments

Algorithms Alg_1 and Alg_2 have been implemented in C++ language using stl and Qt libraries. The real time complexity of the algorithms have been compared on series (system is functional if all system components are functional), parallel (system is operational if at least one system component is functional) and k out of n (system is operational if at least k system components are in operation) systems. The results of the experiments, realized on a computer with CPU Intel Core i5-3210M (2.50 GHz) and 4 GB of RAM, are in Table 1.

From Table 1, we can see that generally, the proposed algorithm is faster than the algorithm from paper [1], but it is slightly slower for parallel systems, in which the whole tree of state vectors (Figure 1) has to be traversed.

6. Conclusion

MCSs (or MCVs) are often used in reliability analysis. The principal problem is to identify them, when only the structure function is known. Therefore, the

System	Components counts	Algorithms [sec]	
		Alg_1	Alg_2
Series	20	0.28	0.00
	22	1.20	0.00
	25	11.02	0.00
Parallel	20	0.27	0.30
	22	1.18	1.25
	25	10.38	10.44
11 out of 10	20	0.34	0.16
12 out of 10	22	1.47	0.65
13 out of 10	25	13.42	6.00

Table 1. Results of the experiments.

simple algorithm for calculation of all MCVs of a coherent system is presented in this paper. The time complexity of the presented algorithm is better than the basic algorithm, which has been presented in [1]. However this algorithm has large memory consumption. Because of that, another research is needed.

References

- [1] KVAŠŠAY, M.: *An Algorithm for finding all minimal cut vectors in reliability analysis*, TRANSCOM 2013, Žilina, June 24–26, 2013 (accepted).
- [2] RAUSAND, M. – HØYLAND, A.: *System Reliability Theory: Models, Statistical Methods, and Applications*, Hoboken, John Wiley & Sons, Inc., 2004, ISBN 0-471-47133-X.
- [3] LISNIANSKY, A. – LEVITIN, G.: *Multi-State System Reliability: Assessment, Optimization and Applications*, Singapore, World-Scientific 2003, ISBN 981-238-306-9.
- [4] ZAITSEVA, E. N. – LEVASHENKO, V. G.: *Importance analysis by logical differential calculus*, Automation and Remote Control, February 2013, vol. 74, no. 2, pp. 171–182.
- [5] ZAITSEVA, E. – PUURONEN, S.: *Estimation of multi-state system reliability depending on changes of some system component efficiencies*, Proc. of European Safety and Reliability Conference (ESREL 2007), Stavanger, Norway, June 25–27, 2007, pp. 253–261. ISBN 0-415-44785-2.

Contact address

Ing. Miroslav Kvaššay, Department of Informatics, Faculty of Management Science and Informatics, University of Žilina, Univerzitná 8215/1, 010 26 Žilina, Slovak Republic,
E-mail address: Miroslav.Kvassay@fri.uniza.sk

STAHOVÁNÍ PŘES PYTHON-SELENIUM

PAVEL STRÍŽ (CZ)

Abstrakt. Autor článku představuje ne tak známý problém stahování webových stránek se zařazenými výpočty přes JavaScript. V tomto článku to bude konkrétně problém se zařazením výpočtu časové známky. S touto situací se setkáme například při automatizovaném stahování rozdání na serveru BBO. Rozdání jsou navíc rozdělena do tří kategorií: hlavní bridžový klub, turnaje a týmové zápasy a webová stránka se musí vhodně napařovat a zpracovat. Další problém je, že se musí vyplnit elektronický formulář než se k výpisu rozdání vůbec dostaneme. Co nejsnadnější přístup k rozdáním je důležitý pro bridžové hráče, jejich trenéry, rozhodčí na turnajích a dokonce i pro diváky, aby si mohli rozdání znovu přehrát a připravit si analýzy.

Klíčová slova. Python, Selenium, JavaScript, stahování, karetní hra bridž.

DOWNLOADING VIA PYTHON-SELENIUM

Abstract. The author introduces not so common problem of downloading web pages with JavaScript computations included. In this article, it is specifically a computation of timestamp. For instance, this problem occurs when we try to download bridge hands and travellers at BBO automatically. Moreover, the hands are divided into main bridge club, tourney and team match hands categories and the web page must be parsed appropriately. Another problem is that an electronic form must be filled in advance before we can get a list of hands. Easy access to hands is important for bridge players, their coaches, tournament referees, and even spectators for later hand preview and post game analyses.

Keywords. Python, Selenium, JavaScript, Downloading, Bridge.

Úvod

Na následujícím stránkách se pokusím čtenáři nastínit krkolomnou cestu stahování odehraných bridžových rozdání na BBO. To jsou klíčové informace pro hráče k analýzám po sehrávce a také pro redaktory, když píšou o turnaji nějaký bridžový článek, tak aby si nemuseli karty překlíkávat. Po stažení tzv. LIN souboru si mohou projít licitaci, sehrávku a v mnoha případech i psané komentáře komentátorů.

Pokusím se bridžovou tematiku omezit na minimum a zaměřit se na jádro problému z programátorského hlediska, ovšem, k tomu jádru vedlo mnoho nezdařených pokusů. Bylo jich tolik, že jsem si už myslel, že jsem krátkozraký a zřejmě řešení pod nosem nevidím. Držel jsem se však zásady, že ve virtuálním světě je vše možné a že nějaké řešení musí existovat.

1. Naznačení problému

Pokud známe odkaz na soubor na internetu, téměř vždy vystačíme s programem `wget`. Například konkrétní známé rozdání na BBO lze stáhnout takto:

```
$ wget "http://www.bridgebase.com/myhands/fetchlin.php
?id=2236880&when_played=1365113944"
```

Stažený soubor si vhodně přejmenujeme a může se karetně bádát. K tomuto odkazu se dá bohužel dostat jen přes formulář na BBO. Uvedu funkční ukázkou přes jiný bridžový server, kde je elektronický formulář a s `wget` si už nevystačíme:

```
$ curl --data "s=AKxx&h=xxxxx&d=AK&c=xx"
http://www.rpbridge.net/cgi-bin/xhe1.pl
```

Na řadu nastoupil `curl`. Jenže když tohle zkusíme na My Hands (Má rozdání) na BBO (následující ukázkou můžeme považovat za testovací vzorek):

```
$ curl --data "username=bobhamman&start_time[Y]=2013&start_time[M]=5
&start_time[d]=12&time_interval[0]=0&time_interval[1]=2"
http://www.bridgebase.com/myhands/index.php >test.htm
```

dostaneme v souboru `test.htm` hlášku: „*Javascript support is needed for this page (to get your local timezone). Your browser either has no Javascript support, or has such support disabled.*“ Jinými slovy na výpočet časové známky se používá Javascript a musíme k tomu mít podporu na straně klienta. Svého času jsem napsal i vedení firmy BBO, jestli by tento triviální výpočet nemohli přeprogramovat v PHP a bylo by vše stáhnutelné. Bez reakce, asi to není zatím možné.

2. Neúspěšné experimenty

Abychom si tedy automatem stáhli webovou stránku se seznamem odehraných rozdání, a ty si stáhli, tak to nešlo. Mohli jsme samozřejmě ručně dát v prohlížeči Soubor – Uložit jako... a HTM výstup parsovat a následně na LIN soubory užít `wget`. To bylo mé řešení na dlouhou dobu. Pokud si však chceme stahovat rozdání jiných hráčů, musí se jít do hloubky a u každého rozdání následuje další webová stránka. Při čtyřiceti rozdáních jste už vyčerpaní.

V této chvíli jsem začínal tušit, že problém, který jsem doufal vyřeším za pět minut, se mi protáhl na několik dosti zoufalých dní. Jde o to, že `wget` i `curl` nejsou na JavaScript podporu stavěny. Podtvdila mi to i internetová fóra.

Mám-li být ke čtenáři upřímný, docela mě to vytočilo. Měl jsem docela dost práce a tohle jsem chtěl mít vyřešené právě do těch pěti minut. Smůla!

V té chvíli odpadly nástroje jako RCurl při projektu R, cURL při PHP, ale též všechny stahovače jako Puf, Snarf, Pavuk, Htget, Lftp, Crowbar či HTTracker.

Jistou naději dávaly nástroje na snímání obrazovky, ale dostatečně jsem je nepřemluvil ke spolupráci. To byly Wkhtmltopdf či WebKit. Nepřemluvil jsem ani nástroj Flexget. Jistou nadějí dávaly jQuery (`.load` či `.get`), DoJo či JSdom, ty

jsou založené na JavaScriptu, ale problém zůstal nedotčen. Na fórech to teoreticky považovali za možné, ale prakticky jsem se nepohnul.

Chvilí jsem to ještě zkoušel přes PHP, jakožto na běžícím serveru s JavaScript podporou, ale příkaz `fsockopen` nezabral. Nezabrala mi ani PHP třída `Snoopy`. Možná mám obě ruce levé. Problém mě začal vyčerpávat, ale osvětlil mě další nápad, zkusit to přes nahozený server či prohlížeč přímo.

Vyzkoušel jsem server PhantomJS a přešel na známé prohlížeče Chromium, Firefox, Netrik, W3M, ELinks a postupně jsem začínal odpadat, když jsem zkoušel Lynx a nadějný prohlížeč LuaKit. Nic! Po pár dnech jsem se nepohl o píď.

Na jedné webové stránce jsem dohledal zajímavý tip na doplněk MozRepl při Firefoxu, zkoušel jsem se připojit přímo i nepřímo přes R, šel uvážit Perl, Java, PHP a další, ale nedostal jsem se ani ke zdokumentovanému rozhraní. Ani pod Windows ani pod Linuxem. Asi jsem používal příliš novou či příliš starou verzi.

Je možné, že jsem některé nástroje vynechal, bylo jich mnoho. Ty klíčové experimenty jsem však zmínil. Zde mé snahy skončily, vyčerpán z pokusů a vyřízený z různých nastavování jsem se věnoval jiným inženýrským problémům. V té chvíli jsem si myslel, že mi tento problém zlomil virtuální vaz.

3. Náznak řešení

Mé každodenní útočiště je typografie. Před nějakým časem jsem řešil zajímavý typografický problém. Naznačím jej, neb to nás posune kupředu. Pokud převádíte dokument z Microsoft Wordu přes LibreOffice.org a `Writer2LATEX`, tak se vám stane to, že vzorce (Microsoft Equation) zůstanou v komerčním formátu WMF. Ani konvertory typu `libwmf-bin` či `python-uniconvertor` vám nepomohou, neb u SVG vám vypadnou UTF-8 sekvence a řešit to přes formáty programů `xFig` či `IPE` je nestabilní půda, o které hovořil matematik Karel Horák na `TEXperience 2012`. Jakýkoliv zásah přes bitmapový formát (PNG) ztrácí informace o vektorových křivkách a už byste se mohli spolehnout jen na OCR a tento druh nástrojů.

Svého času jsem si Microsoft Equation rozšířil o MathType, to je původní program, který Microsoft ořezal a užívá. MathType umí WMF nejen zobrazit, ale i převést do `TEX`ového formátu. Nevýhodou je, že je to zkušební verze na 30 dní a překročení tohoto intervalu znamená např. přeinstalovat virtualizovaný stroj.

Asi si řeknete, že kvůli pár rovnicím si to člověk překliká v MathType sám či přepíše rovnou v `TEX`u z papíru či z PDF, jenže jsem měl na stole článek, kde bylo přes 400 vstupů do matematického režimu a to už jsem odmítl ručně řešit.

Použil jsem pod Windows zajímavou fintu. Souborový formát vbs je spustitelný Visual Basic Script (WSH) a pomocí Sendkeys, <http://ss64.com/vb/sendkeys.html>, lze spouštět a uzavírat programy a navíc řídit jejich chod. Zde je jistý základ, založíme si nový objekt, do kterého posíláme signály, v této ukázce `Ctrl+V` pro vložení obsahu schránky, `Ctrl+S` pro uložení, `Alt+F4` pro zavření okna a `A` pro Ano jako potvrzení před uzavřením okna (`ukazka.vbs`):

```
Set PajaShell = WScript.CreateObject("WScript.Shell")
PajaShell.Run "notepad temp.tex", 9
PajaShell.SendKeys "~V"
PajaShell.SendKeys "~S"
PajaShell.SendKeys "%{F4}"
PajaShell.SendKeys "A"
```

Soubor s příponou VBS se pod Windows spouští jako běžný dávkový soubor (BAT) nebo spustitelný soubor (EXE, COM).

Pak se můj typografický problém automatizace konverze rozpadl na:

- první pokus o převod do T_EXu s externími WMF soubory,
- Bash skript s cyklem přes všechny WMF soubory, který generuje VBS skript, který si spouští MathType a do textového editoru nasype T_EXový kód,
- jednorázovou kontrolu v PDF všech dílčích T_EX souborů, a,
- a zařazení malých T_EXových kódů do řídicího T_EXového souboru. To bylo možné, protože po konverzi ve Writer₂L^AT_EXu zůstává v T_EXovém souboru informace, přesněji T_EXová poznámka, o neúspěšné konverzi WMF.

Se vším jsem byl spokojený, jen se při vlastní konverzi nesmělo dotýkat klávesnice, běželo mi to na hlavním stroji, nikoliv na stroji virtuálním. Ale to byl zanedbatelný problém v daném momentě.

Během řešení tohoto typografického oříšku jsem narazil na alternativní nástroje, Swapy, <http://code.google.com/p/swapy/> a pyWinAuto, <http://pypi.python.org/pypi/pywinauto/>, které jsem sice nevyužil, ale dávaly prostor současný problém řešit pod Microsoft Windows podobně. Tedy ovládat program, přesněji webový prohlížeč, se specifickými zadanými instrukcemi, tedy:

- zadat hráče v bloku My Hands přes formulář na BBO,
- stáhnout si zobrazenou webovou stránku,
- tu si naporovat a dle potřeby zobrazit dílčí webové stránky,
- ty si stáhnout, naporovat a stáhnout příslušné LIN soubory (to lze přes `wget`, bylo ukázáno na druhé straně tohoto článku), kde jsou obsažena rozdání, která nás zajímají.

Mám-li se dobrovolně přiznat, vědecká, chceme-li odborná, krátkozrakost v té době mi neumožnila se k těmto nástrojům vrátit. Měl jsem řešení pod nose, jen si na to vzpomenout. Dokonce jsem nemusel ani zkoumat ony programy založené na Pythonu, mohl jsem jít přímo pod Windows znovu do VBS, ten už jsem uměl k práci přemluvit, jak bylo v drobné ukázce na této straně naznačeno.

4. Řešení

Nebudu drahou čtenářku či drahého čtenáře již déle napínat a poodhalím současně řešení. V té době jsem řešil nějaké parsování přes Python a hledal jsem

knihovny pro XML, ale především pro GnuPlot. Zkoušel jsem mj. na Googlu spojení termínů „Python Bindings“ a mezi prvními odkazy bylo Selenium. Nic mi ten název neříkal. Neznalost Zákona a knihoven Pythonu neomlouvá! Nahlédl jsem na Pypi i na Google Code. Při zběžném procházení dokumentace jsem si říkal, že to je přesně to, co potřebuji. Nahodil jsem si:

```
$ sudo apt-get install firefox python-selenium
```

A ukázkový kód z Pypi mi zas neběžel. To mě dožralo! Otevřelo se mi okno ve Firefoxu, ale pak to spadlo. Říkal jsem si, že to není možné, že to musí nějak jít. Vzpomněl jsem si na letitý problém u T_EX Live, kdy byla v repozitáři vždy verze o rok dva zpožděná. To už je vyřešený problém, ale stál T_EXisty mnoho nervů.

Nahlédl jsem na verzi a v repozitáři (červen 2013) je verze 2.25.0, na Pypi už v nadpisu blikala verze 2.33.0. „Heuréka!“ Říkal jsem si, že to musí být ono, že to je ono! Už to šlo fofrem, jako jízda z kopce:

```
$ sudo apt-get remove python-selenium
```

Ačkoliv jsem z Pythonu pod Windows občas užíval `easy_install`, na Linuxu jsem balíčky drtil přes `pip`, hlavně kvůli snadnému odstranění. Tedy:

```
$ sudo pip install selenium
```

Vzorová ukázka se pohnula. Spustila se mi pod rukama Mozilla Firefox, naklikalo se `yahoo.com` a vyskočilo SeleniumHQ a prohlížeč se zdvořile odporoučel. Jediný problém byl, že když jsem systém zatížil, tak časově zpoždění 0,2 sekundy nestačilo k naběhnutí stránky a vyhledání se nezdařilo. To byl operativní problém změny jednoho čísla na řádku s `time.sleep(0.2)`. Existuje elegantnější řešení?

Při procházení dokumentace již podrobněji jsem dohledal i přímé načtení zdrojového kódu webové stránky, takto vypadalo mé minimalizované řešení.

```
#!/usr/bin/env python
#encoding=utf-8
# Jádro programu...
try:
    # Načtení nezbytných knihoven...
    from selenium import webdriver
    from selenium.common.exceptions import NoSuchElementException
    from selenium.webdriver.common.keys import Keys
    import time
    # Začni pracovat na rozhraní...
    f=open("rozdani-vystup.htm","w") # Ulož si stránku na další parsování.
    browser = webdriver.Firefox() # Otevři si webový prohlížeč.
    browser.get("http://bridgebase.com/myhands/index.php")
    time.sleep(2) # Otevři si formulář a vyčkej chvíli na načtení.
    # Vyplňme formuláře a začněme se stahováním...
    elem = browser.find_element_by_name("time_interval[0]")
    elem.send_keys("m" + Keys.RETURN) # Projdi formulář a zapiš "m"ěsíc.
    elem = browser.find_element_by_name("username")
```

```

elem.send_keys("bobhamman" + Keys.RETURN)
time.sleep(2) # Pokračuj v hledání ve formuláři, zapiš hráče a čekej.
muj=browser.page_source # Do proměnné si uložíme zdrojový kód stránky.
f.write(muj.encode('ascii', 'ignore')) # Trvale uloží zdrojový kód.
time.sleep(3) # Programe, vyčkej chvíli před zavřením okna!
browser.close() # Zavři webový prohlížeč.
# Pokud jádro programu neuspěje...
except:
    print "Stažení se nepodařilo!"

```

Následovalo už jen spuštění (alternativa je `python rozdani.py`):

```

$ chmod +x rozdani.py
$ ./rozdani.py

```

V souboru `rozdani-vystup.htm` jsem měl přesně to, co jsem potřeboval. Zdrojové kódy webové stránky! S tím se dá dále pracovat, parsovat a prohlížet si. Pro účely tohoto nebridžového článku není potřeba dále rozvádět.

Kladl jsem si otázku, jestli půjde ukládat i stránky kódované přes UTF-8, ale to už je problém na úrovni Pythonu, nikoliv již webového prohlížeče.

Závěr

S v článku nabídnutým řešením jsem byl spokojený. Běží mi to na Linuxu na virtualizovaném stroji, takže mohu i časově náročný skript spustit a okno si minimalizovat, aby to nerušilo. Nebo to jen tak pro potěšení sledovat, jak za mě skript dělá klikací práci.

Nevýhodou zůstává, že tento postup nebude možné použít na masivní stahování, kdyby bylo potřeba, neb časové prodlevy je dobré vhodně naddimenzovat. Je možné, že se tento krok bude dát nějak obejít či optimalizovat programátorsky.

Pokud někdo tento problém stahování webových stránek s JavaScriptem vyřešil jinak, rád si řešení vyslechnu či přečtu. Problematické stránky (My Hands na BBO) byly v článku zmíněny, pro případného testovatele. Ostatně stáhnout si i vlastní formulář, <http://www.bridgebase.com/myhands/index.php>, je problém, neb i tam figuruje JavaScript a výpočet časové známky.

Nepovažuji se za hackera, ale při stahování obrázkových galerií nebyla občas jiná možnost, než klikat jak o život. Tento automatizační nástroj (dokumentace) nám život zase o něco usnadnil, jen místo prostých příkazů definujeme kroky psané a potvrzované na klávesnici. Vývoj je dramatický, Selenium IDE, též jako nezávislý Java server s možností rozšíření přes WebDriverJs, řada pluginů ap.

Kontaktní adresa

Ing. Pavel Stríž, Ph.D., Nakl. Martin Stríž, U Škol 940, 685 01, Bučovice, Česká republika,
E-mailová adresa: pavel@striz.cz

OPERAČNÝ SYSTÉM GNU/LINUX VO VYUČOVANÍ NA STREDNEJ ŠKOLE

MARTIN ŠECHNÝ (SK)

Abstrakt. Absolvent strednej školy by mal poznať základné princípy fungovania výpočtovej techniky a vedieť ju používať efektívne. Stredoškolská informatika má obsahovať hardvér, operačné systémy, aplikačný softvér, počítačové siete, programovanie. Škola by mala uprednostňovať využívanie otvoreného a slobodného softvéru. Operačný systém GNU/Linux je preto dobrou voľbou. Prezentujem skúsenosti z troch rokov experimentálneho overovania predmetu Operačné systémy v študijnom odbore Elektrotechnika – Informačné a sieťové technológie, pre ktorý som napísal učebnicu a súvisiace dokumenty. Teória je prakticky aplikovaná v inštalácii a konfigurácii operačného systému GNU/Linux. Znalosti a zručnosti sú cieleňé na zvládnutie najľahšej úrovne certifikačnej skúšky LPI.

Kľúčové slová. Operačný systém, GNU/Linux, škola, informatika, softvér, študijný odbor, učebnica, inštalácia, konfigurácia, certifikačná skúška LPI.

GNU/LINUX OPERATING SYSTEM IN EDUCATION IN SECONDARY SCHOOL

Abstract. A secondary/high school graduate should know the basic principles of operation of computer equipment and know how to use it effectively. Secondary-school computer science should include hardware, operating systems, application software, computer networks, programming. The school should promote the use of a free and open source software. GNU/Linux operating system is therefore a good choice. I present the experience of three years of experimental testing the Operating systems subject in the field of study Electrical Engineering – Information and network technologies, for which I wrote a textbook and related documents. The theory is practically applied in the installation and configuration of the GNU/Linux operating system. Knowledge and skills are focused on mastering the lightest level of the LPI certification exam.

Keywords. Operating system, GNU/Linux, school, informatics/computers, software, field of study, textbook, installation, configuration, LPI certification exam.

Úvod

V rámci učiteľskej práce na strednej škole som sa zapojil do projektu *Tvorba a implementácia inovatívneho programu Informačné a sieťové technológie*¹ z roku

¹Informačné a sieťové technológie, projekt združenia IT škola, <http://www.itsko1a.eu/news/1/18/Zavedenienovoho-studijného-odboru-Info-macne-a-sietove-technologie/d.projektydetail.htm>

2009, spolufinancovaným zo zdrojov EÚ, ktorým sa začal proces tvorby a experimentálneho overovania nového školského vzdelávacieho programu *Informačné a sieťové technológie* v študijnom odbore 26 *Elektrotechnika*. Projekt bežal/beží na niekoľkých školách. Prezентujem skúsenosti z troch rokov experimentálneho overovania v predmete *Operačné systémy* pre 2. ročník. Pre tento predmet som napísal príslušnú časť školského vzdelávacieho programu, tematický výchovno-vzdelávací plán, učebný text a metodiku. Svoj návrh som overil najprv na záujmovom krúžku a potom na predmete počas 3 rokov v 1 až 2 triedach, pričom som ho po každom roku mierne upravoval.

Predmet *Operačné systémy* pre 2. ročník obsahovo pokrýva teóriu o operačných systémoch a praktickú prácu s operačnými systémami MS Windows a GNU/Linux. Vo vytvorenom učebnom texte som sa zameral na teóriu operačných systémov a na konkrétny operačný systém GNU/Linux, s použitím porovnávania k operačnému systému MS Windows. Domnievam sa, že operačné systémy by sa mohli úspešne a lepšie učiť práve takto.

Absolvent strednej školy by mal poznať základné princípy fungovania výpočtovej techniky a vedieť ju používať efektívne. Stredoškolská informatika má obsiahnuť hardvér, operačné systémy, aplikačný softvér, počítačové siete, programovanie. Škola by mala uprednostňovať využívanie otvoreného a slobodného softvéru. Operačný systém GNU/Linux je preto dobrou voľbou. Vysvetľovaná teória je prakticky aplikovaná v inštalácii a konfigurácii operačného systému GNU/Linux počas vyučovania. Znalosti a zručnosti sú cieleňé na zvládnutie najľahšej úrovne certifikačnej skúšky LPI² *Linux Essentials*.³ Dosiahnuté výsledky žiakov môžu byť potvrdené medzinárodne uznávaným certifikátom podobne ako ECDL alebo CCNA.

Slobodný a otvorený softvér by mal dominovať vo všetkých predmetoch využívajúcich výpočtovú techniku, ako aj v prevádzke školy. Dôvodov je niekoľko – úšetrnenie peňazí, výchova k dodržiavaniu predpisov, podpora slobodného výberu a slobodného názoru, univerzálnosť a meniteľnosť softvérového vybavenia.

1. Školský vzdelávací program

Súčasťou projektu boli viaceré činnosti: absolvovanie odborných školení k predmetom, tvorba alebo úprava školského vzdelávacieho programu, tvorba učebnice, overovanie návrhu odučením záujmového krúžku, overovanie návrhu odučením predmetu.

Pri tvorbe nového školského vzdelávacieho programu (ŠkVP) sa musí vychádzať z platného štátneho vzdelávacieho programu v príslušnej skupine odborov,

²LPI (Linux Professional Institute), <http://www.lpi.org/>

³LPI Linux Essentials, <http://www.lpi.org/linux-certifications/introductory-programs/linux-essentials>

Úvod do operačných systémov – teória [10 hodín]
<ul style="list-style-type: none"> • poučenie BOZP • abstraktný hierarchický model počítača, architektúra počítača • pamäte – funkcie, prístup, technológia • operačný systém – definícia, bloková štruktúra • plánovanie procesov a vlákien • virtuálna pamäť • vlastnosti OS, história OS, porovnanie vlastností OS • inšpiratívny film (Pirates of Silicon Valley), pochopenie historického vývoja operačných systémov, porovnávanie, kritické myslenie • licencie, morálny postoj k autorským právam, opakovanie • disk, súborový systém • delenie a formátovanie disku, diskové pole
Virtuálny stroj [6 hodín]
<ul style="list-style-type: none"> • definícia, použitie, princíp virtualizácie, virtuálny stroj • Oracle VirtualBox – úvod, nastavenie, vytvorenie virt. stroja • MS Windows – inštalovanie, konfigurácia, používanie • opakovanie, úlohy
UML modelovanie operačného systému [4 hodiny]
<ul style="list-style-type: none"> • príklad na filme Matrix, pochopenie izolovania funkcií operačného systému, porovnanie s dôverou v morálne hodnoty • modelovanie algoritmov
GNU/Linux úvod [6 hodín]
<ul style="list-style-type: none"> • GNU/Linux – história, distribúcie, licencia, linuxová komunita • zdroje na internete, príprava inštalačného média • používanie live CD (live ISO) Slax, Ubuntu a iné distribúcie • inštalácia
GNU/Linux štruktúra systému [24 hodín]
<ul style="list-style-type: none"> • delenie a formátovanie disku • štruktúra súborového systému • MBR/GPT, bootlo- ader, boot, kernel, init, init skript • etc, dev, media, mnt, proc, root, sys, tmp, usr, var • používateľ, práva k súborom, bezpečnosť • príkazový riadok (bash), príkazy pre prácu so súbormi • programovanie v systéme • správa systému, reštartovanie • balíčkovacie sys- témy, aktualizácia systému, bezpečnosť • konfigurácia siete, firewall, route • konfigurácia serverových služieb (ssh, dns, dhcp, mail, www, proxy) • samba klient a server v sieti Win- dows • oblak (cloud) • konfigurácia tlačiarne a externých zariadení • X11/xorg grafické prostredie • grafické aplikácie • opakovanie
GNU/Linux aplikácie [8 hodín]
<ul style="list-style-type: none"> • webový prehliadač a poštový klient • kancelárske aplikácie (office) • prehrávanie a spracovanie multimédií, hry • programátorské nástroje, edukačný a technický softvér
Projekt [6 hodín]
<ul style="list-style-type: none"> • vytvorenie vlastnej distribúcie GNU/Linux – úloha, analýza, špecifikácia, získavanie in- formácií, riešenie, inštalačné médium, dokumentácia, prezentácia, zverejnenie
Certifikácia [2 hodiny]
<ul style="list-style-type: none"> • certifikácia (LPI, RHCE)

Tabuľka 1. Navrhovaná učebná osnova predmetu *Operačné systémy* [téma – obsah]

konkrétne 26 *Elektrotechnika*, ktorý je v pôsobnosti Štátneho inštitútu odborného vzdelávania⁴ (ŠIOV). Žiaľ, konštatujem, že ŠIOV nepripravil štátny vzdelávací program správne, aj verzia pre školský rok 2013-2014 obsahovo stále nesie zastaralú koncepciu. Dokument je dlhý, neprehľadný, obsahuje veľa balastu. Pri tvorbe ŠkVP som si pozrel aj štátny vzdelávací program pre gymnáziá, ktorý je spracovaný Štátnym pedagogickým ústavom.⁵

Chcel by som rozšíriť cieľovú skupinu učiteľov, ktorí môžu použiť novú učebnicu vo svojich školách a cieľovú skupinu žiakov stredných škôl, ktorým je učebnica určená. Operačný systém MS Windows je popísaný vo veľkom množstve kníh, učebníc, pracovných zošitov.

Vybral som si operačný systém GNU/Linux⁶ preto, že nie je doteraz takto odborne a metodicky spracovaný pre slovenské stredné školy. Škola by mala prioritne používať slobodný softvér (FOSS, free & open source software)⁷ vo vyučovaní, aj pre svoju prevádzku. Slobodný softvér dáva používateľovi slobodu ľubovoľného použitia, slobodu voľného kopírovania, slobodu voľného študovania, slobodu voľnej úpravy, vylepšenia a šírenia. Môže byť bezplatný, prípadne lacný. Autorské práva sú zachované. Práva autora a používateľa sú definované licenciou, najčastejšie štandardizovanou formou slobodnej licencie pre softvér. Prečo slobodný softvér? Táto voľba má niekoľko silných východ – lacné riešenie, legálne riešenie, neobmedzujúce riešenie, tvorivé podnety, rozvoj slobodnej a otvorenej spoločnosti. Škola by mala ponúkať vyvážený mix nástrojov – sloboda, prehľad, možnosť.

Predmet *Operačné systémy* obsahovo súvisí s ďalšími predmetmi, najmä *Softvérové aplikácie*, *Sieťové technológie*, *Programovanie*, *PC architektúra*, *Serverové technológie*.

Učebnicu k predmetu *Operačné systémy* som prepracoval do 2. verzie, v ktorej som pozmenil pôvodný ŠkVP a vyrobil som tak špecializovaný návrh odborného učiva, ktoré by sa mohlo zakomponovať do širšieho spektra študijných odborov. Vzdelávací obsah je zameraný na všeobecnú teóriu o operačných systémoch a spoznávanie konkrétneho operačného systému GNU/Linux. Navrhovaná učebná osnova predmetu (spolu 66 vyučovacích hodín – cvičenia) je v tabuľke 1.

Konkrétne použitie navrhnutého obsahu a počtu hodín závisí od voľby učiteľa vo svojom predmete v závislosti od druhu školy a druhu študijného odboru.

⁴Štátny vzdelávací program pre študijný odbor 26 Elektrotechnika, <http://www.siov.sk/svp-pre-ovp/21658s>, http://www.siov.sk/ext_dok-svp-26/24306c

⁵Štátny vzdelávací program pre Gymnáziá, <http://www.statpedu.sk/sk/Statny-vzdelavaci-program/Statnyvzdelavaci-program-pre-gymnaziaISCED-3a.alej>, <http://www.statpedu.sk/sk/Statny-vzdelavaciprogram/Statny-vzdelavaci-program-pre-gymnaziaISCED-3a/Matematika-a-praca-s-informaciami.alej>, http://www.statpedu.sk/files/documents/svp/gymnazia/vzdelavacie_oblasti/informatika_isced3a.pdf

⁶GNU/Linux, [http://sk.wikipedia.org/wiki/Linux_\(operačný_systém\)](http://sk.wikipedia.org/wiki/Linux_(operačný_systém))

⁷Slobodný softvér, http://sk.wikipedia.org/wiki/Slobodný_softvér

Hlavné ciele predmetu

Získať súbor znalostí o stavbe a fungovaní operačných systémov, získať súbor zručností pri inštalácii a konfigurácii operačných systémov, formovať logické myslenie, rozvíjať tvorivosť, schopnosť riešiť typické úlohy, aj analyzovať komplexné problémy a rozložiť ich na čiastkové úlohy, spolupracovať v tíme, vyhľadávať informácie na internete a v literatúre, dodržiavať bezpečnosť v IT, rozvíjať svoje morálne postoje, kriticky zhodnotiť situácie.

Špecifické výchovno-vzdelávacie ciele

Spôsobilosti využívať informačné technológie: vyhľadávať informácie, analyzovať, používať ich. Schopnosti riešiť problémy: rozpoznať problém, formulovať, analyzovať, hľadať informácie, postupy, metódy riešenia problému, rozdeliť problém na čiastkové úlohy, vyriešiť úlohy, zhodnotiť riešenie, porovnávať, aplikovať postupy v iných oblastiach. Schopnosti tvorivo riešiť úlohy, schopnosti spolupracovať v tíme.

2. Učebnica

Dobrá učebnica má plniť viacero funkcií: motivačnú, komunikačnú, regulačnú (udržiava logickú návaznosť), aplikačnú (poznatky používa v príkladoch a úlohách), integračnú (zahŕňa medzipredmetové vzťahy), inovačnú, kontrolnú, usmerňujúcu. Aby učebnica naplnila požadované funkcie, jej štruktúra je zložená z dôležitých prvkov, ktoré musia byť vhodne zvolené a vzájomne vyvážené [1]:

- text (základný, doplňujúci, vysvetľujúci, poznámky pod čiarou),
- príklady, úlohy, otázky, tabuľky, obrázky, schémy, pojmové mapy,
- orientačné prvky (číslovanie strán a kapitol, obsah, odrážky, register),
- štýl (strana, pozadie, zarovnanie, okraje, písmo, odsek, farby, hlavička, päta),
- informácia (názov, autor, licencia, úvod, záver, literatúra, zdroje, poznámky).

Vybral som si štýl formálnejší, pretože učebnica je určená hlavne strednej odbornej škole v technickom odbore, kde sa od žiakov očakáva technický štýl písania poznámok, dokumentácie, referátov. Spracovanie učebného textu som robil pomocou slobodného softvéru, ktorý je aj opisovaný v texte (GNU/Linux, LibreOffice, Dia, Gimp, Mozilla Firefox). Učebnica je vo formáte PDF, má 99 strán A4. Predstavím ju v prezentácii, lebo text tohto príspevku má málo miesta na tento účel. Učebnica je zverejnená na mojej webovej stránke.⁸ Prípadným záujemcom učiteľom rád poskytnem konzultácie, metodickú príručku a ďalšie učebné pomôcky.

⁸Operačné systémy (GNU/Linux), <http://www.shenk.sk/skola/informatika/operacne-systemy-gnu-linux.pdf>

3. Učiteľ

Aký by mal byť učiteľ informatiky? V prvom rade musí byť odborník v informatike. Inými slovami musí byť uplatniteľný v praxi, zamestnateľný ako informatik mimo školstva. To by malo platiť pre každého učiteľa. Každý učiteľ by mal byť odborníkom vo svojom odbore, zamestnateľným mimo školstva. Potrebujeme zlepšiť kvalitu prípravy nových učiteľov. Dobrým riešením je snaha o zvýšenie kvality školstva, kvality školy, kvality učiteľa, kvality žiaka, kvality vzdelania – kvalitnejšie vzdelanie pre učiteľa, odborná prax učiteľa mimo školy, zahraničné stáže, motivácia príplatkom k základnej tarife vo forme jednoznačne určeného výpočtu platu podľa vzdelania, kvality, tvorivosti, objektívne merateľných činností, a to jasne zadaným výpočtom v záväzných predpisoch [2, 3].

Učiteľ sa má hrať s motiváciou žiakov a má mať v triede rolu sprievodcu v poznávaní. Encyklopédie sú každému dostupné v knižnici alebo na internete. Vo vyučovaní má učiteľ použiť svoje encyklopedické znalosti na to, aby jednoducho, zrozumiteľne a krátko sformuloval to, čo literatúra poskytuje zložito. Učiteľ musí reagovať pohotovo. Učiteľ môže získať prirodzenú autoritu v novom predmete tým, že je odborník a chce ukázať žiakom, ako sa môžu stať odborníkmi aj oni. Netreba sa báť, že učiteľ stratí autoritu, ak dovolí žiakovi vedieť viac ako učiteľ. Učiteľ má skúsenosti, ktoré žiak ešte nemôže mať kvôli podstatne menšiemu veku [4, 5].

Nové vybavenie počítačovej učebne by malo byť kvalitné – výkonné, spoľahlivé, s dostatočnou životnosťou. Čím viac počítačov škola má, tým problematickejšia je správa počítačov a učební. Aby bolo možné spravovať počítače centralizovane automaticky, musia byť hardvérovo rovnaké alebo dostatočne podobné, s rovnakým nainštalovaným softvérom. Centralizovaná automatizovaná správa počítačov je tým ťažšia, čím väčší rozsah použitia počítače majú – viac predmetov, viac učiteľov, viac žiakov. Správca počítačovej učebne, správca počítačovej siete, IT technik – je pracovná náplň, ktorá by nemala byť kombinovaná s prácou učiteľa v jednej osobe. Majú odlišné zodpovednosti a právomoci. Preťažovaný učiteľ informatiky nemôže byť tvorivý, nemôže inovovať svoj predmet v takom tempe, ako potrebuje, trpí vyhorením, alebo odíde zo školy za lepším zamestnaním [6].

Záver

Prezentoval som svoje skúsenosti z troch rokov experimentálneho overovania v predmete *Operačné systémy* pre 2. ročník nového školského vzdelávacieho programu *Informačné a sieťové technológie*, v študijnom odbore 26 *Elektrotechnika*. Uplatnil som svoje pedagogické skúsenosti a tvorivosť tým, že som pre tento predmet napísal príslušnú časť školského vzdelávacieho programu, učebnicu a súvisiace dokumenty. V učebnici som sa zameral hlavne na operačný systém GNU/Linux.

Obsah predmetu a učebnice je obohatený o netradičné motivačné prvky a na konci je zaradená projektová úloha.

Slobodný a otvorený softvér by mal dominovať vo všetkých predmetoch využívajúcich výpočtovú techniku, ako aj v prevádzke školy. Dôvodov je niekoľko – ušetrenie peňazí, výchova k dodržiavaniu predpisov, podpora slobodného výberu a slobodného názoru, univerzálnosť a meniteľnosť softvérového vybavenia.

Literatúra

- [1] PETLÁK, E.: *Všeobecná didaktika*, Bratislava, IRIS, 1997, ISBN 80-88778-49-2.
- [2] National Research Council: *Advancing scientific research in education*, Washington, USA, The national academies press, 2004, ISBN 0-309-09321-X, ISBN 0-309-54598-6, <http://www.nap.edu>.
- [3] MONTALBANO, V. – PORRI, A.: *Attempts of transforming teacher practice trough professional development*, Italy, University of Siena, 2012.
- [4] LUCKIN, R. – NOSS, R.: *Decoding learning: The proof, promise and potential of digital education*, London, Nesta, 2012. 89 pp., http://www.nesta.org.uk/library/documents/DecodingLearningReport_v12.pdf, <http://phys.org/news/2012-11-technology-greatest-impact-classroom.html>.
- [5] European science foundation (ESF-SCSS): *The professionalisation of academics as teachers in higher education*, EU, ESF, 2012, ISBN 978-2-918428-88-6, <http://www.esf.org/publications.html>, report: Social Scientists call for more effective teaching in Higher Education, Phys. org., 2013, <http://phys.org/news/2013-01-socialscientists-effective-higher.html>.
- [6] ŠECHNÝ, M.: *Nové prístupy vo vyučovaní vybraných tém z informatiky na strednej škole*, Sborník príspevků mezinárodní vědecké konference Evropské pedagogické fórum 2012, Pedagogicko-psychologické aspekty výuky, Hradec Králové, MAGNANIMITAS, 2012, 2. vydání, ISBN 978-80-905243-2-3, ETTN 085-12-12016-12-2, <http://www.epf.econference.cz>.

Kontaktná adresa

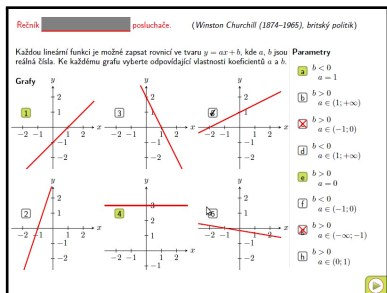
Mgr. Martin Šechný, SPŠE Prešov, Plzenská 1, 080 47 Prešov, Slovenská Republika,
E-mailová adresa: martin.sechny@shenk.sk

TEX A JEHO PRIATELIA

Hezký den TEXovým přátelům, účastníkům OSSConf a čtenářům sborníku, pokud jste se s TEXem ještě nesetkali, tak vězte, že se jedná o volně dostupný typografický systém od profesora Donalda E. Knutha ze Stanfordské univerzity. TEX je jádro (lze si to představit jako motor a kostru auta), obsahuje algoritmy užívané v sazbě, na které se nabalují formáty (další funkce auta) – pdfL^AT_EX, X_YL^AT_EX, LuaL^AT_EX a ConT_EXt. Vedle toho existují tzv. balíčky (angl. packages; příslušenství a výbava auta), na kterých pracují konkrétní jednotlivci či týmy programátorů. Hlavní repozitář tvoří ctan.org, ze kterého je každý rok seskládána TEXová distribuce. Nejznámější jsou TEX Live (repozitář: texlive-full, případně i texworks), MacTEX a proTEXt (úspornější varianta vůči TEX Live). To běžný TEXista používá ke své každodenní práci. Je to plný dílenský ponk pro typografy, amatérské i profesionální.

I letos

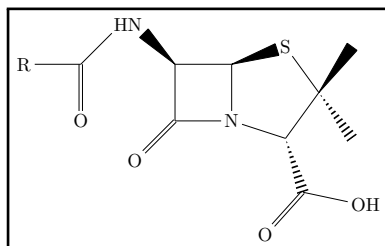
si představíme některé balíčky. Pozvanou přednášku přednese Robert Mařík, jeden z českých vývojářů. Pohovoří o AcroTEXu a balíčcích ocg-p, ocgtools, fancytooltips, skriptu fancy-preview, balíčku dps a okrajově zmíní i další pomůcky. Máme se na co těšit, pro začínajícího TEXistu to budou zajímavé informace co TEX umí, pro zkušenějšího TEXistu pak informace, co už TEX umí.



I letos budeme moci mezi námi přivítat Jiřího Rybičku. Představí nám nejjednodušou sazbu hudebnin pomocí MusiXTEXu a balíčku musixlyr. Přednášku doplní Pavel Stríž, i s pomocnicí Terezkou, ukázkou práce v programu MuseScore.

Také bude představen vyvíjený antiplagiátorský systém na mendelu.cz, i to TEXové přátele zajímá, protože kvalifikační práce se v TEXu píše.

Jedna z velkých typografických oblastí je sazba v chemii. Na tuto oblast poukáže osvětový příspěvek Aleše Kozubíka o balíčku ChemFig. Rudolf Blaško nám představí jak pracuje s cykly v T_EXu a zmíní některé praktické ukázky.



$$\int_1^{2.1} x^2 dx = \left[\frac{x^3}{3} \right]_1^{2.1} = \frac{2.1^3 - 1^3}{3} = 4.612025.$$

$x \in (1, 2.1) \Rightarrow |x| \leq 2.1, f(x) = x^2, f'(x) = 2x, f''(x) = 2, |f'(x)| = 2|x| \leq 3x^2 = 13.23 =: \delta_1,$
 $f''(x) = 2 \cdot 2x, |f''(x)| = 4|x| \leq 6\delta_1 = 12.6 =: \delta_2, f'''(x) = 0, f^{(4)}(x) = 0, |f^{(4)}(x)| = 0 =: \delta_3,$

Oběhová metoda Chyba: $R_n \leq \delta_1 \frac{(b-a)^2}{2n^2} \Rightarrow R_n \leq \frac{3.21 \cdot 1.1^2}{2n^2} = 3.20166.$

Lichoběžková metoda Chyba: $R_n \leq \delta_2 \frac{(b-a)^3}{24n^3} \Rightarrow R_n \leq \frac{62.1 \cdot 1.1^3}{24n^3} = 0.055902.$

Simpsonova metoda Chyba: $R_n \leq \delta_3 \frac{(b-a)^4}{720n^4} = 0$ pro všechny $n \in \mathbb{N}$, t. j. metoda je přesná a iterativní G4 ne.

Oběhová metoda			Lichoběžková metoda			Simpsonova metoda		
n	R_n	δ_n	n	R_n	δ_n	n	R_n	δ_n
1	1.22	1.815848	1	1.22	3.031096	1	1.3823	4.017861
2	1.41	1.302984	2	1.41	1.575308	2	1.2607	2.101259
3	1.46	4.574296	3	1.46	0.848562	3	1.25	14.8855
4	1.86	0.648472	4	1.86	0.382044	4	1.2333	15.615407
5	2.1	0.261	5	2.1	0.261	5	1.9167	29.164372

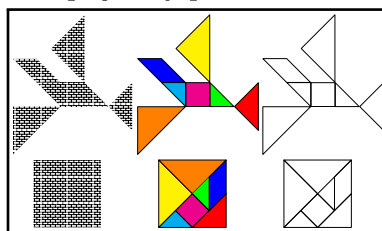
Společně	17.0208	25.2818	Společně	42.3020	Společně	75.0085
$f =$ Oběhová	3.764576	2.201096	$f =$ Oběhová	6.032096	$f =$ Oběhová	4.612025
Přesnost:	4.612025	4.612025	Přesnost:	4.612025	Přesnost:	4.612025
Chyba:	-0.907549	-0.949993	Chyba:	-0.949281	Chyba:	0

Pavel Stříž poukáže na některé typografické problémy, které prakticky řešil.

S Tiborem Menyhártem jsme řešili sazbu symbolů barev, listů, licitace, sehrávky a bridžových diagramů v T_EXu přes WP-QuickL^AT_EX ve webovém prostředí WordPress. Bude naznačen i řešený problém stahování odehraných bridžových rozdání na BBO, ten pravděpodobně zazní ve vývojářské sekci.

U divadelních ochotníků se řešil problém umístování symbolů do vysázené divadelní hry (PDF), a především, zda-li existuje možnost spustit dávkové soubory (hudbu, zvuky a zvukové efekty) přímo z PDF v reálném čase během probíhajícího představení, viz detaily ke studované hře.

S pomocníčkem Kristýnkou představíme i sazbu tangramů, zdá se, že je to příjemný typografický problém jako samotná čínská skládačka, ale zas tak příjemný problém to není.



Od T_EXového stolku se na Vás v Žilině těší,

Pavel Stříž

Pavel Stříž
Bučovice, 21. 6. 2013

CYKLY VO FORMÁTE \LaTeX

RUDOLF BLAŠKO (SK)

Abstrakt. Typografický systém \TeX a špeciálne \LaTeX má svoje nepopierateľné výhody a ostatné systémy na písanie dokumentov (komerčný *Microsoft Office*, resp. voľne šíriteľné *OpenOffice.org*, *LibreOffice*) mu len ťažko môžu konkurovať. Samozrejme aj tieto systémy majú svoje výhody, ale sú orientované skôr ako sekretárske aplikácie a nie sú príliš vhodné na písanie vedeckých a odborných publikácií, ktoré obsahujú špeciálne vzorce (napr. matematické, chemické, ...). Tieto vzorce sa tam dajú vytvoriť. Ale vynaložené úsilie nezodpovedá výslednej kvalite. V predložennom príspevku ukážeme postupy ako v \LaTeX -u vytvoriť vlastné príkazy a následne zautomatizovať ich hromadné použitie pomocou príkazov na tvorenie cyklov (napr. `\foreach`, `\multido`).

Kľúčové slová. \LaTeX , otvorený softvér, cyklus, `foreach`, `multido`.

CYCLES IN THE \LaTeX

Abstract. Typesetting system \TeX and especially \LaTeX has its undeniable advantages and other systems for documents preparation (e.g. commercial *Microsoft Office*, or freely available *OpenOffice.org* or *LibreOffice*) can hardly compete it. Of course, these systems have their advantages too, but are oriented rather to office application and are not well suited for writing scientific and technical publications that include special formulas (such as mathematical, chemical, ...). These formulas can be created even in these systems. But the effort does not correspond to the resulting quality. In the present paper we demonstrate the processes of custom commands creation in \LaTeX and then automate their mass use by cycle commands (e.g. `\foreach`, `\multido`).

Key words and phrases. \LaTeX , open source, cycle, `foreach`, `multido`.

Úvod

V súčasnej dobe stále drvivá väčšina užívateľom používa komerčné a, uvedomejšie užívateľia, aj nekomerčné kancelárske programy (*Microsoft Office*, voľne šíriteľné *OpenOffice.org*, *LibreOffice*). Keďže principiálne je medzi nimi iba jediný rozdiel, a to je platenie za ich používanie, budem tieto systémy spoločne označovať *Office*. Veľkou výhodou týchto programov je veľká otvorenosť a užívateľská prívetivosť týchto programov: „Otvorím nový dokument a hneď môžem začať písať. Keď chcem niečo zmeniť iba si to jednoducho myškou naklikám.“. A v tom je väčšinou problém, málo používané funkcionality programu musí aj skúsený užívateľ hľadať. Toto hľadanie nemusí byť triviálne, najmä ak človek momentálne používa inú verziu od tej, na ktorú je zvyknutý. Keďže každý používa tieto programy systémom otvorím a píšem, skoro nikto neovláda ich zákonitosti a

väzby. Potom nastávajú situácie, že tlač vyzerá ináč ako text na monitore, miešajú a objavujú sa nechcené fonty písom, problém je s hlavičkou a obsahom, . . . Väčšina týchto problémov vyplýva z neinformovanosti užívateľa a z lákavej vidiny, že môže hneď písať a „tváriť sa ako typograf“. Tieto systémy dokážu veľa, ale keď chce niekto kvalitný výstup, potrebuje o nich niečo nemálo vedieť — a nie je to iba klikanie myškou.

Napísať jednoduchý naformátovaný text (kapitoly, odrážky, tabuľky) nie je v týchto programoch problém. Problémy nastávajú pri písaní špeciálnych výrazov (matematiké, fyzikálne, chemické a vo všeobecnosti technické vzorce). Tieto výrazy sa dajú jednoducho vytvoriť pomocou klikacích prednastavených šablón. Problém nastáva pri ich úpravách — užívateľ musí dobre ovládať myš. Nezanedbateľná je aj ich vizuálna podoba a problém zladenia rôznych fontov rovnakých premenných (napr. x_i) v texte a v uvedených výrazoch. Ale keď nie je užívateľ príliš náročný, tieto výstupy postačia.

Uvažujme hypotetickú situáciu, že organizujeme nejakú akciu (napr. konferenciu) a potrebujeme vygenerovať menovite pre väčší počet ľudí reprezentatívny dokument (napri certifikát). Uvedený problém v predchádzajúcich systémoch Office je riešiteľný pomocou systému hromadnej korešpondencie, pričom je možné prepojenie aj na externé databázy a údaje sa dajú modifikovať podľa požiadaviek (napr. filtrovanie, radenie a agregovanie vstupnej databázy). Systém, bol vytvorený pre tvorbu hromadných dopisov, adresných a iných štítkov, či vizitiek. Na tlač certifikátov, alebo iných dokumentov, v ktorých potrebujeme meniť údaje podľa vstupnej databázy, sa dá samozrejme tiež použiť, ale stále je to o klikaní. Makrá sa v MSOffice dajú tvoriť ako postupnosti klávesových príkazov a kliknutí myškou. V skutočnosti sa tieto príkazy zapisujú ako volania metód jednotlivých objektov GUI príslušného programu. Zápis prebieha priamo v skriptovacom objektovom jazyku Visual Basic. Takto uložené makrá môžeme jednoducho modifikovať a doplniť ich o jednoduché cykly, vlastné výpočty, ap.

V typografickom systéme \LaTeX je tento problém oveľa jednoduchší. Vytvoríme si vlastný príkaz (tzv. makro), ktorý bude obsahovať celý certifikát a namiesto mena vložíme parameter, ktorý sa pri vlastnom spustení makra nahradí menom. Takže nepotrebujeme pre každého účastníka vytvárať celý certifikát, vytvoríme ho iba raz a \LaTeX za nás meno nahradí. Zanietení labužník systému Office môže namietat, že takto to funguje aj u nich. V podstate má pravdu, ale je tu jeden „nepatrný rozdiel“. V \LaTeX -u je to jednoduchšie, postačí v zdrojovom súbore napísať uvedené meno ako argument príkazu. Predpokladajme, že je certifikát definovaný v makre s názvom `\OSScert`, potom stačí napísať:

```
\OSScert{Rudolf Blaško}\OSScert{Jozko Hruška}\OSScert{Michal Starec}
. . . .
\OSScert{Iné Meno}\OSScert{Posledné Meno}
```

Prípadne môžeme použiť cyklus. V tomto prípade sa dá s úspechom použiť cyklus `\foreach`. Jeho použitie by mohlo vyzeráť napr. nasledovne:

```
\foreach \MENO in {{Rudolf Blaško}, {Jožko Hruška}, {Michal Starec},
  < ostatné mená, ... > {Iné Meno}, {Posledné Meno}}
{\OSScert{\MENO}}
```

1. Makrá systému L^AT_EX

Na vytvorenie vlastného makra, t. j. vlastného príkazu sa používajú príkazy `\def`, resp. `\newcommand` a ich modifikácie. Každý z nich má svoje výhody a špecifické možnosti.

Príkaz `\def` je jedným zo základných príkazov systému T_EX a používa sa v tvare `\def<názov príkazu> <maska parametrov> {telo definície}`. Príkaz má lokálny charakter, to znamená, že ak ho použijeme vo vnútri nejakého prostredia, resp. ohraničujúcich zátvoriek `{...}`, mimo sa jeho význam stratí. Tento príkaz neoveruje, či už nejaké makro alebo príkaz s identickým názvom existuje, takže v kladnom prípade ho nahradí novým makrom. V názve (analogicky ako v celom T_EX-u a L^AT_EX-u) sa rozlišujú malé a veľké písmená.

Časť `<maska parametrov>` je nepovinná a určuje spôsob práce práve definovaného makra s parametrami. Parametrov môže byť až 9 a musia nasledovať vzostupne za sebou od #1 až po posledný napr. #6. Parametre môžu byť oddelené a obložené rôznymi oddeľovačmi a maska končí prvým výskytom znaku `{`.

Pri použití makra sa požaduje zápis parametrov presne v takom istom tvare, ako je uvedený v maske parametrov. Príkaz sa môže použiť aj bez parametrov, napr. zápis `\def\Konf{KonFERerencia začína!}` definuje nový príkaz `\Konf`, ktorý je pri každom použití nahradený vetou „KonFERerencia začína!“ Ak upravíme makro na tvar `\def\Konf{\it KonFERerencia začína!}`, potom sa daná veta vypíše kurzívou „*KonFERerencia začína!*“. *Problém je iba v tom, že sa kurzívou vypíše aj nasledujúci text za príkazom. Správne definované makro má mať tvar* `\def\Konf{{\it KonFERerencia začína!}}`. To znamená, že je potrebné zdvojiť zátvorky, aby sa kurzíva obmedzila iba na danú vetu, t. j. na skupinu ohraničenú vnútornými zátvorkami, vonkajšie zátvorky sú súčasťou predpísaného tvaru príkazu `\def`.

Ak chceme pridať napr. názov konferencie a dátum, môžeme makro upraviť nasledovne `\def\Konf[#1|#2]{\it KonFERerencia #2 začína #1!}`. Po takejto definícii musíme príkaz `\Konf` používať s maskou `[názov|dátum]`. Ak nebude za slovom `\Konf` nasledovať symbol `[` (resp. `□`), L^AT_EX vyhlási chybové hlásenie. Po zaevidovaní tohto symbolu pokračuje L^AT_EX v načítavaní parametra #1 až do prvého výskytu symbolu `|`, kedy začne načítavať parameter #2 až do prvého výskytu symbolu `]`. Príkazom `\Konf[2.7.2013|OssConf]` dostaneme vetu „*KonFERerencia OSSConf začína 2.7.2013!*“.

Makrá môžeme do seba vnárať, t. j. v definovanom makre môžeme opäť definovať nové makro. Ak používame parametre, potom sa symbol # vo vnútornom makre zdvojuje. Musíme ale rozlišovať medzi parametrami vonkajšieho a vnútorného makra. Ukážeme to na jednoduchom príklade. Definujme makro `\TRANS`:

```
\def\TRANS#1#2#3#4{\def\ptm##1##2{##1\to##2}
  $\ptm{#1}{#2}$, $\ptm{#2}{#3}$, $\ptm{#3}{#4}$, $\ptm{#4}{#1}$}.
```

Zadaním príkazu `\TRANS{\alpha}{\beta}{\gamma}{\delta}` dostaneme výpis „ $\alpha \rightarrow \beta, \beta \rightarrow \gamma, \gamma \rightarrow \delta, \delta \rightarrow \alpha$ “ v matematickom móde. Ako sme videli, jednotlivé parametre sú oddelené v zátvorkách. Teoreticky by sme mohli zátvorky vynechať a použiť zápis `\TRANS\alpha\beta\gamma\delta`, ale kvôli prehľadnosti sa to nedoporučuje. Ak by sme napísali `\TRANS alpha\beta\gamma\delta` tak, by nám \LaTeX vyhlásil chybu. Prvé štyri znaky `a l p h` by použil ako parametre príkazu, posledné `a` by ešte napísal, ale narazil by na príkaz `\beta`, ktorý vyžaduje matematický mód (ten je zavedený pri definícii makra `\TRANS`).

Ako sme už spomínali, príkaz `\def` má lokálny charakter, ak chceme aby definovaný príkaz mal globálny charakter, t. j. aby platil aj mimo definované prostredie, môžeme použiť príkaz `\gdef` alebo `\def` s prefixom `\global`. Uvedené príkazy sú identické, t. j. `\gdef` a `\global\def` znamenajú to isté.

Ďalším príkazom, ktorý môžeme tvoriť vlastné makrá je `\newcommand`, ktorý funguje trochu ináč ako príkaz `\def`. Tento príkaz kontroluje, či príkaz alebo makro (systémové alebo vlastné) s takýmto menom existuje. V kladnom prípade hlási \LaTeX chybu. Ak napriek tomu chceme príkaz predefinovať, musíme použiť `\renewcommand`. Príkazy `\newcommand`, `\renewcommand` majú rovnakú formu zápisu `\newcommand<názov príkazu> <parametre> {telo definície}`.

Ak novovytvorený príkaz neobsahuje parametre, potom je jeho definícia rovnaká ako pri príkaze `\def`. Príkazy `\newcommand\Konf{KonFERerencia začína!}` a `\def\Konf{KonFERerencia začína!}` majú rovnaký význam a ich výstup je identický. Definíciu `\def\Konf[#1|#2]{\it KonFERerencia #2 začína #1!}` môžeme zapísať `\newcommand\Konf[2]{\it KonFERerencia #2 začína #1!}`. To znamená, že pri použití príkazu `\newcommand` nezadáваме masku parametrov, ale iba ich počet v zátvorkách `[]`.

Navyše môžeme parameter `#1` implicitne definovať ako prednastavený. Jeho hodnota sa definuje taktiež v hranatých zátvorkách hneď za počtom premenných. Pri použití sa prvý parameter zadáva v hranatých zátvorkách `[]`, alebo sa vôbec neuvádza a dosadí sa zaň implicitná hodnota. Ostatné parametre počnúc parametrom `#2`, pokiaľ existujú, sa zadávajú rovnako ako pri príkaze `\def`. V príklade `\newcommand\Konf[2][2.7.2013]{\it KonFERerencia #2 začína #1!}` je hodnota parametra `#1` implicitne nastavená na hodnotu `2.7.2013`. To znamená, že príkazy `\Konf[2.7.2013]{OssConf}` a `\Konf{OssConf}` majú rovnaký význam.

2. Príkazy cyklov `\foreach` a `\multido`

V predchádzajúcej časti sme ukázali ako sa definujú v L^AT_EX-u vlastné príkazy a makrá. Na ich hromadné používanie s viac-menej úspešným výsledkom môžeme použiť cykly. Cykly sa dajú tvoriť aj na základnej, tzv. plain úrovni T_EX-u (viď napr. [5]). Užitočnejšie je použiť nejaký hotový balíček. V tejto časti sa budeme zaoberať príkazom `\foreach`. Na jeho použitie musíme v preambule použiť `\usepackage{pgffor}`. Ak používame TiKZ, potom sa balíček `pgffor` načíta automaticky [1]. Príkaz sa používa v tvare `\foreach \prem {zoznam}{príkazy}`. Príkaz má povinný parameter `\prem`, ktorý postupne nadobúda všetky hodnoty z množiny `zoznam` a dosadzuje ich do príkazov `príkazy`.

Množina `zoznam` sa zadáva vymenovaním prvkov oddelených čiarkou. Pri číselných premenných môžeme použiť konštrukciu `{začiatok, ..., koniec}`. Premenná `\prem` sa postupne nahrádza číslami `začiatok`, `začiatok+1`, `začiatok+2`, `začiatok+3` až po maximálnu hodnotu, ktorá nie je väčšia ako `koniec`. Napríklad príkazom `\foreach \xx in {1.1, ..., 10} { \xx}` sa postupne vypíšu čísla 1.1 2.1 3.1 4.1 5.1 6.1 7.1 8.1 9.1. Rovnaký efekt by sme dosiahli tiež oveľa pracnejším príkazom `\foreach \xx in {1.1, 2.1, 3.1, 4.1, 5.1, 6.1, 7.1, 8.1, 9.1} { \xx}`.

Inou možnosťou je použiť cyklus `\multido` z rovnomenného balíčka `\multido`. Používa sa v tvare `\multido{premenné}{počet opakovaní}{príkazy}`. V povinnom parametri `premenné` sa deklaruje typ, počiatočná hodnota a prírastok premennej, ktorá je výhradne číslom a používa sa v parametri `príkazy`. Definuje sa v tvare `\prem=poč.hodnota+prírastok`. Ak je prírastok záporný, píše sa v tvare `\prem=poč.hodnota+-prírastok`. Prvé písmeno názvu `\prem` určuje jej typ:

Dimension (d D) — dĺžková miera, výsledok je určený v jednotkách `sp`. Napr. `\multido{\dx=4cm+-1em}{2}{ [\dx]}` dáva výstup [7458719sp] [6745171sp].

Number (n N) — počiatočná hodnota a prírastok sú čísla s rovnakým počtom desatinných miest (maximálne 8), pričom počiatočná hodnota môže byť tiež celým číslom, výsledkom je číslo s rovnakým počtom desatinných miest, ale fixovaným. Napr. `\multido{\nx=4+1.05}{2}{ [\nx]}` dáva výstup [4] [5.05] a `\multido{\nx=4.05+1.05}{2}{ [\nx]}` dáva výstup [4.05] [5.10].

Integer (i I) — počiatočná hodnota a prírastok sú celé čísla, výsledkom je celé číslo. Napr. `\multido{\ix=4+1}{2}{ [\ix]}` dáva výstup [4] [5].

Real (r R) — počiatočná hodnota a prírastok sú čísla s maximálne 4 desatinnými miestami, výsledkom je reálne číslo s odrezanými nulami za desatinnou bodkou. Tento typ je rýchlejší ako **Number**, ale s relatívne početnými numerickými výpočtovými chybami. Napr. `\multido{\rx=4+1.05}{2}{ [\rx]}` dáva výstup [4.0] [5.05] a `\multido{\rx=4.05+1.05}{2}{ [\rx]}` dáva výstup [4.05] [5.1] a `\multido{\rx=2+3.05}{6}{ [\rx]}` dáva výstup [2.0] [5.05] [8.1] [11.15001] [14.20001] [17.25002].

Príkaz môžeme použiť aj bez premennej, napr. `\multido{8}{\LaTeX{}}` a dostaneme `\LaTeX \LaTeX \LaTeX \LaTeX \LaTeX \LaTeX \LaTeX \LaTeX`.

Približne (s presnosťou $\varepsilon = 0.1$) vypočítajte $\int_{2.1}^1 x^3 dx$.

$\int_{2.1}^1 x^3 dx = - \int_1^{2.1} x^3 dx, \int_{2.1}^1 x^3 dx = \left[\frac{x^4}{4}\right]_{2.1}^1 = \frac{1^4 - 2.1^4}{4} = -4.612025.$

$x \in (1, 2.1) \Rightarrow |x| \leq 2.1 := \delta, f(x) = x^3, f'(x) = 3x^2, |f'(x)| = 3|x|^2 \leq 3\delta^2 = 13.23 := \delta_1,$
 $f''(x) = 3 \cdot 2x, |f''(x)| = 6|x| \leq 6\delta = 12.6 := \delta_2, f'''(x) = 6, f''''(x) = 0, |f''''(x)| = 0 := \delta_4.$

Obdĺžniková metóda Chyba: $R_n \leq \delta_1 \frac{(b-a)^2}{n} = \frac{3(2.1-1)^2 \delta^2}{n} \leq 0.1$
 $\Rightarrow n \geq \frac{3(2.1-1)^2 \delta^2}{0.1} = 160.083 \Rightarrow$ teoreticky $n = 161$, prakticky $n = 100.$

Lichobežníková metóda Chyba: $R_n \leq \delta_2 \frac{(b-a)^3}{12n^2} = \frac{6(2.1-1)^3 \delta}{12n^2} \leq 0.1$
 $\Rightarrow n^2 \geq \frac{6(2.1-1)^3 \delta}{12 \cdot 0.1} = 13.9755 = 3.738382^2 \Rightarrow n = 4.$

Simpsonova metóda Chyba: $R_n \leq \delta_4 \frac{(b-a)^5}{180n^4} = 0$ pre všetky $n \in \mathbb{N}$, t. j. metóda je presná a nezávisí od $n \Rightarrow n = 2.$

Obdĺžniková metóda			Lichobežníková metóda		Simpsonova metóda	
$x_0: 1$	$y_0 = 1$		$x_0: 1$	$y_0 = 1$	$x_0: 1$	$y_0 = 1$
$x_1: 1.011$	$y_1 = 1.033364$	$y_1 = 1.033364$	$x_1: 1.275$	$2y_1 = 4.145344$	$x_1: 1.55$	$4y_1 = 14.8955$
$x_2: 1.022$	$y_2 = 1.067463$	$y_2 = 1.067463$	$x_2: 1.55$	$2y_2 = 7.44775$	$x_2: 2.1$	$y_2 = 9.261$
$x_3: 1.033$	$y_3 = 1.102303$	$y_3 = 1.102303$	$x_3: 1.825$	$2y_3 = 12.156781$		
$x_4: 1.044$	$y_4 = 1.137893$	$y_4 = 1.137893$	$x_4: 2.1$	$y_4 = 9.261$		
$x_5: 1.055$	$y_5 = 1.174241$	$y_5 = 1.174241$				
$x_6: 1.066$	$y_6 = 1.211356$	$y_6 = 1.211356$				
.....						
$x_{97}: 2.067$	$y_{97} = 8.831235$	$y_{97} = 8.831235$				
$x_{98}: 2.078$	$y_{98} = 8.972979$	$y_{98} = 8.972979$				
$x_{99}: 2.089$	$y_{99} = 9.116231$	$y_{99} = 9.116231$				
$x_{100}: 2.1$	$y_{100} = 9.261$	$y_{100} = 9.261$				
Spolu s:	415.153877	423.414877	Spolu s:	34.010875	Spolu s:	25.1565
$f \approx \frac{(1-2.1)^2}{100}:$	-4.566693	-4.657564	$f \approx \frac{(1-2.1)^2}{34}:$	-4.676495	$f \approx \frac{(1-2.1)^2}{34}:$	-4.612025
Presne:	-4.612025	-4.612025	Presne:	-4.612025	Presne:	-4.612025
Chyba:	+ 0.045332	- 0.045539	Chyba:	- 0.06447	Chyba:	+ 0

Obr. 1. Približne (s presnosťou $\varepsilon = 0.1$) vypočítajte $\int_{2.1}^1 x^3 dx$.

3. Príklad na numerické integrovanie

Na záver článku uvedieme jednoduché makro, ktoré dokáže numericky spočítať integrál $\int_a^b x^k dx$. Makro má 4 parametre `\RdroCB[n]{a}{b}{k}`, pričom prvý parameter určuje počet deliacich bodov výpočtu (maximálne $n = 100$). Ak je táto hodnota desatinné číslo z intervalu $(0, 1)$, potom sa považuje za presnosť a počet deliacich bodov sa dopočíta. Pri výpočte sa zobrazí tabuľka s deliacimi bodmi a prisluchajúcimi iteračnými hodnotami. Záver tabuľky tvorí presná hodnota integrálu, vypočítaná približná hodnota a chyba výpočtu (obr. ??, ??). Pri výpočte bol použitý cyklus `\foreach` spolu s balíčkom `fp` (Fixed Point package — výpočty v pevnej rádovej čiarky), ktorý dovoľuje používať matematické výpočty priamo v zdrojovom kóde `\LaTeX`-u a s balíčkom `xifthen`, ktorý umožňuje vetviť makrá do rôznych logických častí na základe ich argumentov. K makru na riešenie úlohy prináleží makro s identickými parametrami, ktoré zadáva problém.

```
%-----
\newcommand\droCB[4][.1]{\tjP[\riA]{-(#1)}\tjP[\riB]{(#1)-1}\def\ak{\ifthenelse}%
\def\ZLE{\color{red}BAD:} $\k=#2$, $a=#3$, $b=#4$ {\color{red}OK:} $a\ne b$}%
```

```

\tjI[\Xa]{#2}\tjP[\Xb]{(#2)-1}\tj0[\Xc]{(#4)-(#3)}%
\rbPr{\rALL}{(\Xa)*(\Xb)*((\Xc)-1)}%
\ak{\rALL=0}{\ZLE}{Približne \ak{\riA=0 \and \riB=0}{\rbPr{\rI}{max(#1,.001)}
(s~presnosťou~\varepsilon=\riI$)}{} vypočítajte

$$\int_{\rb{#3}}^{\rb{#4}} x^{\rb{JExp{#2}}}{\rm d}x$$

\ak{\riB=1}{\rbPr{\gN}{trunc(min(max(#1,5),100),0)}
(použite aspoň  $\gN$  deliacich bodov)}{}.)}

```

Výpočty a testovania boli riešené pomocou vlastných makier, napr. na testovanie celočíselnosti alebo kladnosti hodnôt boli použité príkazy (výsledkom je logická hodnota 0 alebo 1):

```

\newcommand\tjI[2][\rrT]{\rbx{#2}%
\FPifint{\rbV}\FPset{#1}{1}\else\FPset{#1}{0}\fi}
\newcommand\tjP[2][\rrT]{\rbx{#2}%
\FPifpos{\rbV}\FPset{#1}{1}\else\FPset{#1}{0}\fi}

```

Kompletný zdrojový kód obidvoch makier a všetkých pomocných testovacích a výpočtových makier je zverejnený na <http://frcatel.fri.uniza.sk/~beerb/latex/num-in.pdf>. Pri výpočte v cykle bolo potrebné si zapamätať medzivýpočty, t. j. definovať globálne premenné a zabezpečiť ich korektné sčítavanie. To bolo riešené pomocou príkazu priradenia `\let` a pomocných premenných.

```

\foreach \gX in {\gZ, ..., \gK} %%%..... VLASTNY VYPOCET
{\rbPr[8]{\Go}{(\DH)+(\gX)*((\HH)-(\DH))/(\So)}\ak{\gX<\SoJ}{\xxD{\rbPr{\bo}{1}}%
{\rbPr{\bo}{sgn(\Go)}}\rbPr[8]{\Yo}{(\bo)*abs(\Go)^(#2)}{\rbPr[8]{\Yo}{0}}%
\rbPr[8]{\G1}{(\DH)+(\gX)*((\HH)-(\DH))/(\S1)}\ak{\gX<\S1J}{\xxD{\rbPr{\bl}{1}}%
{\rbPr{\bl}{sgn(\G1)}}\rbPr[8]{\Y1}{(\bl)*abs(\G1)^(#2)}{\rbPr[8]{\Y1}{0}}%
\rbPr[8]{\Gs}{(\DH)+(\gX)*((\HH)-(\DH))/(\Ss)}\ak{\gX<\SsJ}{\xxD{\rbPr{\bs}{1}}%
{\rbPr{\bs}{sgn(\Gs)}}\rbPr[8]{\Ys}{(\bs)*abs(\Gs)^(#2)}{\rbPr[8]{\Ys}{0}}%
\rbPr{\QL}{0}\rbPr{\QP}{0}\rbPr{\Q1}{0}\rbPr{\Qs}{0}%
\ak{\gX<\SoJ}{\rbPr{\QL}{1}}{\ak{\gX>0 \AND \gX<\SoJ}{\rbPr{\QP}{1}}}%
\rbPrG[8]{\VL}{(\hL)+(\QL)*(\Yo)}\global\let\hL\VL%
\rbPrG[8]{\VP}{(\hP)+(\QP)*(\Yo)}\global\let\hP\VP%
\ak{\gX<\S1J}{\rbPr{\Q1}{2}}\ak{\gX=0 \OR \gX=\S1}{\rbPr{\Q1}{1}}{}%
\rbPrG[8]{\V1}{(\h1)+(\Q1)*(\Y1)}\global\let\h1\V1%
\ak{\gX<\SsJ}{\rbPr{\Qs}{4}}\ttI{(\gX)/2}{\rbPr{\Qs}{2}}{}%
\ak{\gX=0 \OR \gX=\Ss}{\rbPr{\Qs}{1}}{}{}%
\rbPrG[8]{\Vs}{(\hs)+(\Qs)*(\Ys)}\global\let\hs\Vs%

\ak{\gX<\SoJ}{\xxT2{\hfill\!|\!$x_{\gX}\!|:$& %%%..... VLASTNE TABULKY VYPOCTOV

$$\rb[4]{\Go}$$


$$\ak{\gX=\So}{\hfill y_{\gX}=\$&}$$


$$\rb[6]{\Yo}$$


$$\ak{\gX=0}{\hfill y_{\gX}=\$&}$$


$$\rb[6]{\Yo}$$


$$\xxT2{\&&&&}}$$

\hfill\ak{\gX<\S1J}{\xxT1{\hfill\!|\!$x_{\gX}\!|:$&

$$\rb[4]{\G1}$$


$$\hfill\rbJ{\Q1}y_{\gX}=\$&}$$


$$\rb[6]{(\Q1)*\Y1}$$


$$\xxT1{\&&&}}$$

\hfill\ak{\gX<\SsJ}{\xxT1{\hfill\!|\!$x_{\gX}\!|:$&

$$\rb[4]{\Gs}$$


$$\hfill\rbJ{\Qs}y_{\gX}=\$&}$$


$$\rb[6]{(\Qs)*\Ys}$$


$$\xxT1{\&&&}}$$

}%%..... KONIEC \foreach

```

Záver

Príkazy na tvorenie cyklov spolu s rôznymi ďalšími balíčkami značne rozširujú možnosti L^AT_EX-u. V článku je uvedený postup ako využiť cykly pri numerickom integrovaní. V spojení s balíčkom TikZ zase dovoľujú kresliť rozličné netriviálne a mnohorozvetvené útvary [1].

Približne vypočítajte $\int_1^{2.1} x^3 dx$ (použite aspoň 5 deliacich bodov).

$\int_1^{2.1} x^3 dx = \left[\frac{x^4}{4} \right]_1^{2.1} = \frac{2.1^4 - 1^4}{4} = 4.612025.$

$x \in (1, 2.1) \Rightarrow |x| \leq 2.1 := \delta, f(x) = x^3, f'(x) = 3x^2, |f'(x)| = 3|x|^2 \leq 3\delta^2 = 13.23 := \delta_1,$
 $f''(x) = 3 \cdot 2x, |f''(x)| = 6|x| \leq 6\delta = 12.6 := \delta_2, f'''(x) = 6, f''''(x) = 0, |f''''(x)| = 0 := \delta_4.$

Obdĺžniková metóda Chyba: $R_n \leq \delta_1 \frac{(b-a)^2}{n} \Rightarrow R_5 \leq \frac{3(2.1-1)^2 \delta^2}{5} = 3.20166.$

Lichobežníková metóda Chyba: $R_n \leq \delta_2 \frac{(b-a)^3}{12n^2} \Rightarrow R_5 \leq \frac{6(2.1-1)^3 \delta}{12 \cdot 5^2} = 0.055902.$

Simpsonova metóda Chyba: $R_n \leq \delta_4 \frac{(b-a)^5}{180n^4} = 0$ pre všetky $n \in \mathbb{N}$, t. j. metóda je presná a nezávisí od n .

Obdĺžniková metóda			Lichobežníková metóda			Simpsonova metóda		
$x_0: 1$	$y_0 = 1$		$x_0: 1$	$y_0 = 1$		$x_0: 1$	$y_0 = 1$	
$x_1: 1.22$	$y_1 = 1.815848$	$y_1 = 1.815848$	$x_1: 1.22$	$2y_1 = 3.631696$		$x_1: 1.1833$	$4y_1 = 6.627981$	
$x_2: 1.44$	$y_2 = 2.985984$	$y_2 = 2.985984$	$x_2: 1.44$	$2y_2 = 5.971968$		$x_2: 1.3667$	$2y_2 = 5.105259$	
$x_3: 1.66$	$y_3 = 4.574296$	$y_3 = 4.574296$	$x_3: 1.66$	$2y_3 = 9.148592$		$x_3: 1.55$	$4y_3 = 14.8955$	
$x_4: 1.88$	$y_4 = 6.644672$	$y_4 = 6.644672$	$x_4: 1.88$	$2y_4 = 13.289344$		$x_4: 1.7333$	$2y_4 = 10.415407$	
$x_5: 2.1$	$y_5 = 9.261$	$y_5 = 9.261$	$x_5: 2.1$	$y_5 = 9.261$		$x_5: 1.9167$	$4y_5 = 28.164352$	
						$x_6: 2.1$	$y_6 = 9.261$	
Spolu s:	17.0208	25.2818	Spolu s:	42.3026		Spolu s:	75.4695	
$f \approx \frac{(2.1-1)s}{3}$:	3.744576	5.561996	$f \approx \frac{(2.1-1)s}{24}$:	4.653286		$f \approx \frac{(2.1-1)s}{36}$:	4.612025	
Presne:	4.612025	4.612025	Presne:	4.612025		Presne:	4.612025	
Chyba:	- 0.867449	+ 0.949971	Chyba:	+ 0.041261		Chyba:	+ 0	

Obr. 2. Približne vypočítajte $\int_1^{2.1} x^3 dx$ (požte aspoň 5 deliacich bodov).

Literatúra

- [1] BLAŠKO, R., KOZUBÍK, A.: *L^AT_EX, teória grafov a trochu umenia*, Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2012, Žilina, 2.–4. júla 2012, str. 85–90, ISBN 978-80-970457-2-2.
- [2] KNUTH, D. E.: *The T_EXbook*, 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*, Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2012, Žilina, 2.–4. júla 2012, str. 91–96, ISBN 978-80-970457-2-2.
- [4] KOZUBÍK, A., KOZUBÍKOVÁ, Z.: *Tabulkové výpočty v systéme L^AT_EX*, Otvorený softvér vo vzdelávaní, výskume a IT riešeníach, zborník medzinárodnej konferencie OSSConf 2012, Žilina, 2.–4. júla 2012, str. 97–102, ISBN 978-80-970457-2-2.
- [5] OLŠÁK, P.: *T_EXbook naruby*, Brno, Konvoj 1997, 468 str., ISBN 80-85615-64-9.

Kontaktná adresa

RNDr. Rudolf Blaško, PhD., Katedra matematických metód, Fakulta riadenia a informatiky, Žilinská univerzita, Univerzitná 8215/1, 010 26 Žilina, Slovenská republika,
E-mailová adresa: beerb@frcatel.fri.uniza.sk, <http://frcatel.fri.uniza.sk/~beerb/>

TÝDENNÍ DOVOLENÁ V ZEMI TANGRAMŮ

KRISTÝNKA HŮLKOVÁ (CZ) A PAVEL STRÍŽ (CZ)

Abstrakt. V článku se čtenář seznámí s jedním z přístupů sazby tangramů, staré čínské skládací hry. Autor zmiňuje úspěšné i neúspěšné pokusy kolem získání nekódovaných dat s větším počtem tangramů. Taktéž diskutuje v principu základní, ale i náročnější kroky nezbytné při sazbě tangramů a také kroky spojené s jejich tiskem. Mezi klíčové $\text{T}_{\text{E}}\text{X}$ ové balíčky v jeho práci patří `tikz`, `pdfpages` a `xcolor`.

Klíčová slova. $\text{T}_{\text{E}}\text{X}$, hry, skládačky, tangram.

A HOLIDAY WEEK IN THE LAND OF TANGRAMS

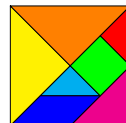
Abstract. The article presents one of the approaches of typesetting tangrams, an old Chinese dissection puzzle. The author mentions his successful and unsuccessful tries of finding suitable unencrypted tangram data of large quantity. He also discusses basic and advanced steps needed to undertake to typeset and print a sheet of tangrams. The most important packages for this typography task are `tikz`, `pdfpages`, and `xcolor` packages.

Key words and phrases. $\text{T}_{\text{E}}\text{X}$, Games, Puzzle, Tangram.

Úvod

Pokud v rámci diskuze někdo vysloví šachy, všichni ví, o čem je řeč, pokud někdo vyřkne bridž nebo go, tu a tam někdo ví, a když někdo vysloví termín tangram, tuší jen támhle pán vzadu a tady paní vpředu. Dokonce ani balík her `sgt-puzzles` tuto hru neobsahuje.

Nejedná se totiž formálně úplně o logickou hru (obsahuje vždy více řešení), ale o skládačku. Nám nejbližší přirovnání je puzzle. Zde však máme v typické sadě 7 dílků, kterými máme vyplnit předem daný prostor (na ukázce je základní čtverec, ale obrazců je celá řada). Počet dílků a obměna výplně obvykle určuje obtížnost hry. Hra vznikla v Číně a ve své době hrála roli dnešních IQ testů. Tangram skládá obvykle člověk sám, ale může to paralelně skládat více hráčů jako soutěž o nejrychlejšího. Je to hra vhodná pro děti od útlého věku.



Jak se sazeč na dovolené dostane k této sazbě? Poměrně jednoduše. Pomáhám divadelním ochotníkům a ti dělají různé akce pro děti. Tentokrát to byly tzv. dílničky. Čím mladší dítě, tím těžší matematický problém. Nápad se mi zalíbil a organizátorkám jsem poslal seznam myšlenek, co bych zvládl za 7 dní zrealizovat – malba na špunty od vína, karty, pexeso, vystřihovánky běžné či polyhedry, pak mě napadly pentomina a nakonec tangramy, neb k těm jsem se chtěl vrátit.

Jedna z organizátorek tangramy znala a ródeo kolem příprav tangramů začalo.

1. Hledání databáze tangramů – hon na data

Se sazbou některých her z dostupných dat (většinou ve formě textových souborů) jsem se setkal. Sázal jsem dvě knihy logických her (ISBN 978-80-87106-52-5 a ISBN 978-80-87106-53-2), založených především na balíku her `sgt-puzzles`, programu bludišť Daedalus a bridžových pětikartových koncovkách.

V linuxovém repozitáři najdeme programy `glpieces` a `gTans`. Měl bych po pravdě napsat, že to nebyl týden v zemi tangramů, ale rok a týden, neb před rokem jsem se snažil data rozluštit a nepodařilo se mi to tehdy, ani za pomoci jednoho z autorů. Popisky a názvy proměnných jsou totiž v katalánštině (zdrojové kódy, verze 5.0) a ve francouzštině (`gTans`, verze 1.2).

Po roce jsem však nad těmi daty začal vyhrávat, dekoval jsem u `glpieces` rozklad na úroveň Bézierových křivek a to jsem věděl, že za týden nestihnou. Mnohem úspěšnější jsem byl s `gTans`, ale tam to též začalo lítat síny a kosíny. Když jsem si uvědomil, že mám na výběr, vysázení a výrobu týden, musel jsem jít na jistotu.

Postupně jsem vyzkoušel Tangram v přehrávači při programu Mathematica a můj oblíbený herní server `ihsan.biz`, ale nebylo to pro mě vhodné.

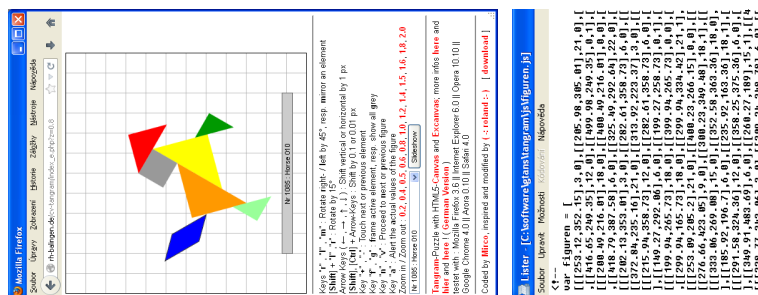
Prošel jsem celou škálu Windows programů typu Tangram-7, Temple Of Tangram, Tangram 3D a další. Nic, žádný posun kupředu.

Světlo naděje se rozbřesko při procházení serveru `tangrams.ca`. Tam jsem proklepl všechny odkazy. Zaujal mě projekt `Tanzzle`, ale tam byl formát dat zvláště realizovaný pomocí čísel bez mezer. Též zajímavě vypadal program na `games.ztor.com`, kde se dal zjistit výpis souřadnic objektů (poslední řádek pod `Help and Settings`), i přesto, že je to program ve Flashi, kde se dají takové věci snadno znepřístupnit. Napsal jsem autorovi o pomoc, bez odpovědi.

Téměř na den jsem se zastavil na jiném odkazu serveru `tangrams.ca`. Seznam mých možností se rozšířil. Tangram 1.0 měl svůj editor a data v textové podobě. Až jsem však data neparsoval zjistil jsem, že lichoběžník s rotací mění velikost! To je nádherná programátorská finta ve virtuálním světě, ale takový lichoběžník nelze vyrobit, neb to musí zůstat jako jeden kousek, nikoliv několik variant. V nouzi jsem samozřejmě mohl použít jednu polohu lichoběžníku a tangramy si vyfiltrovat (byl tam přepínač s hodnotou 0 a 1), ale to mi přišlo divné.

2. Výběr programu

Po předchozích zklamáních jsem si vysázel jeden tangram v `TikZ`u objekt po objektu, a říkal jsem si, že jestli bude nouze, vysázím si takto několik tangramů. To by na nedělní odpoledne bylo dostatečné, na výrobu tangramové knihy nikoliv. Knihy o tangramech lze na internetu dohledat minimálně dvě (R. C. Read a Sam Loyd). Zaměřil jsem tedy svou pozornost na editory tangramů a jejich výpisy.



Obrázek 1. Rozhraní vybraného programu a struktura dat tangramů.

Nejčitelnější formát byl při phdTangram2002, ale tangramů v té kolekci mnoho nebylo, pro děti by to stačilo, na větší projekt nikoliv. Hledal jsem dál.

Tu jsem si všiml zajímavé věci u <http://rh-balingen.de/cv-tangram/>. Po stisku klávesy »a« (u německé i anglické verze) se vypíše seznam souřadnic. Ale tentokrát to byl výpis jako proměnná. To se mi velmi zalíbilo a zkoumal jsem tento webový server dál. Ve spodní pravé části bylo mé oblíbené tlačítko »download« a už jsem mohl zkoumat HTML, PHP, JS či JSON. Vše tam bylo k ruce. To jsem si říkal, že bude stačit jen projít kratší zdrojové kódy (několikrát kratší vůči dvěma programům na Linuxu) a mělo by být pořešené. Zdrojové kódy byly pro změnu v němčině. Dekódoval jsem velmi málo. Napsal jsem autorovi o pomoc, ten se ještě ten den ozval a osvětlil mi některé obraty ze svého pohledu.

První klíčový problém byl posun objektu. Ten vždy probíhal od počátku souřadnic (levý horní roh kreslicí plochy, TikZ má roh vlevo dole), já vždy uvažoval od pozice nakresleného objektu. Také záleželo na pořadí posunu, rotace a zrcadlení.

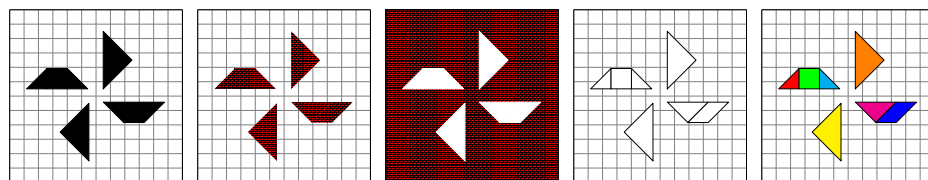
Druhý klíčový problém byl, že se posouvalo těžiště objektu, nikoliv střed boxu známého z typografie. Nu, odborná krátkozrakost, pojmenujme to takto. V \TeX u jsem se tedy vracel z těžiště do počátku souřadnic a až poté následoval posun.

Třetí klíčový problém bylo znaménko rotace a zrcadlení. Už nestačila oblíbená záměna znaménka u rotace (změna chodu hodinových ručiček). Také vstupovalo do hry měřítko a skoky rotací o 15 stupňů. To už jsem dokázal vyčíst z JavaScriptu. Drobný program v Pythonu to vše zvládl předpřipravit a rovnou generovat \TeX ový zdrojový kód do připravenéhoustru s původní holubicí.

3. Stavební prvky sazby

Abychom mohli tangramy vysázet, potřebujeme vyřešit řadu dílčích úkolů. Celý zdrojový kód je rozsáhlý v několika souborech a je ve vývoji, zaměříme se proto na přednášce na vysvětlení principů a klíčových momentů sazby. Na tomto místě si uvedeme seznam řešených oblastí bez zdrojových kódů:

- Definování a užití vlastní barvy.
- Přepínač stylů.
- Definování bodu.
- Transformace.
- Mřížka a orámování.
- Práce s výplněmi.
- Užití více stylů.



Obrázek 2. Tři formy zadání a dvě formy řešení jednoho tangramu: mlýn.

4. Příprava tiskových podkladů: lícování při oboustranném tisku

V této fázi jsem doufal, že mám tangramy pořešeny, ale to mě ještě čekalo seskládání stran před výrobou. Podíváme se na méně časté typografické problémy a zmíníme si řešení.

Náš úkol je rozmístit několik stran dokumentu na jednu tiskovou stranu, ale tak, aby původní strany oboustranně lícovaly. Potřebujeme to na výrobu kartiček.

4.1. Jednostranný tisk

Tento tisk je nejnámější, nepotřebujeme duplexní jednotku, ale na jednu sadu kartiček bychom spotřebovali dva papíry.

4.2. Oboustranný tisk, dlouhá hrana

Další možnost je zaměřit pozornost na oboustranný tisk, standardně užíváme dlouhé hrany, takto získáváme typ skript. Jenže začátky řádků zadní strany jsou na konci přední strany. Takže nám lícuje sazební zrcadlo dle vertikály, ale nikoliv texty. Pokud bychom v řádku měli stejné kartičky či rubová strana byla stejná (karty, pexeso), první by lícovala s poslední ap. Jenže když je každá kartička jiná, nepomůže nám to.

4.3. Oboustranný tisk, krátká hrana

Nepomůžeme si ani užitím krátké hrany. Dostáváme typ poznámkového bloku, chceme-li kalendáře. Lícuje nám zrcadlo dle horizontály, ale nikoliv texty. Horní část zadní strany se nám objeví ve spodní části přední strany. Lícovaly by přední a zadní strany první s poslední kartičkou ve sloupci ap.

4.4. Zrcadlo

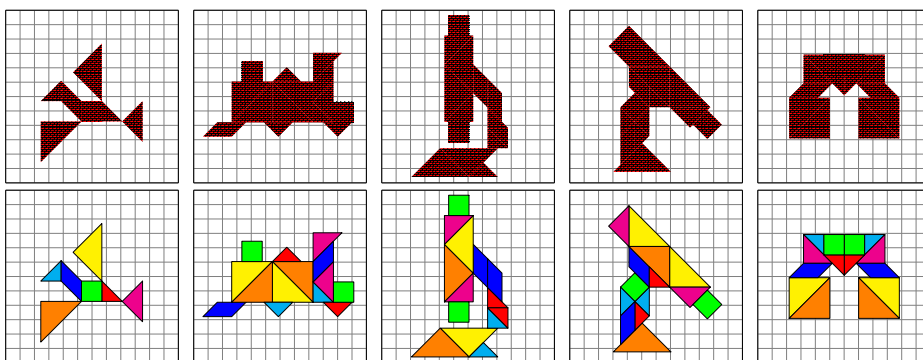
Vyzkoušeli jsme standardní tiskové možnosti, hloubější nastavování znamená si dělat poznámky či si tiskový profil ukládat. Lepší bude připravit si to na úrovni sazby. Nápad je užít zrcadlo, přes balíček `pdfpages` a parametr `reflect`. Jenže ani to ve finále nepomůže. U tangramů zrcadlo nevádí, ale neodpovídá to zadání. Ale horší je, že se zrcadlově vytisknou texty. Pokud to nevezmeme jako plánovaný vtip, je to nepoužitelné.

4.5. Nouzové řešení: Použít balíček `graphicx`

Jedna z možností je zrcadlově si otočit co bychom chtěli mít otočené. Je to příkaz `\scalebox` z balíčku `graphicx`, ale je potřeba si hlídat všechny objekty na straně. Není to zdaleka ideální řešení. Např. pomocí `\scalebox{-1}[1]{pokus}` je vysázen `awloq`.

4.6. Univerzální řešení: Zrcadlit zazrcadlené

Vhodným řešením je si v pomocném souboru zadní (obvykle sudé) strany kartiček (v našem případě řešení tangramů) zrcadlově otočit a pak tento soubor použít na další zrcadlení, ale již přes mříž $n \times m$. Je to jak v pravděpodobnosti dvojitá negace jevu dává původní jev. Jenže v mezidobí se zároveň přeskládají kartičky. Na konferenci si ukážeme vzorky vstupních podkladů i výsledků po nařezání.



Obrázek 3. Sada pěti tangramů: zadání a jejich řešení před přeskládáním na arch A4. První je holubice z dat z třídy lehcích tangramů o 7 dílcích. Dále následují dvojitě tangramy o 2·7 dílcích: lokomotiva, mikroskop a dalekohled jsou přebrány z knih. Poslední tangram nazvaný monstrum je z vlastní tangramové zahrádky za použití zmíněného editoru. Příprava zabrala několik minut.

5. Seskládání stránek

Nyní si ukážeme používanější metodu. Jedna kartička se rozmístí po celé straně, pak se nám problém lícování z předchozí kapitoly úplně vyhne. Jen oboustranný tisk nastavujeme po dlouhé hraně papíru.

5.1. Příprava cvičné stránky

Stačí nám minimální $\text{T}_{\text{E}}\text{X}$ ový dokument, ale ukažme si vysázení strany s velkou číslicí v jejím středu (`vzorek.tex`).

```
% Spuštění dvakrát. Formát: pdflatex nebo xelatex nebo lualatex...
\documentclass[a4paper]{article}
\usepackage{tikz} % Pomocný balíček nejen na absolutní pozicování.
\pagestyle{empty} % Vyčistí záhlaví a zápatí.
```

```

\begin{document} % Tělo dokumentu: začátek.
\begin{tikzpicture}[remember picture, overlay]
\node[scale=50] at (current page.center) {\bfseries\sffamily\thepage};
\end{tikzpicture} % Ukonči vykreslování.
\end{document} % Tělo dokumentu: konec.

```

5.2. Naskládání

Bratr nakladatel například používá dvoukrokovou metodu. V prvním kroku si jednu stranu (obvykle je to vizitka) namnoží na požadovaný počet, tím vznikne nové PDF o potřebném počtu stran (vzorek-16.tex). Např. pro 16 stran:

```

% Formát: libovolný LaTeXový.
\documentclass{article}
\usepackage{pdfpages,tikz}
\begin{document}
\foreach \x in {1,2,...,16} {\includepdf[fitpaper]{vzorek.pdf}}
\end{document}

```

To byl krok č. 1. Krok č. 2 je výstupní PDF přeskádat na tiskovou stranu.

```

% Formát: pdflatex, xelatex, lualatex
\documentclass[a4paper]{article}
\usepackage{pdfpages}
\begin{document}
\includepdf[pages={-},nup=4x4,frame]{vzorek-16.pdf}
\end{document}

```

Jedná se o poměrně rychlé a užívané řešení.

5.3. Sazba na mřížku

Pokud bychom v předchozí ukázce chtěli barevné orámování (`frame`) či dekoraci, už to není triviální řešení. Využijeme TikZ a sazbu na mřížku. Tohoto principu bylo použito na sazbu poustříků při OSSConf 2012.

Papír se rozdělí na potřebnou mřížku. Například získáme rozměr jednoho obdélníku. Ten po délce a výšce ještě rozpulme. Tím získáme střed. Nu a nyní můžeme přeskakovat po těchto délkách a získáme mříž. Na tu zavěsíme stránku.

```

% TeXujeme dvakrát, formát: pdflatex, xelatex, lualatex
\documentclass[a4paper]{article}
\usepackage{tikz}
\def\pocetx{4} \def\pocety{4} \def\soubor{vzorek.pdf} \def\strana{1}
\begin{document} % Získání potřebných rozměrů:
\pgfmathparse{\paperwidth/\pocetx} \let\dilek=\pgfmathresult
\pgfmathparse{\dilek/2} \let\dilekx=\pgfmathresult
\pgfmathparse{\paperheight/\pocety/2} \let\dileky=\pgfmathresult
\pgfmathparse{2*\pocetx-1} \let\azdox=\pgfmathresult
\pgfmathparse{2*\pocety-1} \let\azdoy=\pgfmathresult
\begin{tikzpicture}[remember picture, overlay, inner sep=0pt]

```

```

\foreach\x in {1,3,...,\azdox} { % Sazba na mřížku. Jedna osa.
\foreach\y in {1,3,...,\azdoy} { % Zde je pak druhá osa.
\node[xshift=\x*\dilekx, yshift=\y*\dileky, draw=red] at
(current page.south west){\includegraphics[width=\dilek pt]{vzorek.pdf}};
} % Konec cyklu přes \y.
} % Konec cyklu přes \x.
\end{tikzpicture}
\end{document}

```

5.4. Balíček pdfpages ještě jednou

Pokud by nám vadil pomocný soubor, můžeme vyzkoušet následující hrubé řešení.

```

% Formát: pdflatex, xelatex, lualatex
\documentclass[a4paper]{article}
\usepackage{pdfpages}
\begin{document}
\includepdfmerge[nup=2x2,frame]
{vzorek,1, vzorek,1, vzorek,1, vzorek,1}
\end{document}

```

Zde je ale problém, že vypisujeme tu samou informaci »vzorek,1« několikrát do povinného parametru. Nepřišel jsem na to, jak se dostat dovnitř a užít tam cyklus. To je důležité, když změním 2×2 na cokoliv jiného, pak musíme zasáhnout i do toho parametru. Při zapsání 5×5 už máme zbytečně mnoho práce.

Řešením by bylo si celý tento soubor vygenerovat, v Pythonu, Perlu či jiném programovacím jazyce. My si na závěr článku ukážeme užitečnost Lua_{TeX}u, užijeme Lua za běhu _{TeX}u. Příkaz si vygenerujeme jednoduše tehdy, když jej zrovna potřebujeme.

```

% Použijeme formát lualatex.
\def\x{3} \def\y{3} % Nastavme si mřížku.
\documentclass[a4paper]{article} % Formální začátek dokumentu.
\usepackage{pdfpages} % Zapněme si potřebný balíček.
\begin{document} % Zde začíná tělo dokumentu.
\begingroup % Globální příkaz s lokální změnou uvnitř.
\catcode'\~ =11 % Dočasně vypni aktivní znak kvůli nerovná se (~).
\gdef\vysazej{\directlua{ % Začátek příkazu a LuaTeXování.
c=\x*\y; t="vzorek,1"; % Nebo x=\x; y=\y; c=x*y;
r=string.char(92).. "includepdfmerge[nup=" .. tostring(\x)
.. "x" .. tostring(\y) .. ",frame]{ " % Začátek budoucího příkazu.
for pocet=1,c do % Opakuj název souboru a ukládané strany dle potřeby.
r=r..t % Přidej do generovaného příkazu opakovaný textový řetězec.
if pocet~c then r=r.." ," end % Přidej čárku. Nebo if not(pocet==c)...
end % Konec cyklu proměnné pocet.
r=r.."}" % Uzavři generovaný příkaz a jeho povinný parametr.
tex.print(r)} % Vypiš příkaz do TeXu. Konec příkazu \directlua.

```

```

} \endgroup % Uzavření příkazu \vysazej, uzavření skupiny.
\vysazej % Vygenerovaný příkaz se jedenkrát spustí.
\typeout{\vysazej} % Na terminál a do log souboru vypiš obsah příkazu.
\end{document}

```

Nyní máme k dispozici příkaz `\vysazej` a ten si můžeme volat dle potřeby. Zároveň vidíme užitečnost příkazu `\typeout`, kdy můžeme na příkaz nahlédnout na terminálu a v LOG souboru.

Poněvadž LuaTeX je v bouřlivém vývoji, je dobré vědět i o možnosti použít příkaz `dofile`, kdy si Lua kód umístíme mimo TeXový zdrojový kód. Odpadá starost s aktivní vlnkou, ale to jsou jen syntaktické krásy. Pro zájemce můžeme doporučit webové stránky projektu LuaTeX, poslední verzi manuálu a Lua.

Dalším krokem ve světě TeXu je formát ConTeXt, ale to je kapitola sama o sobě s masivním rozvojem a experimentálními záležitostmi, proto to není vhodné pro začínající TeXisty, ale je dobré o existenci formátu vědět. Někteří TeXisti by se mnou nesouhlasili, mohli by tvrdit, že ConTeXt může použít každý, pokud ano, studiu dokumentace a procházení ukázek se stejně nelze vyhnout.

6. Zkoumaná nastavení u výroby

Na první straně tohoto článku je vidět úsporná výrobní varianta. Ta však na kancelářském stroji není vhodná. Potřebujeme oboustranný tisk a nastavit rezervy kvůli posunu při nabrání papíru podavačem. Po několika úvahách jsme zvolili tisk na barevný papír jednostranně. Odpadá drahý barevný tisk, problém posunu linek a přípravy rezervy. Vznikl však nový problém seskládání jednoho dílku na papír.

Další zajímavá řešená situace je automatická rotace po delší straně tangramu (podobně jako tomu bylo u fraktálových poustrů), buď přímo přes TikZ nebo na úrovni parsování dat. Díky tomu půjde zrealizovat přechod z A4 na A5.

Přecházíme na tisk jednoho tangramu dvakrát kvůli možnosti soutěžení (rodič a dítě, mezi dětmi ap.). Jde v soutěži o to, kdo složí daný tangram rychleji.

7. Závěr

Sazba logických a deskových her je atraktivní téma, minimálně z pohledu typografie. Existuje řada balíčků k sazbě šachu, bridže, go, othello, sudoku, křížovek a dalších her. Některé balíčky jsem zmínil ve výuce. Nejnovější TeXové pokusy najdeme v balíčku `logicpuzzle` od Josefa Klebera. Je to opět jiný pohled na TikZ. Tangramy ani v tomto balíčku nenajdeme. Článek se pokusil tuto mezeru zaplnit.

Kontaktní adresy

Kristýnka Hůlková, Moravský Písek, Česká republika

Ing. Pavel Stríž, Ph.D., Nakl. Martin Stríž, U Škol 940, 685 01, Bučovice, Česká republika,
E-mailová adresa: pavel@striz.cz

PŘEDSTAVENÍ PROGRAMU MUESCORE

TEREZKA HŮLKOVÁ (CZ) A PAVEL STRŽÍŽ (CZ)

Abstrakt. Autoři článku, neteř Tereška a strýc Pavel, představí svobodný program MuseScore, <http://musescore.org/>. Je to program vhodný na vysázení notového zápisu a přehrání vytvářené skladby. Tohoto zajímavého programu jsme si všimli minulý rok (2012), kdy jsme, po pravdě řečeno trochu nemotorně, sázeli státní hymnu Slovenské republiky v MusiX_TE_Xu a LilyPondu, a to ve chvíli, kdy jsme si potřebovali hymnu přehrát. Sázeční programy tuto možnost standardně nenabízí. V článku je představena běžná práce v tomto programu. Na konferenci bude navíc poukázáno na několik konvertorů hudebních formátů, jako TAudioConverter, Audacity a SoundConverter. Vedle toho představíme i přílehlý archiv hudebnin, <http://musescore.com/>.

Klíčová slova. Hudba, hudebniny, notový zápis, grafický zápis hudby, tabulatura.

AN INTRODUCTION TO MUESCORE

Abstract. The authors, a niece Theresa and her uncle Paul, introduce an open source software named MuseScore, <http://musescore.org/en>, a musical notation program which is a complete scorewriter and notation editor. We have noticed this excellent program last year (2012) while we were typesetting the national anthem of Slovakia in MusiX_TE_X and LilyPond, esp. when we needed to listen to what we had just typeset. Regular typesetting engines couldn't provide such a feature. In the article, the authors show several features of this software. During the OSSConf 2013 conference, we will also introduce some converters, such as TAudioConverter, Audacity, and, SoundConverter. Along the way, a musical archive, <http://musescore.com/>, is presented.

Key words and phrases. MuseScore, Musical Notation, Scorewriter, Sheet Music.

1. Úvod

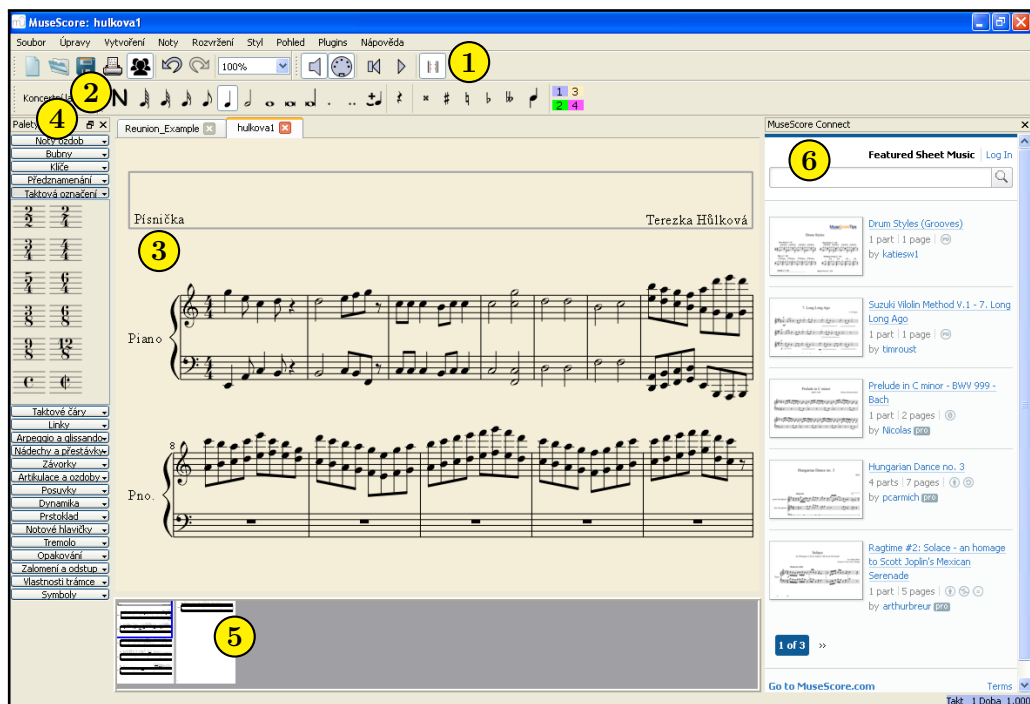
Setkali jsme se se sazbou hudebnin pomocí MusiX_TE_Xu i LilyPondu, ale skladba se nedala přímo přehrát. Upřímně řečeno, také to nejsou vhodné nástroje pro nezletilé neprogramátory.

Když si chceme skladbu editovat a zároveň přehrát, existují alternativní nástroje, zmiňme namátkou z repozitářů: Frescobaldi, Denemo, jEdit, Impro-Visor, Rosegarden, TuxGuitar, Aria Maestosa, Canorus, NoteEdit či NtEd. Za poslední roky se jich urodilo. Program MuseScore [čtème mjuzísko:r] zaujal hlavně tím, že u programu je dostupný archiv písniček a lze pracovat s více hudebními nástroji.

Na druhé straně jsme se setkali s komerčním programem Reason, ale to je produkt už pro vyspělé, zkušené a profesionální hudebníky, řekli bychom.

2. Práce v MuseScore

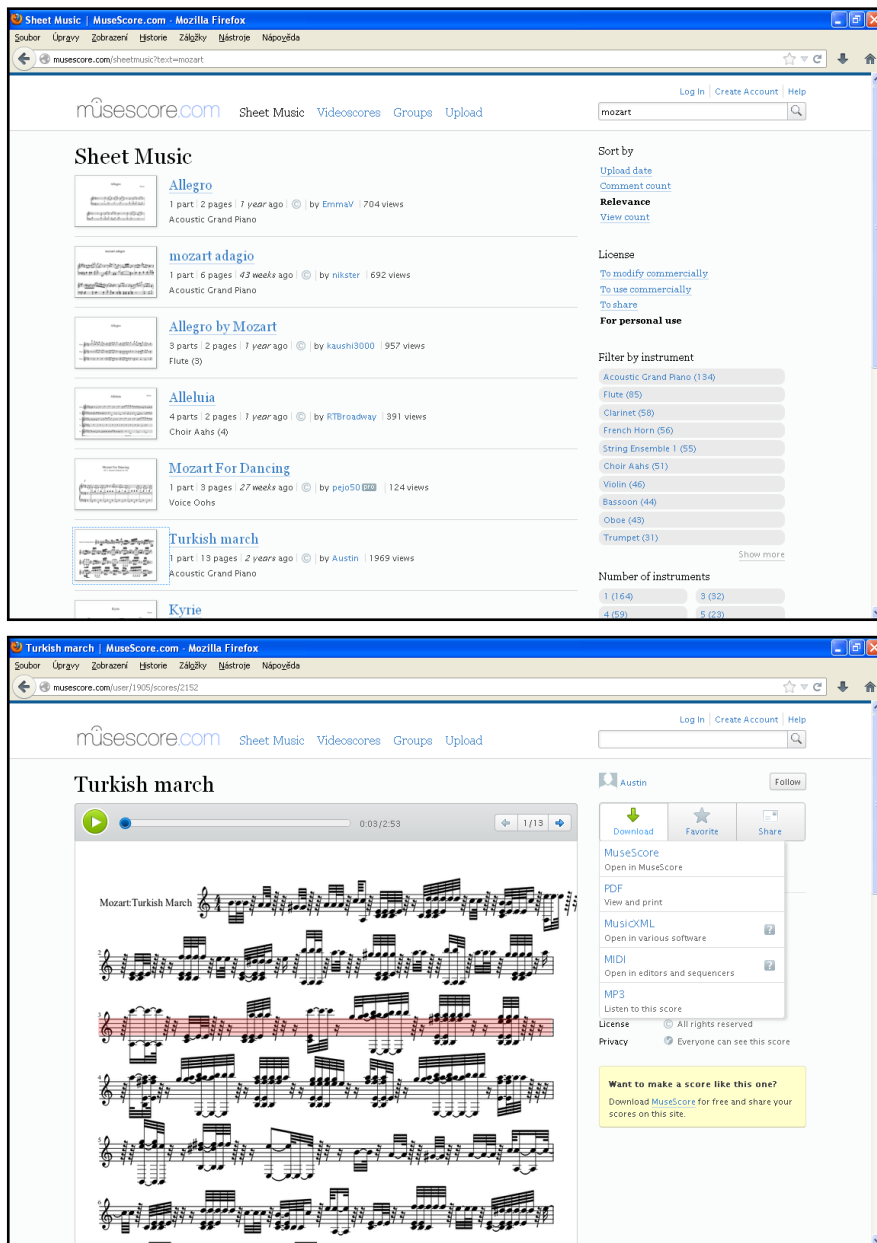
Po instalaci, která je pod Linuxem, Mac OS X (v repozitáři hledejme *muse-score*) i Microsoft Windows, standardní, uvidíme po spuštění následující:



- ① Z menu ovládáme funkce programu, za pozornost stojí spuštění písničky v libovolné části přípravy skladby a libovolné části písničky.
- ② Editaci a volbu not lze provádět z klávesnice nebo za pomoci ikoněk.
- ③ Zde se provádí editace notového zápisu. Jedná se o WYSIWYG.
- ④ Většinu hudebních symbolů lze dohledat v panelech v levé části MuseScore. Na snímku můžeme vidět rozkliknutý blok taktových označení.
- ⑤ Ve spodní části můžeme vidět náhled na tiskovou stránku.
- ⑥ Za pozornost jistě stojí i možnost stažení písniček z archivu s možností následné editace, po registraci můžeme do archivu uložit vlastní skladby.

3. Typografie

MuseScore je zajímavý produkt i z pohledu typografie. Výstupy můžeme získat v PDF, PS, PNG a SVG, ale též jako písničky MID, WAV, FLAC a OGG. Nativní formát je MusicXML či její komprimovaná verze MSCZ a MSCX. Standard MusicXML je



Obrázek 1. Ukázka z vyhledávání skladby na musescore.org. Skladatel nebo název skladby se zadává v horním pravém poli. Z výsledků v levé části webové stránky si naklikneme »Turkish march« a dostáváme se do spodního okna. Zde vidíme skladbu a červeně podbarvená je právě hraná část. V pravé části si můžeme stáhnout editovatelný soubor MSCZ, PDF, MXL, MID a MP3.

CHEMICKÁ SADZBA S BALÍČKOM CHEMFIG

ALEŠ KOZUBÍK (SK)

Abstrakt. Chémia, rovnako ako matematika, so sebou prináša zvýšené nároky na počítačovú sadzbu. Na rozdiel od matematiky, však \LaTeX nedisponuje špecializovaným prostredím pre sadzbu chemických vzorcov a rovníc. V tomto článku autor predstavuje balíček `ChemFig`, ktorý pomáha riešiť tento problém.

Kľúčové slová. chémia, sadzba, \LaTeX , balíček `ChemFig`.

THE CHEMISTRY TYPESET WITH THE CHEMFIG PACKAGE

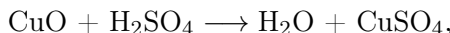
Abstract. Chemistry as well as mathematics, bring with it increased requests on the computer typesetting. Unlike mathematics but \LaTeX does not have a specialized environment for typesetting chemical formulas and equations. In this paper, the author introduces the `ChemFig` package that helps to solve this problem.

Key words and phrases. chemistry, typesetting, \LaTeX , `ChemFig` package.

Úvod

V predchádzajúcich ročníkoch sme sa oboznámili so základmi práce s typografickým systémom $\text{\TeX}/\text{\LaTeX}$. Naučili sme sa základy hladkej sadzby [1, 2], vysporiadali sme sa so spracovaním obrázkov [3, 5]. Taktiež sme si ukázali sadzbu matematických vzorcov a rovníc v rozsahu danom publikáciami [4, 6, 8]. V knihe [7] sa potom môžeme oboznámiť, ako poznatky využiť pre dosiahnutie skutočne kvalitnej úpravy dokumentu. Ďalšou oblasťou, ktorá si vyžaduje osobitnú pozornosť pri sadzbe a jej spracovanie možno označiť za neelementárne je sadzba chemického obsahu, a to najmä z oblasti chémie organickej.

Ak pri sadzbe anorganickej chémie si väčšinou vystačíme s matematickým prostredím napr. môžeme vysádzať rovnicu



či vytvoriť symbol $^{12}_6\text{C}$, tak v chémii organickej už toľko šťastia nemáme. Tu prichádzajú požiadavky na zobrazovanie molekúl, štruktúrnych vzorcov, cyklických zlúčenín a pod. Preto si predstavíme balíček `ChemFig`, ktorého autorom je Christian Tellechea a manuál ku nemu [9] je voľne dostupný na internete. Jedná sa o balíček, ktorý je založený na balíčku `TikZ` a oproti iným nástrojom pre sadzbu chémie, ako sú `ppctex` alebo `mhchem` sa vyznačuje jednoduchou syntaxe.

1. Predstavenie balíčka ChemFig

Balíček `ChemFig` zaradíme do TEX -ového dokumentu štandardným spôsobom, teda príkazom `\usepackage{chemfig}` ak používame $\text{L}\text{A}\text{T}\text{E}\text{X}$, pre $\text{C}\text{o}\text{n}\text{T}\text{E}\text{X}\text{t}$ použijeme `\usemodule{chemfig}` atď. `ChemFig` si vyžaduje balíček `TikZ` a pokiaľ nebol načítaný pred tým, `ChemFig` si ho sám automaticky načíta.

Základný príkaz balíčka `ChemFig` je `\chemfig{<kod>}`, kde argument `kod` predstavuje postupnosť znakov, ktorá charakterizuje molekulu, ktorej zobrazenie požadujeme. Molekula je následne vykreslená v prostredí `tikzpicture`. Pre zobrazenie výsledku môžeme použiť buď priamo kompiláciu cez `pdflatex` alebo staršiu cestu `tex` \rightarrow `dvi` \rightarrow `ps` \rightarrow `pdf`.

Základná syntax príkazu `\chemfig` je:

```
\chemfig{<atom1><vazba>[<uhol>,<koef>,<n1>,<n2>,<tikz kod>]<atom2>}
```

kde:

- `<uhol>` je uhol väzby medzi atómami,
- `<koef>` je koeficient, ktorý násobí základnú dĺžku väzby,
- `<n1>` a `<n2>` sú počty atómov vstupujúcich a vystupujúcich z väzby,
- `<tikz kod>` príkazy balíčka `Tikz`, napr. nastavenie farby a pod.

Široké možnosti príkazu `\chemfig` si krátko predstavíme v ďalších odstavcoch.

Kód	Výsledok
<code>\chemfig{X-Y}</code>	X — Y
<code>\chemfig{X=Y}</code>	X = Y
<code>\chemfig{X~Y}</code>	X ≡ Y
<code>\chemfig{X>Y}</code>	X ► Y
<code>\chemfig{X<Y}</code>	X ◄ Y
<code>\chemfig{X>:Y}</code>	X ... Y
<code>\chemfig{X<:Y}</code>	X ... Y
<code>\chemfig{X> Y}</code>	X ▷ Y
<code>\chemfig{X< Y}</code>	X ◁ Y

Tabuľka 1. Prehľad zobrazení chemických väzieb

2. Zobrazenie chemickej väzby – základ sadzby chémie

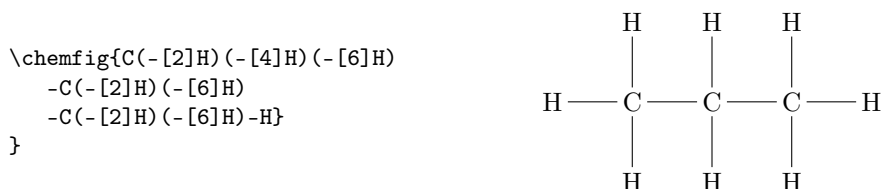
Na ilustráciu si ukážme jednoduchú molekulu propánu $\text{CH}_3\text{—CH}_2\text{—CH}_3$, ktorú sáďeme príkazom `\chemfig{CH_3-CH_2-CH_3}`. Samozrejme, tu uvedené

jednoduché väzby nie sú ani zďaleka všetkými typmi väzby a jej zobrazenia, s ktorými sa pri sadzbe chémie môžeme stretnúť. Podrobnejší zoznam rôznych typov väzby a ich zobrazenia uvádza tabuľka 1.

Rozostup medzi čiarami slúžiacimi k znázorneniu viacnásobnej väzby je možné upraviť príkazom `\setdoublesep{<veľkosť>}`. Jej štandardná veľkosť je prednastavená na hodnotu 2pt.

Samotné zobrazenia väzby sa však, najmä v organickej chémii, často vyskytujú v rámci kompletných štruktúrnych vzorcov, ktoré znázorňujú detailnú štruktúru zlúčeniny. Neraz sa štruktúra molekuly znázorňuje aj s príslušnými uhlami medzi väzbami jednotlivých atómov.

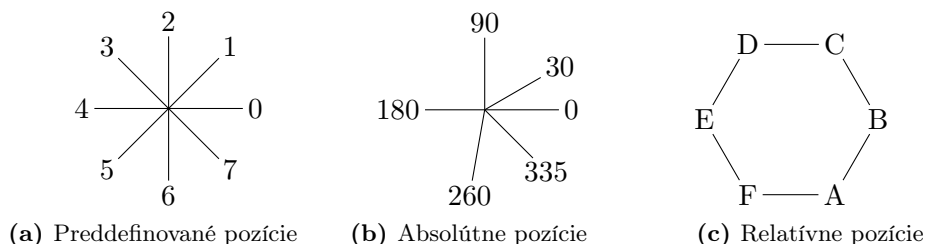
Tak napríklad uvedenú jednoduchú molekulu propánu $\text{CH}_3\text{---CH}_2\text{---CH}_3$ môžeme vysádzať aj v podobe štruktúrneho vzorca nasledovným spôsobom:



Významným voliteľným argumentom každej väzby je `<uhol>`. Jeho implicitnou hodnotou je 0° . Pre jeho zmenu môžeme použiť buď niektorú z ôsmich preddefinovaných pozícií 0–8, tak ako to zobrazuje obrázok 1a.

Absolútnu hodnotu uhla zadávame v tvare `[:<číslo>]`, kde veličina `<číslo>` je absolútna veľkosť uhla v stupňoch, meraná proti smeru hodinových ručičiek od pozície nula. Jej hodnota sa môže pohybovať v intervale $(0, 360)$. Niekoľko pozícií s absolútnym vyjadrením uhla je na obrázku 1b.

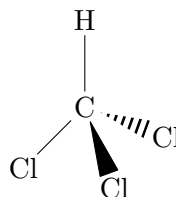
Poslednou možnosťou je relatívna hodnota, vzhľadom na aktuálnu pozíciu poslednej zobrazenej väzby. Zadáva sa v tvare `[::<číslo>]`. Táto hodnota môže byť kladná aj záporná a môže obsahovať aj desatinné čísla. Obrázok 1c zobrazuje šesť pozícií a väzieb, ktoré sa posúvajú vždy o rovnaký uhol 60° .



Obr. 1. Ukážky rôznych možností zadania uhla väzby

Ako príklad, aj so zdrojovým kódom, si zobrazme štruktúrny vzorec molekuly chloroformu:

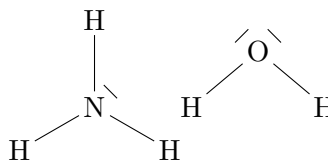
```
\chemfig{C(-[5]Cl)(-[2]H)(<[: -70]Cl)(<[: -20]Cl)}
```



Podrobnejšou verziou štruktúrneho vzorca je tzv. elektrónový vzorec, ktorý okrem samotných väzieb vyznačuje aj ostatné elektróny vo valenčnej sfére. Konfigurácia elektrónov sa zobrazí príkazom `\lewis{<pozícia>,<stav>,<atom>}`.¹ Pritom <pozícia> je určená hodnotou 0–7, rovnako ako pri preddefinovanej pozícii väzby, <stav> určuje či ide o elektrónový pár, samostatný elektrón alebo elektrónovú dieru. Implicitnou hodnotou je dvojica elektrónov.

Použitie si ilustrujeme na molekulách amoniaku a vody. U vody si všimnime kumuláciu dvoch pozícií elektrónových párov na pozíciách 1 a 3 v molekule kyslíka.

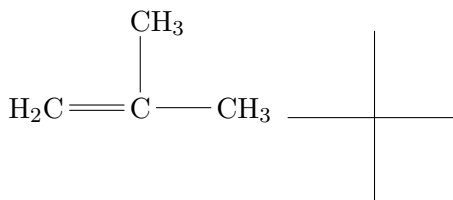
```
\chemfig{\lewis{1,N}(-[2]H)(-[:210]H)(-[: -30]H)}
\chemfig{[:40]H-\lewis{13,O}(-[: -80]H)}
```



3. Rozvetvené molekuly

Pre zobrazenie štruktúry zložitejších molekúl potrebujeme mať k dispozícii možnosť vetvenia väzieb v štruktúrnom vzorci. Kód vetvenia jednoducho zapíšeme do zátvoriek v rámci špecifikácie väzieb v parametri príkazu `\chemfig`. Ako príklad z kategórie jednoduchších si ukážme kumulovaný štruktúrny vzorec izobutylénu, čiže 2-metylpropénu. Pre priblíženie polohy základného riadku pri vetvení je zobrazený samostatný kríž väzieb bez atómov.

```
\chemfig{H_2C=C(-[2]CH_3)-CH_3}
\hspace{.2cm}
\chemfig{-(-[2])(-[6]) -}
```

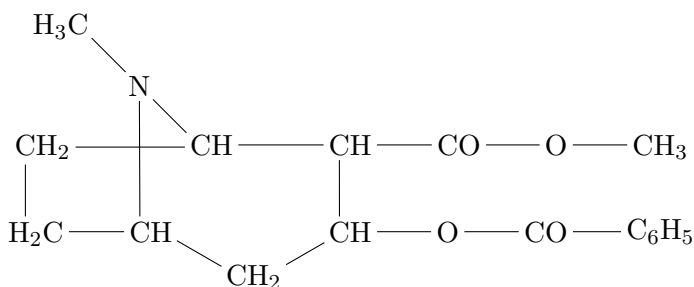


¹Odvođené z anglického pomenovania týchto vzorcov ako Lewis diagram.

Pri zložitejších molekulách sa vyskytuje potreba spojiť aj vzdialené atómy. K tomu slúži operátor „háčika“² `?[<meno><väzba><tikz>]`, kde `<meno>` umožňuje pomenovať háčiky, ak je ich potrebné vytvoriť viac, a pomenovaním tak špecifikovať, ktoré atómy majú byť spojené, `<väzba>` udáva typ väzby a `<tikz>` je nejaký dopĺňajúci kód TikZ-u, ktorým možno napr. špecifikovať farbu. Ako ukážku si zobrazme pomerne zložitú molekulu kokaínu. Najskôr zdrojový kód:

```
\chemfig{H_2C(-C?[a]H-[: -30]CH_2-[: 30]C?[b]H-O-CO-C_6H_5)-[2]
  CH_2-[ ,1.7]CH(-[3]N?[a]-[3]H_3C)(-[ ,1.35]C?[b]H-CO-OCH_3)}
```

a teraz výsledok, t. j. vzorec molekuly kokaínu:

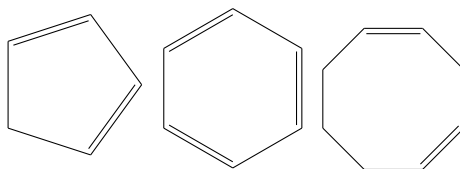


4. Cyklické zlúčeniny

Balíček `ChemFig` disponuje nástrojmi, pre jednoduché kreslenie n -uholníkov. Základná syntax tohto príkazu je `\chemfig{*n(<molekula>)}`, kde `<molekula>` predstavuje kód zloženia príslušnej molekuly v duchu vyššie uvedených pravidiel. Pre ilustráciu si ukážme niekoľko n -uholníkov:

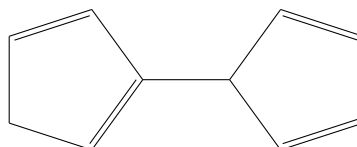
```
\chemfig{*5(-----)}
\chemfig{*6(=-----)}
\chemfig{*8(-----)}

```



Rovnako ako u acyklických zlúčenín, aj u vzorcov cyklických zlúčenín môžeme vykresliť vetvenie molekuly. Situáciu si ilustrujeme na príklade:

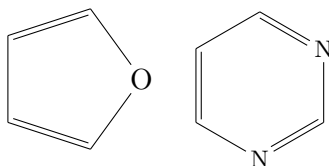
```
\chemfig{*5(=
  (-[0]*5(-----))---)}
```



²V anglickom originále hook.

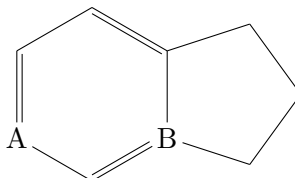
Do jednotlivých vrcholov je možné vložiť ľubovoľný text, čo je významné najmä pre potreby vykreslenia heterocyklických zlúčenín. Opäť ilustrujme na príklade furánu a pyrimidínu:

```
\chemfig{*5(=O--)}
\hspace{0.5cm}
\chemfig{*6(-N=-N=-)}
```



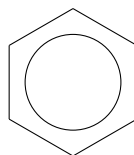
Len mierna je odlišnosť spájania viacerých cyklov v rámci jednej molekuly. Je potrebné špecifikovať vrchol, v ktorom sa začína pripojený cyklus. Nasledujúci príklad ilustruje pripojenie päťčlánkoveho cyklu na tretom atóme šesťčlánkoveho cyklu:

```
\chemfig{A*6(=B*5(-----)===)}
```



Pre úplnosť symbolov používaných vo vzorcoch cyklických zlúčenín si ešte ukážme syntax pre vykreslenie benzénového jadra:

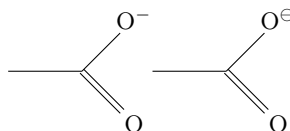
```
\chemfig{**6(-----)}
```



5. Zobrazovanie iónov

Príkaz `\chemfig` využíva matematické prostredie $\text{T}_{\text{E}}\text{X}$ -u, čo umožňuje aj jednoduchú sadzbu iónov. Treba však mať na pamäti, že pre zobrazenie záporného náboja treba znamienko mínus vždy uzavrieť do skupinových zátvoriek, aby nedošlo k zámene so symbolom jednoduchej väzby. Pokiaľ ste puritánmi, je možné znamienka plus a mínus vysádzať uzavreté do krúžkov pomocou príkazov `\ominus` a `\oplus`. Rozdiel ilustruje nasledujúca ukážka:

```
\footnotesize \chemfig{(-[1]O^{-})=[7]O}
\hspace{1em}
\footnotesize \chemfig{(-[1]O^{\ominus})=[7]O}
```



6. Rovnice chemických reakcií

Pre účely zápisu chemických reakcií v podobe rovníc prináša balíček ChemFig dva nové príkazy. Sú to `\chemsign` a `\chemrel`. Príkaz `\chemsign` má jeden voliteľný argument, ktorý udáva veľkosť nezlomiteľnej medzery, ktorou je na oboch stranách oddelené znamienko, zadávané ako povinný parameter príkazu. Implicitná hodnota je medzera vo veľkosti 0.5em. Príkaz `\chemrel` má dva nepovinné parametre, kde prvý umožňuje umiestniť popis nad znamienko relácie a druhý parameter pod znamienko relácie. Povinným parametrom je potom symbol relácie, pričom sa zadáva v syntaxi bežnej pre TikZ. Možné typy šípok sú ilustrované v tabuľke 2.

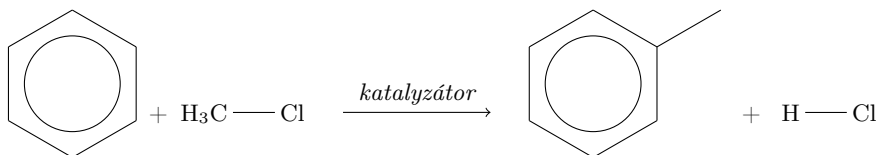
Kód	Výsledok
<code>A\chemrel{->}B</code>	A \longrightarrow B
<code>A\chemrel{<-}B</code>	A \longleftarrow B
<code>A\chemrel{<->}B</code>	A \longleftrightarrow B
<code>A\chemrel{<>}B</code>	A \rightleftharpoons B
<code>A\chemrel{->,red,thick}B</code>	A $\xrightarrow{\text{red,thick}}$ B

Tabuľka 2. Ukážky typov šípok v chemickej rovnici

Na ilustráciu ich použitia si ukážeme dve chemické reakcie. Najskôr zdrojový kód:

```
\setchemrel{0pt}{1.2em}{6em}
\chemfig{*6(-----)}\chemsign+\chemfig{H_3C-Cl}
\chemrel[\itshape\footnotesize katalyzátor]{->}
\chemfig{*6(---(-)---)} \chemsign+ \chemfig{H-Cl}
```

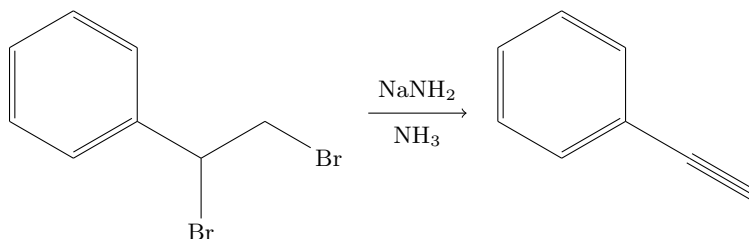
a výsledný zápis katalytickej reakcie:



Ako druhú ukážku si uvedieme rovnicu syntézy fenylacetylénu. najskôr opäť zdrojový kód rovnice:

```
\chemfig{*6(==*6(-(-Br)-(-Br))--=)}
\chemrel[\chemfig{NaNH_2}][\chemfig{NH_3}]{->}
\chemfig{*6(=-(~)-==)}
```

a po ňom výsledný zápis reakcie:



7. Záver

V článku sme si predstavili základné prvky balíčka **ChemFig**, ktoré umožňujú zvládnuť sadzbu dokumentov s obsahom materiálu z oblasti chémie. Podané základné informácie akiste nie sú vyčerpávajúce a chemický profesionál by iste hľadal mnohé iné prvky. Postačí však pre základnú orientáciu a ako základ k samostatnému rozširujúcemu štúdiu manuálu [9].

Literatúra

- [1] BLAŠKO, R.: *L^AT_EX nie je farba na maľovanie*. Zborník príspevkov medzinárodnej konferencie OSSConf 2010, 1.–4. júla 2010, Žilina, str. 43–52. ISBN 978-80-970457-0-8.
- [2] BLAŠKO, R.: *L^AT_EX nie je farba na maľovanie, ale na písanie*. Zborník príspevkov medzinárodnej konferencie OSSConf 2011, 6.–9. júla 2011, Žilina, str. 223–236. ISBN 978-80-970457-1-5.
- [3] BLAŠKO, R.: *Nebojme sa obrázkov v L^AT_EX-u*. Zborník príspevkov medzinárodnej konferencie OSSConf 2011, 2.–4. júla 2012, Žilina, str. 79–85. ISBN 978-80-970457-2-2.
- [4] KOPKA, H. – DALY, P. W.: *L^AT_EX – podrobný príručce*. Computer Press, Brno, 2004. ISBN 80-722-6973-9.
- [5] KOZUBÍK, A.: *Naučím vás kresliť alebo predstavenie balíčka TikZ*. Zborník príspevkov medzinárodnej konferencie OSSConf 2011, 2.–4. júla 2012, Žilina, str. 91–96. ISBN 978-80-970457-2-2.
- [6] RYBIČKA, J.: *L^AT_EX pro začátečníky*. KONVOJ, Brno, 2003. ISBN 80-7302-049-1.
- [7] RYBIČKA, J. – ČAČKOVÁ, P. – PŘICHYSTAL, J.: *Průvodce tvorbou dokumentů*. NAKLADATELSTVÍ MARTIN STRÍŽ, Bučovice, 2011. ISBN 80-7302-049-1.
- [8] STRÍŽ, P.: *Sazba v T_EXu a kresba v METAPOSTu*. NAKLADATELSTVÍ MARTIN STRÍŽ, Bučovice, 2011. ISBN 978-80-87106-51-8.
- [9] TELLESCHEA, CH.: *ChemFig: A T_EX package for drawing molecules*. http://www.ctan.org/pkg/chemfig/chemfig_doc_en.pdf.

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

ZKUŠENOSTI S PLUGINEM WP-QUICK \LaTeX

TIBOR MENYHÉRT (SK) A PAVEL STRÍŽ (CZ)

Abstrakt. Košický bridžový klub přešel na nový systém pro správu obsahu WordPress a řešil se staronový problém hezkého vysázení symbolů barev (piky, srdce, kára, trefy). Zkusili jsme několik standardních možností a ty nás neuspokojily. Po několika experimentech došlo až na \TeX ový svět a jeho možnosti. Vyzkoušeli jsme několik dostupných pluginů a WP-Quick \LaTeX vyšel z testů nejlépe. Poukážeme na principy práce s ním (instalaci, nastavení a používání v tomto webovém prostředí), vyřešíme sazbu vlastních symbolů barev přes $TikZ$, bridžové licitace, sehrávky, ale i sazbu diagramů s možností přepínání pohledů – na jeden (ohodnocení listu, problémy licitace, prvního a pozdních výnosů), dva (problémy hlavního hráče, angl. single dummy problems, a obránců) i všechny listy hráčů (angl. double dummy problems).

Klíčová slova. Karetní hra bridž, WordPress, pluginy, \TeX , WP-Quick \LaTeX , $TikZ$.

EXPERIENCE WITH THE WP-QUICK \LaTeX PLUGIN

Abstract. The bridge club of Košice city has changed their content management system to the WordPress web software and an unfamous problem of typesetting pretty suit symbols (spades, hearts, diamonds, clubs) arised again. After several common attempms which didn't satisfied us, the authors touched the \TeX world and tried several available WordPress plugins. The WP-Quick \LaTeX plugin came as a winner. We show how to install, setup and use this plugin within the WordPress environment. Along the way, we point out to a typesetting of the suit symbols as well as the bidding in bridge, play and hand diagrams with a player's view switcher to see one (hand evaluation, bidding, opening and later leading problems), two (declarer known as a single dummy and defenders problems) and all hands (known as a double dummy problems).

Key words and phrases. Bridge, WordPress, Plugins, \TeX , WP-Quick \LaTeX , $TikZ$.

Úvod

Pokud hrajete karty, potrpíte si na vysázení znaků barev, podobně jako šachisti šachových figur. Po nějaké době se jakékoliv znaky okoukají a tu a tam je vhodná změna. Platí to i pro webové prostředí, kde se běžně bridžové články publikují. Podívejme se na základní možnosti, které autor takového článku má.

1. Možnosti přes HTML

Nejrychlejší možnost je přebrat existující řešení, např. znaky používané v Bridžové Revue při Českém bridžovém svazu, <http://revue.bridgecz.cz/>:

```




```

Výsledek je: ♠♥♦♣

Řešení je použitelné, ale není to vektor, při tisku článku by nám to vadilo. Další možnost je vzít v úvahu běžné písmo, pokud znaky barev obsahuje.

```
&#9824;&#9827;&#9829;&#9830;
```

Výsledek je vektorový (v článku jako rastr): ♠♥♦♣

Pokud si s tím trochu pohrajeme, můžeme nastavit barvu a použít kaskádové styly a pracovat se znaky jako s UTF-8 znaky, nikoliv HTML entitami. Více o tom na této webové stránce.

```
<font color="blue">&spades;</font>
<font color="red">&hearts;</font>
<font color="orange">&diamonds;</font>
<font color="green">&clubs;</font>
```

Vektorový výsledek (zde opět jako rastr): ♠♥♦♣

To je přijatelné řešení, pokud se nám však líbí T_EXové symboly nebo máme znaky vlastní, znamená to učinit další kroky.

2. Možnosti přes T_EX

V T_EXu je standard získat symboly přes matematický režim: ♠, ♥, ♦, ♣.

```
 $\spadesuit, \heartsuit, \diamondsuit, \clubsuit$ 
```

Jak však vidíme, srdce a kára jsou jen obrysy a změna barvy nezasáhne výplň: ♠, ♥, ♦, ♣. Před tímto pokusem si v preambuli zapínáme balíček xcolor.

```
% Do preamble: \usepackage{xcolor}
 $\spadesuit, {\color{red}\heartsuit}, \diamondsuit, \clubsuit$ 
```

Jaké jsou alternativy při užití T_EX Live? Můžeme se podívat na důležité symboly přes T_EXovou dokumentaci:

```
 $\text{doc symbols}$ 
```

Jinou kresbu najdeme přes balíčky txfonts a pxfonts: ♠, ♥, ♦, ♣ – ♠, ♥, ♦, ♣.

```
 $\spadesuit, \heartsuit, \diamondsuit, \clubsuit$  --
 $\varspadesuit, \varheartsuit, \vardiamondsuit, \varclubsuit$ 
```

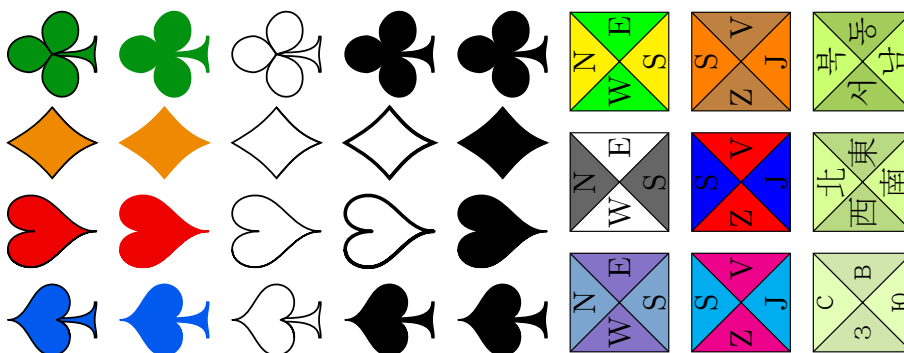
Užitím běžné kresby na piky a trefy a variantní kresby na srdce a kára získáme dvou- či čtyřbarevnou kombinaci: ♠, ♥, ♦, ♣ – ♠, ♥, ♦, ♣. Podobné barvy se používají v programu Tenace či na BBO fórech. Karetní svět ještě na čtyři barvy u karet (lepší viditelnost) nepřešel, zatím se tu a tam drží čtyřbarevné licitační kartičky.

Zdrojový kód nezveřejňujeme, necháváme za domácí úkol drahému čtenáři. Vidíme však, že kresba znaků jsou jen výplně bez obrysů. To je náramná škoda!

Jistou variantní kresbu ještě získáme přes balíček `MnSymbol` (str. 66) či `arev` (str. 67), ale to nejsou uspokojivá řešení. Zbrusu nový balíček `tikzsymbols` (duben 2013) karetní znaky neobsahuje. Dále bychom museli zkoumat už přímo písma.

3. Vlastní kresba přes TikZ

Uvažovali jsme tak, že by stálo za uvážení vytvořit si vlastní symboly. Šli jsme na to přes `TikZ` a poněvadž je zdrojový kód rozsáhlý, nechť laskavý čtenář nahlédne a stáhne si jej zde. Používáme pět variant a doplňkovou kresbu karetního stolu. Lze si nastavit barvy, sílu linky obrysu, včetně vypnutí obrysu, a znaky lze škálovat.



4. Konverze pro webové stránky

Poprvé byly znaky použity na karetní koncovky publikované v časopise *Bridžová Revue* a jako kapitola o bridži v knize o logických hrách (ISBN 978-80-87106-52-5). Zaujalo to kolegy v Košicích, kteří právě přecházeli na nový systém správy obsahu WordPress, a napadlo je, jestli by znaky nešlo u nich začlenit.

Pokud si přejeme dostat $\text{T}_{\text{E}}\text{X}$ ové symboly na webovou stránku, řešení je napínavější. Například přes `METAPOST` můžeme získat SVG přímo, podobně konverzí PDF. Tato konverze do SVG občas zazlobí, natahují a přetahují se křivky, síla linky není jednotná ap. Mnohem stabilnější je převést je do rastrového formátu.

Ukážeme si situaci na jednoduchém znaku, podobně se řeší například znaky pro GeoGebra (`GGB icons`). V $\text{T}_{\text{E}}\text{X}$ u máme ikonky přímo k dispozici, pro web se převádí symbol po symbolu. Může to být ale cokoliv: logo, značka, mapa ap.

Pokud máme obrázek v $\text{T}_{\text{E}}\text{X}$ u můžeme převést (`T_{E}X Live` a `GraphicsMagick`):

```
$ pdflatex muj.tex
```

```
$ pdfcrop --hires --margins 0.5 muj.pdf
```

```
$ gm convert -transparent white -resize 40% muj-crop.pdf muj.png
```

A přes kaskádový styl v hlavičce HTML můžeme vertikálně posouvat:

```
<style type="text/css">vobr{position:relative;top:-0.8ex;}</style>
```

V těle HTML dokumentu pak užijeme:

```
<vobr></vobr>
```

To však nové tagy u některých prohlížečů nezobrazuje optimálně, také nevýhodou je, že je to vertikální nastavení od oka.

Navíc, přijdeme o informaci o zdrojových kódech a nelze do obrázku zasáhnout bez náročnějšího zpětného \TeX ování. Další problém je držet účaří linky, neb máme v ruce rastrový obrázek a o tuto informaci jsme přišli. To lze u jednoho dvou obrázků řešit ručně, ale nikoliv u většího počtu či u častějších zásahů. Získat rozměry hloubky v \TeX u lze (\setbox a \dp přes boxy), ale není to jednoduché \TeX ování s převodem například do pixelů.

5. WordPress a WP-Quick \LaTeX

Nebudeme již čtenáře napínat. WordPress je komplexní webové prostředí založené na PHP rozšiřitelné o pluginy (opět známé PHP). Existují dokonce pluginy pro \TeX ovou podporu. Například na Wikipedii či MathWorld se používá texvc , WikiTeX a podobné nadstavby a velmi důležitou vlastností při vkládání obrázků je, že v atributu alt zůstává \TeX ový zdrojový kód, ten si stačí vykopírovat!

\TeX ových pluginů pro WordPress je několik, většinou staršího data se zaměřením jen na matematický režim. WP-Quick \LaTeX je aktivně rozvíjen (roky 2011 a 2012; navíc autor pluginu poukázal na možnost zařadit Ti\kern-0.25exZ do matematického režimu) a lze obsáhnout nejen matematický režim. Používá zmíněnou fintu s převodem PDF do SVG nebo rastrových formátů (PNG a GIF).

Lze začlenit preambuli dokumentu i ji rozšířit. Takto vypadá typické užití v příspěvku (tlačítka lze snadno nastavit přes plugin Visual Editor Custom Buttons):

```
[latex]
[+preamble]
\usepackage[utf8]{inputenc}
\usepackage[IL2]{fontenc}
\usepackage{tikz}
[/preamble]
Linka pomocí  $\text{Ti\textit{k}Z}$ :
\begin{tikzpicture}
\draw(0,0)--(2cm,1cm);
\end{tikzpicture}
[/latex]
```

Postup je sofistikovanější. K ukázce se připojí či nahradí preambule dokumentu, výsek se zahashuje a pokud se liší od existujících za \TeX ovaných ukázek v temp adresáři (/q1-cache/) pošle se to na server autora pluginu, tedy na serveru s nainstalovaným WordPressem nemusí být \TeX ová distribuce. Výhoda tohoto přístupu je, že kdyby náhodou jejich server neběžel, tak se obrázky načítají z tohoto adresáře. Cena za tuto službu je, že nemůžeme aktualizovat \TeX ovou distribuci.

Getting started Basic Settings Advanced System About

Activate QuickLaTeX on a page, post, or comment with the shortcode `[late=page]`. Then you may insert LaTeX expressions directly in the text by surrounding them with `$...$` or place them displayed with `\[...]` as you usually do typing offline LaTeX documents.

You may also use display environments `equation`, `align`, `displaymath`, `eqnarray`, `multiline`, `flalign`, `gather`, and `alignat`.

Here is example of a page with LaTeX formulas (how it appears in Wordpress editor):

```
[latepage]
At first, we sample  $f(x)$  in the  $\mathbb{N}$  ( $\mathbb{N}$  is odd) equidistant
points around  $x^*+*$ :
\begin{equation}
f_k = f(x_k), \quad x_k = x^*+kh, \quad k = \frac{N-1}{2}, \dots,
\frac{N-1}{2}
\end{equation}
where  $h$  is some step.
Then we interpolate points  $\{(x_k, f_k)\}$  by polynomial
\begin{equation}
p_{(N-1)}(x) = \sum_{j=0}^{(N-1)} (a_j x^j)
\end{equation}
Its coefficients  $\{a_j\}$  are found as a solution of system of
linear equations:
\begin{equation}
\left( \begin{array}{c} p_{(N-1)}(x_k) = f_k \end{array} \right), \quad \text{quad } k = \frac{N-1}{2},
\dots, \frac{N-1}{2}
\end{equation}
\end{pre}


Getting started Basic Settings Advanced System About



QuickLaTeX processes native LaTeX shorthands on the pages activated by [late=page] tag. However you can activate LaTeX syntax interpretation site-wide (on every post, page and comment). No tags are needed in this mode.



Use LaTeX Syntax Site-wide  OFF



This option activates LaTeX interpretation on all pages, regardless of the use of [late=page]. This is useful for a site that is authored exclusively by LaTeX users. If this is turned on, all authors must be made aware that $ is a special symbol, and that the standard dollar symbol must be typed $.



Exclude $ ... $  ON



The use of the traditional TeX $ conflicts with the non-LaTeX user's expectation of its meaning. This option allows us to turn its interpretation off, so standard dollar signs need no special treatment and will not confuse the non-mathematical user.



Please setup LaTeX preamble for the whole* website below. You can define new commands and include additional packages as usual:



```
\usepackage{amsmath}
\usepackage{amsmaths}
\usepackage{amsymb}
```



Getting started Basic Settings Advanced System About



Set default styling. These settings can be overridden on a per formula basis using \quicklatex{} within its LaTeX code. For more precise tuning, modify quicklatex-format.css.



Font Size



Choose formula font size to match text on your website.



Font Color



RGB triplet in CSS format - six digit hexadecimal number. Please see CSS Colors for examples.



Background Color



Formulas are rendered with transparent background by default. Setup opaque color in CSS format otherwise.



Displayed Equations Alignment



Equation Number Position



Getting started Basic Settings Advanced System About



QuickLaTeX converts formulas into PNG images and tries to cache them on your site for maximum performance. By default QuickLaTeX is tolerant to mistakes in LaTeX code and stops only on critical errors. You can tune these settings here.



Image format



Output image format. 'Auto' is recommended choice.



Cache images locally  ON



Absolutely essential for performance of your site. Also we recommend you to use WP Super Cache for optimal speed.



Debug mode  ON



Forces QuickLaTeX compiler to be strict and show all warnings and errors.



Update QuickLaTeX Settings >


```

Edít button • Košícký bridžový klub — WordPress - Mozilla Firefox

Soubor Úpravy Zobrazení Historie Záložky Nástroje nápověda

new.bridgekosice.sk/wp-admin/post.php?post=264&action=edit

Košický bridžový klub + Přidat

Nástenka

Články

Multimédia

Stránky

Komentáře

Galerie

Profil

Nástroje

Visual Editor Custom Buttons

All buttons Add new

Zbalit menu

Edít button Add new

Pika

Button Content

Wrap Selection Single Block

Content

```
<img src='http://new.bridgekosice.sk/wp-content/ql-cache/quicklatex.com-adc316178dd1252c2613d5b7e774d582_13.svg' width='15' height='19' />
```

Display In Editor

Visual Editor Text Editor

Button Icon

Pika

Custom Icons

Add your custom icons by creating a new folder called `vecb` in the wordpress upload directory and adding your icons (20x20px) there. The correct path for the custom icons should be: `...wp-content/uploads/vecb/`. When added, the icons will automatically show up the Button Icon dropdown menu.

Quicktag Label

If not set, button title will be used.

Obrázek 1. Nastavení pluginu WP-QuickL^AT_EX (horní čtyři okna) a definování tlačítka přes doplňkový plugin ve WordPressu Visual Editor Custom Buttons (spodní snímek obrazovky).

Na produkčním serveru v Košicích jsme záměrně T_EXově vznikající obrázky znaků převedly na načtení PNG obrázků na úrovni HTML. Tím se vlastně obešel plugin, ale zůstaly vypočtené informace o vertikálním posunu rastrových obrázků.

6. Náročnější sazba

Tímto se webové problémy částečně přesunuly a transformovaly na problémy typicky T_EXové. Například můžeme nejen vysázet jeden znak, ale celý list, licitaci, sehrávku či s přepínačem listy jednotlivých hráčů.

Vstupem je PBN soubor, který se parsuje na T_EXové příkazy. Některé věci šly řešit třebaš přes balíček `xstring`, ale parsování na T_EXovou míru bylo snadnější.

Sazba listu hráče se rozpadá na T_EXový příkaz se čtyřmi parametry.

Licitace a sehrávka se rozpadá na sazbu několika řádků či sazbu tabulky.

Podobně nejtěžší sazba celého rozdání je sazba větší tabulky. Ukázka bude přes tabulky uvnitř tabulky, lze užít `TikZ` ap. Ukážeme si jádro zapínání a vypínání listů, to bylo lepší řešit na úrovni T_EXové, resp. úrovni pluginové, než na úrovni běžnému uživateli WordPressu nepřístupné úrovni parsování.

Jedná se o rychlé řešení, kdy v posledním parametru zapínáme (1) či vypínáme (0) konkrétní list. Dalo by se to dál T_EXově zjednodušovat, ale pro naše potřeby bylo řešení dostačující (nechť se soubor jmenuje `ukazka.tex`).

```
\documentclass{article}
\pagestyle{empty}
\def\meano{1}\def\prazdne{}
\begin{document}
\def\jedenstul{%
\begin{tabular}{c}
$\spadesuit$ \ifx\piky\prazdne ---\else\piky\fi \\
$\heartsuit$ \ifx\srdce\prazdne ---\else\srdce\fi \\
$\diamondsuit$ \ifx\kara\prazdne ---\else\kara\fi \\
$\clubsuit$ \ifx\krize\prazdne ---\else\krize\fi \\
\end{tabular}%
} % Konec příkazu \jedenstul.
\def\bgdiagram #1#2#3#4#5#6{%
\viditelnost#6
\begin{tabular}{ccc}
&\parsun#2 % North
\ifx\prvni\meano\jedenstul\fi & \\
\parsun#5 % West
\ifx\ctvrty\meano\jedenstul\fi & My table &
\parsun#3 % East
\ifx\druhy\meano\jedenstul\fi \\
&\parsun#4 % South
\ifx\treti\meano\jedenstul\fi\\
\end{tabular}%
```

Diagram rozdania s priebehom dražby a zohrávky:

Board 1
Deals: North
Vul: none

♠ Q102
♥ 98
♦ 962
♣ AKJ6

♠ AJ54
♥ 76
♦ A85
♣ 8432

♠ K76
♥ AKQ52
♦ Q107
♣ 95

♠ 983
♥ 1043
♦ K43
♣ Q107

North	East	South	West
Houlberg	Arnolds	LundMadsen	van Zwol
Pass	2♥	All pass	

Contract 2♥ by East

Trick	Leader	1.	2.	3.	4.
1.	S	♦3	♦5	♦J	♦Q
2.	E	♥A	♥3	♥6	♥8
3.	E	♥K	♥4	♥7	♥9
4.	E	♥Q	♥10	♣2	♣2
5.	E	♠K	♠8	♠4	♠2
6.	E	♠7	♠3	♠J	♠Q
7.	N	♣A	♣5	♣7	♣3
8.	N	♣K	♣9	♣10	♣4
9.	N	♣6	♣2	♣Q	♣8
10.	E	♠6	♠9	♠A	♠10
11.	W	♦A	-	-	-

Made 3 — EW +140

To isté rozdanie po výnose z pohľadu hlavného hráča

♠ AJ54
♥ 76
♦ A85
♣ 8432

♠ K76
♥ AKQ52
♦ Q107
♣ 95

♦3

Rozdanie z pohľadu severu po výnose

♠ Q102
♥ 98
♦ 962
♣ AKJ6

♠ AJ54
♥ 76
♦ A85
♣ 8432

♦3

Poprosím o pripomienky k prezentovaným diagramom.

Share on [facebook](#) [Google+](#) [twitter](#)

[Print](#)

This entry was posted in [Bridžové programy](#) by [---](#).
[Bookmark the permalink.](#)

Kľúčové slová

2013 [alertovanie bridžmejty](#) [bridžové zásady](#) [dražba](#) [Gejza Boros](#) [história](#) [hodnotenie listu](#) [Karpatský pohár](#) [kartové symboly](#) [kitchínový systém](#) [košice](#) [liga](#) [Magister Michalovce](#) [Milec](#) [minimulti](#) [MŠKovce](#)
[multi](#) [neoprávnená informácia](#) [nerovnovážny výsledok](#) [obrana otvorenia pokutu](#) [poľské dvojfARB](#) [radáčny systém](#) [silný tréner](#) [skórovacie programy](#) [skórovanie](#) [Stará Ľubovňa](#) [stolové](#)
listy stránky KBK terminológia TeX Tibor Menyhért Trussol training Veľká cena Košíc väžený výsledok výsledky výuka zaujímavé rozdania zaťažovníci zohrávka žltina

Obrázek 2. TeXová sazba viděna na produkčním serveru. V horním okně je rozdání, licitace a sehrávka, v okně spodním můžeme vidět přepnutí listů na linku Východ-Západ a Sever-Západ.

```

} % Konec příkazu \bfdiagram.
\def\parsun#1.#2.#3.#4 {\def\piky{#1}\def\srdce{#2}%
\def\kara{#3}\def\krize{#4}}
\def\viditelnost#1#2#3#4 {\def\prvni{#1}\def\druhy{#2}%
\def\treti{#3}\def\ctvrty{#4}}
\bfdiagram{1}{QT2.98.J962.AKJ6}{K76.AKQ52.QT7.95}
{983.JT43.K43.Q107}{AJ54.76.A85.8432}{1111} % Užití 1: levá ukázka.
\bfdiagram{1}{QT2.98.J962.AKJ6}{K76.AKQ52.QT7.95}
{983.JT43.K43.Q107}{AJ54.76.A85.8432}{1010} % Užití 2: pravá ukázka.
\end{document}

```

Zde je vysázená minimalizovaná ukázka (všechny listy a listy dvou hráčů):

\$ pdflatex ukazka.tex

<p>♠ AJ54 ♥ 76 ♦ A85 ♣ 8432</p>	<p>♠ QT2 ♥ 98 ♦ J962 ♣ AKJ6</p> <p>My table</p> <p>♠ 983 ♥ JT43 ♦ K43 ♣ Q107</p>	<p>♠ K76 ♥ AKQ52 ♦ QT7 ♣ 95</p>	<p>♠ QT2 ♥ 98 ♦ J962 ♣ AKJ6</p> <p>My table</p> <p>♠ 983 ♥ JT43 ♦ K43 ♣ Q107</p>
---------------------------------------------	----------------------------------------------------------------------------------------------------------	---------------------------------------------	----------------------------------------------------------------------------------------------------------

Naše první pokusy na produkčním serveru lze vidět zde, komplexnější ukázky lze nalézt zde. Na poslední straně článku uvádíme ukázky viděné z pohledu členů klubu. Potěšilo to, takto vidět \TeX ové výstupy přes známé webové rozhraní. Můžeme si všimnout i nastavené průhlednosti, ta se dá přes plugin vypnout.

Závěr

Pro naše potřeby a experimenty byla \TeX ová makra dostatečná. Jako zajímavost uvádíme, že o jistou XML standardizaci pro bridž se pokusil Thomas Andrews, viz jeho webové stránky <http://bridge.thomasoandrews.com/xml/>.

Další přání bridžistů bylo animovat sehrávku, to v \TeX u lze (balíček `animate`), ale problém by bylo se zpřístupněním na internetu. Nic jiného než PDF či neinteraktivní animovaný GIF by asi nešlo a nenapadlo nás nic alternativního. Třeba někdo tuší a poradí. BBO nabízí lepší řešení, tzv. Handviewer, více o tom zde.

Kontaktní adresy

RNDr. Tibor Menyhért, Bridžový klub Košice, Slovenská republika

Ing. Pavel Stríž, Ph.D., Nakl. Martin Stríž, U Škol 940, 685 01, Bučovice, Česká republika,
E-mailová adresa: pavel@striz.cz

SAZBA NOT V SYSTÉMU MUSIX_TE_X

JIŘÍ RYBIČKA

Abstrakt. Systémy postavené na principu T_EXu se postupně dostaly do mnoha oblastí přípravy dokumentů nejrůznějších typů. Mezi speciální úkoly sazby patří bezesporu příprava notových partů, které se dříve celé kreslily ručně za použití pouze zjednodušených pomůcek. Řešení notové sazby je více, jednou z nejkompexnějších možností je balíček MusiX_TE_X, zaměřený především na vícesnovové a vícenástrojové sazby (klavírní a varhanní party, partitury) klasické hudby. Příspěvek ukazuje základní principy a schopnosti balíčku a možnosti využití.

Klíčová slova. Sazba not, MusiX_TE_X.

USING MUSIX_TE_X TO TYPESET MUSIC

Abstract. Systems based on T_EX principles have penetrated into many areas of document preparation and special typesetting. One of the specific tasks undoubtedly is the preparation of music, which were formerly drew by hand using only very simplified tools. There are more ways to solve this task, but one of the most comprehensive option is the MusiX_TE_X package, focusing primarily on music with multiple staves and more instruments (piano and organ parts, music scores) in classical music area. This paper shows basic principles of this package and its main possibilities.

Key words and phrases. Score typesetting, MusiX_TE_Xpackage.

Úvod

Typografický systém T_EX byl primárně koncipován jako procesor pro automatizaci ruční sazby textu. Obsahuje proto efektivní algoritmy pro řádkový zlom v odstavci a řadu dalších prvků uplatňovaných v běžném textu (ligatury, mezerování, různé penalty apod.).

Sazba not se v mnohém od sazby textu zásadně liší. Uvažujeme-li jen jednu notovou osnovu s jedním hlasem, můžeme snad nalézt určitou paralelu s textovou sazbou, je však třeba tento případ chápat spíše jako výjimku než pravidlo. Běžně je nutné sázet více notových osnov pod sebou, a to u nástrojů, jako je například klavír nebo varhany, kde jsou dvě osnovy pro ruce a případně další osnovy pro nohy, nebo více osnov při sazbě partitur, kde více nástrojů (lidských hlasů, textů) tvoří dohromady souhrn informací o skladbě. Přitom platí jednoduchý princip – co je *pod sebou*, to při provedení skladby *zazní společně*. Notovou sazbu lze tedy chápat jako dvojdimenzionální úkol – více řádků materiálu se svislým zarovnáním.

Druhou podstatnou odlišností od běžného textu je zcela originální sada znaků, symbolů a nejrůznějších vazeb tvořících dohromady dokonalý obraz notového partu.

S obojím je zároveň spojena řada pravidel – podobně jako existují typografická pravidla pro sazbu obyčejného textu, je potřebné respektovat řadu pravidel pro sazbu not. Tato pravidla mají stejný účel jako v textu – čtenář not nesmí být nikde na pochybách, jak má záznam číst, a pokud možno má být ještě part uspořádán tak, aby čtení bylo co nejjednodušší. K tomu přispívají ustálené a léty prověřené zvyklosti, jak se jednotlivé prvky a celky vyjadřují.

Co je MusiX_{TEX}?

Představíme-li si na základě úvodní části, jaký úkol musí systém pro sazbu not řešit, zcela jistě dospějeme k závěru, že použít systém TEX nebude nijak snadné a že spíše bude potřebné řadu jeho „přirozených“ vlastností potlačit než využít.

Do jisté míry je to pravda, ale síla TEX u není jen v algoritmech a prvcích pro textovou sazbu, nýbrž také v možnostech tvorby příkazů umožňujících zcela změnit chápání vstupu a umožňujících řešit řadu obecných úloh. A celý balík MusiX_{TEX} tuto sílu demonstruje.

MusiX_{TEX} má svou pohnutou historii – je to následník původního balíku Music_{TEX}, který začal vytvářet Daniel Taupin kolem roku 1991. Po jeho tragické smrti v roce 2003 byly prováděny jen drobné korekce, později však byly provedeny větší změny, výrazné doplnění manuálu a přizpůsobení novým distribucím TEX u. Z těchto důvodů zde bylo možné pozorovat řadu negativních jevů se zpětnou nekompatibilitou některých maker. Nyní lze však předpokládat víceméně standardní chování a množství změn by mělo být již malé.

Základní principy balíku však zůstávají stejné: způsob sazby not, jejich tvary (součástí jsou i speciální fonty obsahující všechny potřebné elementy) a práce s mezerami a zarovnáním. Balík je určen především pro víceosnovové a vícenástrojové sazby, s čímž je spojena relativní složitost a obecnost.

Jak na to...

Nejprve si balíček připojíme příkazem `\usepackage{musixtex}`.

Sazba not je chápána jako sazba dvourozměrného pole: nástroje a hlasy tvoří řádky, tóny znějící zároveň tvoří sloupce. Z historických důvodů se tedy hudba zapisuje po sloupcích čtených *zespodu*, z těchto sloupců se pak skládají jednotlivé řádky partitury zleva doprava.

Elementem sazby je sloupec uzavřený do páru maker `\notes` a `\enotes`:

```
\notes nástroj1 & nástroj2 & ... \enotes
```

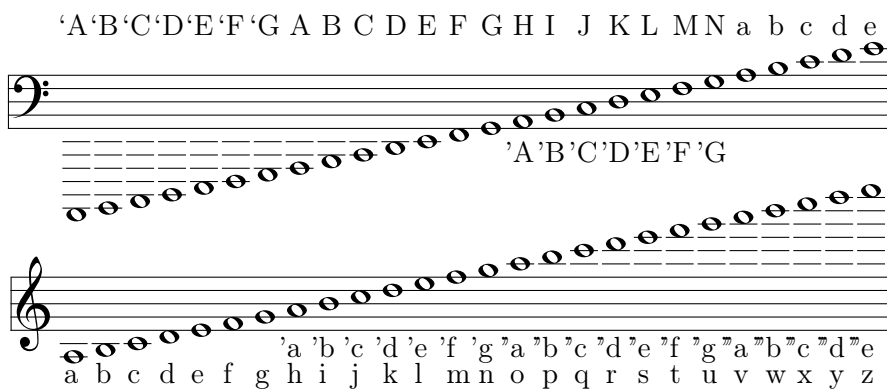
Má-li nějaký nástroj více osnov, odděluje se materiál jednotlivých osnov svislítkem.

Příkaz `\notes` má více variant, které určují mezerování za příslušným sloupcem. Čím více velkých písmen, tím větší mezera: `\Notes`, `\NNotes` apod.

Příkaz `\enotes` lze zkrátit na `\en`.

Jednotlivé noty jsou určeny výškou (fyzikálně vyjádřeno zvukovou frekvencí) a délkou trvání (celé, půlové, čtvrtové atd.). Délka noty je dána příslušným příkazem, jehož parametrem je výška noty: `\wh` je celá nota, `\hl` nebo `\hu` je půlová nota s nožičkou dolů nebo nahoru, `\ql` a `\qu` čtvrtová, `\cl` a `\cu` osminová, `\ccl` a `\ccu` šestnáctinová atd.

Výška noty je dána písmenem, jak ukazuje velmi pěkně zpracovaný přehled převzatý z [1, s. 104] a uvedený na obr. 1.



Obrázek 1. Výšky not.

K vysázení kousku hudby (tzv. extraktu) nám ještě chybí několik drobností: nastavení počtu osnov, klíčů, taktu apod., což můžeme v příkladu demonstrovat následující ukázkou:



```
\begin{music}      % prostředí (skupina)
\parindent 13mm
\instrumentnumber{1} % jeden nástroj
\setclef1{\bass}    % nastavení klíčů
\setclef2{\treble}
\setname1{\bfseries Klavír} % název Klavír
\setstaves1{2}      % se dvěma osnovami
\generalmeter{\meterfrac{4}{4}} % zvolen takt 4/4
\startextract       % začátek sazby osnovy
\notes \qu{'C}| \hl j\en
\notes \ql{'G}| \en
\notes \qu{'E}| \hl l\en
\notes \ql{'G}| \en
\endextract
\end{music}
```

Víceřádková sazba

Notový úsek delší než jeden řádek již není vymezen makry `\startextract` a `\endextract`, ale `\startpiece` a `\endpiece` (nebo podobnými variantami).

Mezerování mezi notami v případě delšího úseku už nemůže být dáno jen naturální vzdáleností definovanou příslušnou variantou příkazu `\notes`, dochází zde k procesu podobnému zarovnání řádků do bloku při sazbě odstavce. Situace je však podstatně komplikovanější – notová sazba by neměla „zlomit řádek“ uvnitř taktu, celá sazba by měla vycházet na celistvý počet vyplněných řádků. Na tento úkol samotný překlad v `TEX`u nestačí, je zde nutná podpora externího programu.

Při prvním průchodu překladem `TEX`u se posbírají potřebné informace, následně se zpracují externím programem a ve třetím kroku se provede finální sazba. Přenos informací se děje textovými soubory s rozšířeními `*.mx1` a `*.mx2`. Schematicky lze proces znázornit takto:

$$\text{T_EX} \xrightarrow{\text{mx1}} \text{musixflx} \xrightarrow{\text{mx2}} \text{T_EX}$$

Dalším důležitým parametrem, který ovlivňuje celkový výsledek víceřádkové sazby, je `\mulooseness`. Je jistou myšlenkovou obdobou parametru odstavcové sazby `\looseness` umožňující zkrátit nebo prodloužit sazbu odstavce o n řádků. Podobně například zadáním `\mulooseness=-1` můžeme snížit původní počet řádků notového úryvku o jeden řádek. Řádkem zde samozřejmě myslíme systém osnov pro všechny nástroje partitury.

Malý příklad sazby s mezerami vypočtenými pomocí programu `musixflx` navíc ukazuje další z mnoha možností. Vlevo i vpravo je stejný materiál, ale v ukázce vlevo je nastaveno `\mulooseness=1`:

Zdrojové texty ukázek jsou následující:

```
\begin{music}
\smallmusicsize
\generalmeter\meterC
\generalsignature{2}
\startpiece
\Notes \upz o\ql{oom}\qp\en\bar
\Notes \upz o\ql{oom}\qp\en\bar
\Notes \ql{oopo}\en\bar
\Notes \hl{on}\en\mulooseness=1
\Endpiece
\end{music}
```

```
\begin{music}
\nobarnumbers\smallmusicsize
\generalmeter\meterC
\generalsignature{2}
\startpiece
\Notes \upz o\ql{oom}\qp\en\bar
\Notes \upz o\ql{oom}\qp\en\bar
\Notes \ql{oopo}\en\bar
\Notes \hl{on}\en
\Endpiece
\end{music}
```

Trámce, oblouky

Trámce a oblouky spojují více not, z toho důvodu musí přesahovat přes více „sloupců“ tvořených páry `\notes... \enotes`. Trámce mohou být i vícenásobné, mohou různě začínat a různě končit. Základní koncept tedy předpokládá, že určíme začátek trámce, jeho umístění v osnově a požadovaný sklon. K tomu existují příkazy typu `\ibu{n}{p}{s}`, v nichž n je referenční číslo trámce (na ně se můžeme odkazovat v dalších místech), p je umístění v osnově a s je sklon. Vícenásobný trámec vyrobíme analogickým příkazem `\ibbu`, `\ibbbu` atd. Písmeno „u“ může být zaměněno za „l“ a dostaneme trámec dole.

Noty příslušející k trámci musí mít nožičky spojené s tímto trámcem, nemůžeme tedy použít běžného příkazu pro sazbu noty, ale speciálního `\qb{n}{p}`, kde n je referenční číslo trámce a p je umístění hlavičky noty v notové osnově.

Konec trámce zajišťuje makro `\tbu{n}`. Když to tedy shrneme všechno dohromady, dvě noty spojené dvojitým trámcem budou zapsány takto:



```
\Notes\ibbu 0c3\qb 0c\en
\Notes\tbu 0\qb 0e\en
```

Je asi zřejmé, že tento mírně komplikovaný zápis se bude dost často opakovat, proto jsou k dispozici různé zkratky ve značném množství. Uvedeme alespoň několik příkladů – viz [1, s. 24]:



```
\startpiece\nobarnumbers
\Notes\Dqbu gh\Dqbl jh\en\bar
\notes\Dqbbu fg\Dqbb l hk\en\bar
\Notes\Tqbu ghi\Tqbl mmj\en\bar
\notes\Tqbbu fgj\Tqbb l njh\en\bar
\Notes\Qqbu ghjh\Qqbl jifh\en\bar
\notes\Qqbbu fgge\Qqbb l jhgi\en
\endpiece
```

Oblouky mají podobný charakter – začátek `\isluru` nebo `\islurd` s označením oblouku referenčním číslem a umístěním v osnově, ukončení pomocí `\tslur` a číslo oblouku:



```
\NOTes\islurd0g\qu g\tslurd0d\qu d\en
```

Slova písně

Pokud bychom sázeli díla s texty, určitě využijeme rozšíření možností sazby textu pomocí balíčku `musixlyr` (viz též [2]). Slova však můžeme umístit do notového systému i základními prostředky `MusiXTeXu`, k čemuž využijeme zejména příkazy `\zcharnote`, `\lcharnote` nebo `\ccharnote`. Všechny mají stejný princip – umísťují libovolný materiál jednořádkového textového boxu do notové osnovy na určené místo, první z nich zarovnává vlevo, druhý vpravo, třetí na střed vzhledem k místu, kde se vyskytuje. Uveďme stručný příklad:



```
\NOTes\zcharnote{L}{Je -}\qu h\en
\NOTes\zcharnote{L}{de,}\upz j\ql j\en
\NOTes\zcharnote{L}{je -}\qu f\en
\NOTes\zcharnote{L}{de}\lpz h\qu h\en
```

Zde se přímo nabízí, že bychom si zápis `\zcharnote{L}{Je -}` zkrátili vlastním příkazem, například `\tx{Je -}`, kde využijeme definice

```
\def\tx#1{\zcharnote{L}{#1}}
```

a máme tak zároveň možnost ovládat veškeré texty po vzhledové stránce z jednoho místa. Můžeme také využít již definovaného příkazu `\zsong`, který má tu výhodu, že text umísťuje do vhodně zvoleného místa pod notovou osnovu.

Velikost textu a související noty však musíme doladit „ručně“. Pokud máme například textu mnoho a pod noty se nevejde, sáhneme po nějakém způsobu oddálení not – místo `\NOTes` použijeme `\NOTEs` apod.

Argument příkazu `\zcharnote` je tzv. LR box, tedy pro jednořádkový materiál. Není ovšem žádný problém sem umístit `parbox` nebo `tabulku` a můžeme sázet více hlasů nebo text, v němž mohou nastat řádkové zlomy. Vytvořme si například `\def\tx#1{\zcharnote{L}{\begin{tabular}{@{}l@{}}#1\end{tabular}}}` a použijme pro sazbu více slok:




```
Vy-le tě- la ho-lu bič-ka
Kdyby by-ly modré o- či
Ne-by- la to ho-lu bič-ka
Když mi- lo- va- ti
mne nechceš
```

```
\generalmeter{\meterfrac 44}
\generalsignature{4}\noindent
\startextract
\Notes\tx{Vy-le- tě- la\\Kdyby by-ly\\
Ne-by- la \ to\\
Když \\quad mne nechceš}%
\Qqbu efgh\en
\Notes\tx{ho-lu- bič-ka\\modré o- či\\
ho-lu- bič-ka\\mi- lo- va- ti}%
\Qqbu ijki\en
\endextract
```


Příkazy negenerující mezery

Na uvedeném příkazu `\zcharnote` jsme mohli pozorovat, že jeho uvedením se notová sazba neposunuje. Místo, kde je uveden, může být obsazeno více různými prvky. Tento princip je opakován u dalších příkazů, které mají obvykle prefix „z“. Například sazbu akordu musíme složit z not, které neposunují sazbu (jsou nad sebou) a z poslední noty, která s sebou nese i následující mezeru. Malá ukázka:



```
\NOTes \zq{fhj}\qu m\en
\NOTes \dpz c\zq{ceg}\cu j\en
```

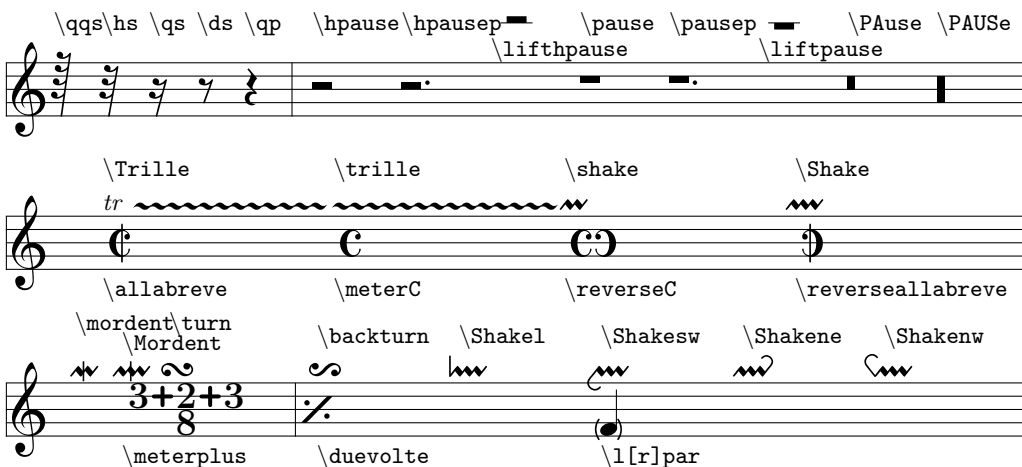
V ukázce je názorně vidět, jak funguje příkaz pro vytvoření noty bez mezery `\zq`. Dalším příkazem v ukázce, který rovněž neposunuje sazbu, je `\dpz` (staccato) a spolu s ním všechny další akcenty. Pravidlem je, že při sazbě napřed použijeme všechny příkazy neposunující sazbu a jako poslední pak notu, která zároveň vytvoří následující mezeru. Příklady not vytvářejících příslušnou mezeru jsou shrnuty v následujícím přehledu:



```
\notes\wh c\en\NOTes\whp e\en
\notes\hu d\en\notes\hl j\en
\notes\qu d\en\notes\ql j\en
\notes\cu d\en\notes\cl j\en
\notes\ccu d\en\notes\ccl j\en
\notes\cccu d\en\notes\ccccl j\en
\notes\ccccu d\en\notes\cccccl j\en
```

Všehochuť na závěr

Uvedeme několik dalších užitečných příkazů, které jsou přehledně umístěny v ukázce převzaté opět z [1, s. 104].



```
\qqs\hs \qs \ds \qp \hpause\hpausep \pause \pausep \PAuse \PAUSE
\liftpause \liftpause
\Trille \trille \shake \Shake
\allabreve \meterC \reverseC \reverseallabreve
\mordent\turn \Mordent \backturn \Shakel \Shakesw \Shakene \Shakenw
\meterplus \duevolte \l[r]par
```

`\setvoltabox` `\setvolta` `\coda`
`\metron` 1. 2.

`\Coda` `\segno` `\Segno` `\caesura` `\cbreath` `\PED` `\sPED` `\DEP` `\sDEP`

`\fermataup` `\arpeggio d5` `\octfinup` `\boxit A` `\circleit B`
`\Fermataup` `\bracket` `\uptrio` `\leftrepeat` `\rightrepeat`
`\fermatadown` `\octfindown` `\leftrepeat` `\rightrepeat`
`\fermatadown` `\leftrepeat` `\rightrepeat`

Závěr

Na velmi malém prostoru tohoto příspěvku se nemůžeme věnovat všem prvkům tak rozsáhlého balíku, jakým MusiX_{TEX} bezesporu je. Jeho 140stránkový manuál vyžaduje pečlivé přečtení, pokud bychom chtěli vytvářet notovou sazbu netriviálních klavírních nebo vícenástrojových partů. Lze rovněž využít řady rozšiřujících stylů, které jsou v celém balíčku přítomny (speciální kytarové prvky, prvky pro středověké zápisy, zvětšení některých možností trylků, trámců atd.)

Článek by měl tvořit jednoduchý návod, jak a kde začít. Pokud čtenář využije možností popisovaného balíku, dosáhne jistě precizního a trvanlivého výsledku.

Reference

- [1] Taupin, D., Mitchell, R., Egler, A. MusiX_{TEX}. Using \TeX to write polyphonic or instrumental music. 2011. Dostupné na <http://texdoc.net/texmf-dist/doc/generic/musixtex/musixdoc.pdf>, též v distribuci \TeX Live příkazem `texdoc musixtex`.
- [2] Dunker, R. `musixlyr` – a MusiX_{TEX} Extension Package for Lyrics Handling (Version 2.1c). 2003. Dostupné na <http://icking-music-archive.org/software/musixtex/addons/mxlyrdoc.pdf>, též v distribuci \TeX Live příkazem `texdoc musixlyr`.

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/>

JAK SE T_EXISTA MEZI DIVADELNÍMI OCHOTNÍKY NEZTRATÍ

PAVEL STRÍŽ (CZ)

Abstrakt. Autor příspěvku doslova divadelně zabloudil mezi bučovické ochotníky, a ukázalo se, že i T_EXnik se může zložit. Představíme si sazbu značek mimo sazební zrcadlo přímo v T_EXu, to se hodí například při sazbě skript, podkladů pro kombinovanou formu výuky a vlastních divadelních her (prosté číslování i poznámkování replik), ale i pro případ, že je na sazebním vstupu PDF soubor vzniklý mimo T_EXové prostředí. To byla i řešená situace a bude možné nahlédnout na studovanou hru, chceme-li, na sazební výstup. Poznámky technika k jednotlivým značkám (hudba, zvuky a zvukové efekty) se také mohly sázet automatem, ale pracovně skončily „jen“ na webové stránce, viz <http://pomocnik.striz.cz/#skladby>. Během přednášky si pustíme několik ukázek ze hry *Stráždlo cantervillské* od Oscara Wilda a to svobodně v přehrávači MPlayer a MPlayer2.

Klíčová slova. T_EX, sazba, divadlo, ochotníci.

HOW A T_EXIST IS NOT LOST AMONG THEATER AMATEURS

Abstract. Some time ago, I have more or less accidentally met theater amateurs in Bučovice city (the Czech Republic, the South Moravia), I joined them and it showed to be true that a T_EXist can be a helpful member of such a group. The author introduces typesetting on the margin side with a possibility of cross-referencing counters. We use this approach in textbooks, books and sometimes in the theater plays. In the article we show a real application of typesetting symbols into an external PDF file. The other typesetting material was used on the author's webpage only. During the talk we will listen to music, sounds and sound effects from a family horror story/drama studied these months: *The Canterville Ghost* by Oscar Wilde, freely in MPlayer and MPlayer2.

Key words and phrases. T_EX, Typesetting, Theater, Theater Amateurs.

Úvodem vlastní značka

Rádi bychom na úvod demystifikovali záhadu tvorby stylů, chceme-li logických a vizuálních značek. Jde se na to přes `\def` (Plain T_EX) či `\newcommand` (L^AT_EX).

```
% Do preamble: \usepackage{xcolor}
\def\znacka#1{Má {\color{red}vlastní} {\color{green}značka} s texty: #1.}
\znacka{Pokus}
```

Vysází se: Má **vlastní značka** s texty: Pokus. Touto cestou můžeme dělat libovolně jednoduché, ale též libovolně složité styly. Součástí příkazů může být libovolný sazební podklad (texty, matematika, obrázky). Značky si lze představit jako substitute, správně terminologicky řečeno jako T_EXové expanze.

1. Vystrčení objektu

V mnoha případech potřebujeme sázet nad (diakritické značky, intonace), pod (podtržení), vlevo (šipky), vpravo (čtvereček či Q.E.D., z latiny quod erat demonstrandum, česky: což mělo být dokázáno) od sazebního podkladu, ale častokrát potřebujeme sazební podklad dát mimo sazební zrcadlo. Běžně se používá příkaz `\marginpar`. Odkazujeme na literaturu. Zkusíme si to sami na začátku odstavce.

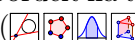

Pokus Tato ukázka byla vysázena tímto příkazem na začátku odstavce:

```
\makebox[Opt][r]{Pokus\hspace*{\parindent}}%
```

Můžeme si odskočit na druhou stranu sazebního zrcadla.

Zde

```
\makebox[Opt][l]{\hspace*{-\parindent}\hspace*{\textwidth}Zde}%
```

Touto krátkou ukázkou se nejčastěji vystrkují příjemné a barevné symboly (otazníky, vykřičníky, symboly pro knihy, postavičky ap.) pro skripta pro kombinovanou formu výuky. Lze doporučit na testy např. ikonky dělané v TikZu (vektor, původní rastr) pro GeoGebru () či méně známý balíček bclogo ().

2. Křížové odkazy křížově

Podobně můžeme zařadit do značky čítač. Například seznam úkolů, nápadů, myšlenek, osob ap. Jak je aktivovaná značka, tak si upraví čítač. \TeX má své čítače, ale jsou situace, kdy je lepší si udělat extra čítač.

Zde si ukážeme přístup křížení u křížových odkazů. Používá se to u replik. V textu se odkazujeme na komentáře, které jsou v závěru hry. A v závěru hry se odkazujeme na stranu, kde byla replika zmíněna. Podobný přístup znají akademičtí pracovníci u seznamu literatury, kdy například u referencí máme seznam stran, kde byla kniha citována.

```
% Iniciace čítače
```

```
\newcounter{citac}
```

```
\setcounter{citac}{0}
```

```
% Představení
```

```
\refstepcounter{citac}
```

```
\label{zacatek:\thecitac}
```

```
Vysvětlení je na str.~\pageref{konec:\thecitac}.
```

```
% Vysvětlení
```

```
\setcounter{citac}{0} % Nezbytné vynulování
```

```
\refstepcounter{citac}
```

```
\label{konec:\thecitac}
```

```
Představení je na str.~\pageref{zacatek:\thecitac}.
```

Pak už jen stačí příkazy bloku Představení a Vysvětlení obalit do kratších příkazů, které si dle potřeby voláme. Princip je tedy takový, že vedle křížového odkazu si voláme budoucí křížový odkaz a obráceně. Je to situace pro badatele. Na hloubější zkoumání lze doporučit např. balíček `refcount`.

Autorské poznámky

- [2] *Strana 31. O sebevrahoti. První třetina.* Na tomto místě, před začátkem (ve středu a na konci) dvojhy, je rozložena ve Šbirec absurdní minihra (antimihra) o sebevrahoti a jeho neúspěšných devíti pokusech umřít. Sebevrah nehraje žádnou roli ve dvojhře. On přihlížející nevidí a neslyší. Ale minihra je jen o mimice a gestech, bez mnoha slov, a necht' si ji začatí zkušební divadelní režisér, uzná-li tak za vhodné. Přihlíží, sází si na něj, kazí a smje se mu samotný Bůh spolu s Dáblem; za posmívání a sledování, pobíhání a smíchu svatých i nevatých stvoření. Bůh a Dábel sedí ale židličkách, ostaní stojí a fandí sebevrahoti. Je to opak celé dvojhy, člověk chce, ale není mu umožněno zemřít. První tři pokusy jsou (další tři jsou na straně 90 a poslední tři jsou na straně 161):
- 1) oběšení,
 - 2) zastelení,
 - 3) podřezání žil.
- Může na úplném začátku říci: „Chci se tady a teď zabít!“
- [10] *Strana 32. Němec Oto.* Němci dělali, dětaji a budou dělat problémy na Zemi, v Nebi i Pekle. Bráno sarkasticky, samozřejmě.
- [21] *Strana 33. Zde se s mým divadelním opomentem strhla jedna z těch větších diskuzí.* Můj plán bylo rozebrat byrokraci a naznačit chaotičnost a nefunkčnost větší posmrtných. Něchtěl jsem jít do extrémů. Oponent byl toho názoru, že to není ono a ať předělám. Já byl proti. V pracovní verzi existovaly navíc vulgární výrazy, které mi tam také vadily, ale nevěděl jsem, co s nimi. Poté jsem výrazy upravil, ale do struktury textu jsem nezasáhl. Obhájil jsem svůj plán reprezentovat nesmyslnost a zbytečnost Rozcestí a považmo celé první hry. Systém je, ale nefunguje. Bezdělně však funguje neefektivnost, duplicita a užívání si bohémského života; pokud není Bůh nabližku. Bůh je kapitolu sama o sobě. Ryba smrdí od huby, co dočítat.
- [31] *Strana 36.* První, co člověk vidí před příchodem na Rozcestí, je sám Dábel. To je radost, že? Má mnoho podob a tváří. Není to Dábel z nejnižší vrstvy Pekla (dle Danovy vize). Nejnížší vrstvy se bojí. Chová se jako milión let staré dítě, puberták. Kde jsou ostatní? Nejsou, to byla ta bitva před miliónem let s Nebem. Je to takový stroek, ale zodpovědnost má dost. Ve druhé třet dvojhy je spíše drsným mužem.
- [34] *Strana 36.* Poslední obraz reprezentuje poslední vjem ze Země. Obrázky mužů byt čistě a z vosku. Vosk je vhodný materiál k umělecké tvorbě.
- [39] *Strana 36.* Když nevíme, jestli je Dábel muž nebo žena, tak ani nevíme, jestli je sexuálně orientován na muže, nebo ženy, nebo obojí.
- [68] *Strana 39. Říkal!*
- [90] *Strana 40. Bradě* je olympijskou hrou. Česká republika pravidelně vysílá týmy. Mj. bradě je technicky jedna z nejlépe propracovaných karetních her. Získala si své srdce mnoha lidí, včetně autora. Hrává se nejčastěji v karetních klubech; ve Zlíně takový zádný není. V Uherském Hradišti jeden klub je. Nebud' te však zklamání, protože bradě lze hrát i přes Internet. Mjji oblibený server je <http://game.s.yahoo.com/>. Hrát můžete po internetové registraci.
- [101] *Strana 41.* Také Edj je jen služebník, i když svatý.

- To Rozcestí, to byl taky nápad! Jak můžou svatí rozhodovat tak důležité věci, jako je posláni obřizných případů sem, nebo do Nebe, nebo zpátky na Zem? To má rozhodovat jen Bůh, Nikdo jiný; ale nedá se, práče je moc. Svatý je jednou svatý a konec. Já se snad zase namíchnu. Zrudnu a bude mi celé desetiletí zle. Ne, budu klidný — to se mi stalo za druhé světové a to mi stačilo. Ted' už budu klidný, žádná genocida mě už nerozhodí.
- Abych tedy pořádně vysvětlil to Rozcestí. On když člověk pořádně umře, tak frčí přímo do Nebe, nebo přímo k nám.
- No jo, jenže když ten člověk pořádně neumře, tak se musí rozhodnout takový případ zgruntu. Zvážit všechna pro a proti, všechna fakta. Pak teprve rozhodnout, jestli pofrčí k nám, do Nebe, nebo zpátky na Zem. I takové případy tu byly. Ono Rozcestí je taková brána mezi *(plešká se rukou do hrudi) mým* Podsvětím a *(ukazuje rukou nahoru) jeho* Nebem. Jsem rád, že Boha a ty jeho svatě potkávám, jen tam. A tam chodím, jen když se nudím, a chci se nadýchat toho jejich svatého vzduchu. Pozdravít, pokecat se svatýma, a pak hezky domů. Do Pekla. —
- Obecně se myslí, že cesta na Rozcestí je nekonečně dlouhá. Jako v geometrii průsečík dvou rovinněček protínajících se v nekonečnu. Tam, v nekonečnu, že jsou ty dveře. Kdepak. Je to tu normální cesta. . . . zde dřevo . . . zde kasy železa, tam je to rozflákané . . . —
- Však se podívejme, jak to tam na Rozcestí vypadá. —
- (Bílá opona jde nahoru. Je vidět naprosto normální kancelář. Několik stolů, židle, nějaký ten počítač. Od země až po strop spisy a knihy, dále haldy krabic, papíry a lejsier. Vlevo v zadním rohu jsou dveře s nápisem Peklo, vpravo v zadním rohu jsou dveře s nápisem Nebe. Vpravo v předním rohu jsou ještě jedny dveře s nápisem Země.)*
- (Za stolem sedí úředník. Úředník je skryt za hromadou krabic a papírů, jsou jen drobné vidět nohy a ples hlavy (po úroveň čela, níž mu vidět není), zbytek je ukryt za hromadou papírů, ve které se později přehrabuje.)*
- Ďábel vypravěč** > Však to vidíte, kanclík jako tučty jiných. Ekonom a vše to zoptimalizovat. Je to chaos, nelogika, nepořádek, nesmyslnost

Obrázek 1. Ukázka divadelní hry s obousměrnými křížovými odkazy, viz odstavec [21].

3. Víc o sazbě divadelních her

Většinou si zvýrazníme postavu, obvykle přes `\textbf`, a pak se setkáme s různými přístupy vysázení komentářů, obvykle jak se má co říci či udělat, setkáme se například se složenými závorkami a přes `\textit`. Obvykle se vše vypisuje slovy, v dnešní době by mladší čtenáře nepohoršily emotikony. Upozorníme na nový balíček (březen 2013) `tikzsymbols` od Bena Vitecka. Vzorky: 😊 😢 😡 🍷 a další.

```
\dLaughey \dNursey[] [yellow] [blue] [red] \dSadey[] [red] \dSnowman
```

Zájemci o praktickou ukázkou této sazby necht' nahlédnou například na PDF *Sbírky divadelních prvotín*. Ukázka je na předchozí straně. Více k sazbě divadelních her hledejte prosím ve starším článku ve *Zpravodaji C&TUG* (1–2/2009).

Zajímavý problém je výpočet množství textů replik jednotlivých postav. To jsem neřešil, i když se \TeX ový zdrojový kód dá parsovat a exportovat, elegantnější řešení nabídl Zdeněk Wagner přes XML (zatím nepublikováno).

4. \TeX nik technikem

Ochotníkům pomáhám s výběrem a pouštěním hudby, zvuků a efektů. K tomu si potřebuji dělat poznámky co a kde pustit. Ačkoliv jsem obdržel hru jako soubor Wordu, přesazba by musela držet stránkové a řádkové zlomy, aby všichni účastníci měli stejné texty. Proto jsem užil PDF a do něj sázel symboly dle potřeby.

Jedná se o absolutní umístování na straně. To umí *TikZ*, ale situace je komplikovanější v tom, že takto potřebujeme rozhodit symboly na různé strany, nikoliv jen na jednu. Proto jsem si udělal jeden pomocný soubor, který se díky \TeX ovému balíčku `fancyhdr` načítá na každé nové straně a vždy znovu. Pokud symbol patří na novou stranu, vysází se. Takto vypadalo naše rozšíření preambule:

```
\usepackage{tikz}
\usepackage{pdfpages}
\usepackage{fancyhdr}
\pagestyle{fancy}
\fancyhf{}
\renewcommand{\headrulewidth}{0.0pt}
\renewcommand{\footrulewidth}{0.0pt}
```

Jádro příkazu `\umisti` (umístí značku) vypadá takto:

```
\def\umisti #1 #2 #3 #4 {% Co-sázet Strana Pozice-na-straně Popisek
\ifnum\thepage=#2% Patří to na tuto stranu?
\begin{tikzpicture}[remember picture, overlay]
\node (moje-#2-#3) [circle, thick, draw=black, fill=yellow,
xshift=1cm, yshift=-#3cm, minimum width=1cm] at
(current page.north west) {\thepage#1};
\end{tikzpicture}%
\fi% Konec \ifnum.
}% Konec příkazu \umisti.
```

V těle dokumentu byly jen tyto dva řádky:

```
\fancyhead[C]{\input sazet.tex}
\includepdf[pages={-},pagecommand={}] {original/sc.pdf}
```

Struktura příkazů v souboru `sazet.tex` vypadá takto:

```
\umisti a 3 1 prichody-lidi
\umisti b 3 2.5 predehra
\umisti c 3 26.5 ne-cervi
```

Na straně třetí se vysází značky ve vertikální pozici 1, 2,5 a 26,5 centimetrů od horního okraje papíru a jeden centimetr od levého okraje (to zařizuje parametr `xshift`). Takový soubor jsem si editoval. T_EX je potřeba spustit dvakrát.

5. Praktické ověřování

Tímto způsobem jsem si připravil PDF podklad se značkami a dle potřeby je přidával, odebíral a posouval. Značky dle zvyklostí patří k poznámkám, ale v mnoha případech je lepší si to dát o řádek či dva výš, aby spuštění hudby nebylo pozdě, aby člověk nezaspal.

Funguje mi to tak, že skript mi generuje sérii dávkových souborů na spuštění hudby. Nevěnuji se editaci hudby a šel jsem na to jinak. Vyzkoušel jsem řadu přehrávačů médií, ale žádný nebyl programovatelný. Kdyby nebyl tento požadavek, přehrávačů je mnoho, od VLC počínaje po Ampache a YouTube konče.

Částečně by vyhovoval Xine, ale ten můj hardware či knihovny nezvládaly (chrástilo mi to při přehrávání). Také to vypadalo, že to nezvládá některé nezbytnosti. Těmi byly pustit hudbu od do, opakově, zpomalovat skladbu, s nekonečnou smyčkou, parametrem nastavit hlasitost ap. To vše splnil až přehrávač MPlayer (repozitář `mplayer`). Navíc program obsahuje MEncoder, takže jsem rovnou převáděl i zvukové formáty. Zde je ukázka na extrakci zvukové stopy z FLV souboru.

```
> mplayer.exe -dumpaudio jazz.flv -dumpfile jazz-paja.mp3
```

Na extrakci a výřez hudební stopy z rozsáhlého videa jsem používal VirtualDub. Vše musí fungovat pod Microsoft Windows kvůli ostatním ochotníkům, pro případ kdybych nemohl pomáhat a musel mě někdo zastoupit.

Pustit písničky za sebou znamená jen spustit přehrávač několikrát, pustit písničky přes sebe realizují takto:

```
> start /min cmd.exe /k call 03-a-predehra.bat
> start /min cmd.exe /k call 03-b-teremin.bat
```

Samostatné BAT soubory si řídí další chod, především pozastavení a sled zvuků.

Můj největší problém v té chvíli byla nezastavitelná aktualizace seznamu písem, která trvala dlouhé, nekonečné a hlavně nepřijatelné vteřiny. To jsem si nemohl dovolit a přešel jsem na přehrávač MPlayer2 (repozitář `mplayer2`) s přepínačem `--font=./mplayer/subfont.ttf`, případně přes užití `fc-cache`, což snad zabere. Ukázky ze studovaného dramatu si pustíme na OSSConf 2013.

6. Klikatelnost PDF

Vše mi fungovalo. Dokud nám nedošlo, že bude v rohu technika naprostá tma a nebudu moci sledovat text z papírů. Tma je klíčová, zvlášť u hororového dramatu. Byl na mě vznesen tip, jestli neudělat záložní PDF klikatelné – to jsem si dělal jen pro případ, kdybych zapomněl vytištěnou verzi, kdybych zapomněl notebook, tak by hudba nebyla vůbec. To umíme přes balíček `hyperref` (aktualizace 12/2012). Problém je v tom, že o tuto klikatelnost přijdeme, když absolutně umístujeme objekt přes `TikZ` na stránce, zde je minimální ukázka na ověření nefunkčnosti.

```
\documentclass{article}
\usepackage{tikz,fancyhdr,pdfpages}
\pagestyle{fancy}
\usepackage{hyperref}
\begin{document}
\fancyhead[C]{%
\href{http://ossconf.soit.sk/}{%
\begin{tikzpicture}[remember picture, overlay]
\node[circle,draw,fill=yellow] at (current page.center) {Klikni sem!};
\end{tikzpicture}}%
}% Konec \href.
}% Konec záhlaví.
Sazba.%\includepdf[pagecommand={}]{}{empty.pdf}
\end{document}
```

Jsme schopni získat klikatelný text uvnitř uzlu (`\node`), nikoliv však celé pohyblivé prostředí `tikzpicture`. Řešením je užít `TikZové` prostředí uvnitř `TikZového` prostředí, je to podobná finta jako dělat tabulky uvnitř tabulky.

```
\documentclass{article}
\usepackage{tikz,fancyhdr,pdfpages,hyperref}
\pagestyle{fancy}
\begin{document}
\fancyhead[C]{%
\begin{tikzpicture}[remember picture, overlay]
\node at (current page.center) {%
\href{http://ossconf.soit.sk/}{%
\begin{tikzpicture}
\node[circle,draw,fill=yellow] {Klikni sem!};
\end{tikzpicture}}%
}% Konec \href.
};% Konec \node.
\end{tikzpicture}}%
}% Konec záhlaví.
Sazba.%\includepdf[pagecommand={}]{}{empty.pdf}
\end{document}
```

7. Spuštění dávky z PDF

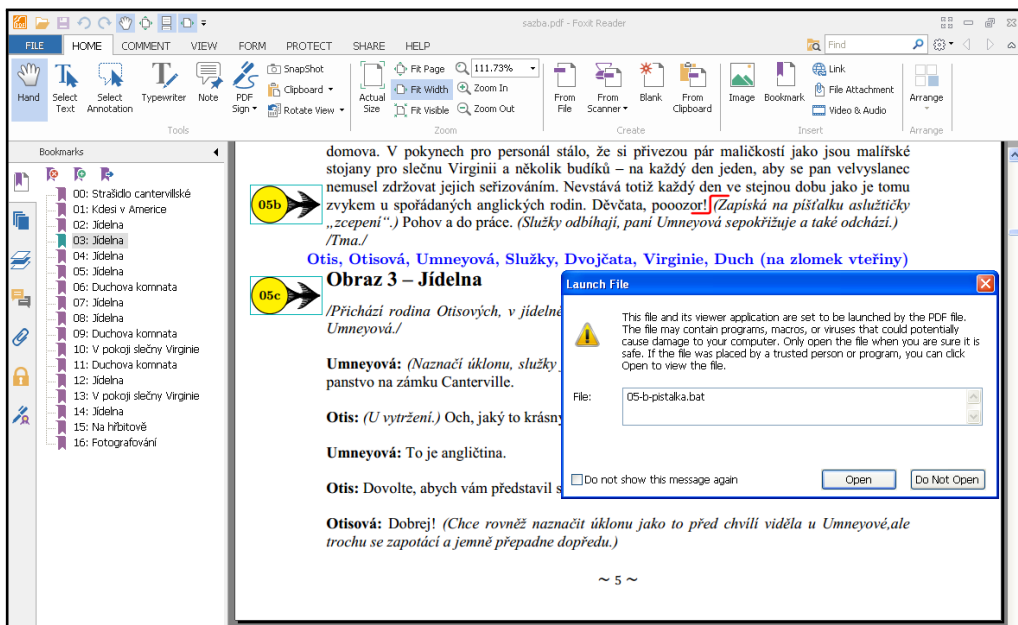
Situaci spuštění systémového prostředí (`\write18`) z T_EXu a spuštění dávkového souboru nakliknutím odkazu v PDF jsem řešil před mnoha lety, tak mě velmi překvapilo, že mé kódy po letech nefungovaly (ani přes nového pomocníka balíček `bashful`). Nepomohlo ani přímé přiložení dávkového souboru do PDF souboru balíčky `embedfile`, `attachfile` či `attachfile2`. Byl jsem značně rozladěn a hledal jsem přijatelné řešení nekonečné hodiny.

Dle nepodložených informací byly na začátku roku 2013 masivní útoky přes formát PDF a tak se bezpečnost PDF prohlížečů výrazně zpřísnila. Vyzkoušel jsem nástroje, které se mi jen dostaly do ruky: Adobe Reader (`acroread`), Xpdf, Evince, Okular, ePDFView, `qpdfview`, nedejbožh nějaké podpůrné JavaScript nástroje typu `pdf.js`. Pokaždé se mi vrátilo hlášení, že spuštění není možné.

Nabízelo se jedno z posledních řešení: přejít na starší prohlížeč PDF, kde by bezpečnost měla být nižší. Ale k tomuto zoufalému kroku jsem nepřistoupil.

Před několika lety jsem řešil přímou editaci PDF souborů a byla to třída nástrojů sama o sobě. Ty jsem začal zkoušet a narazil jsem na staronový FoxIt Reader. V tomto jediném programu a jen pod Microsoft Windows se mi hypertextovým nakliknutím podařilo dávkové soubory spustit.

Humorně dodávám, že na internetových fórech je program kritizován, že něco takového umožňuje, nu, mně to zachránilo divadelní krk.



Obrázek 2. Nabídnuté řešení přes FoxIt Reader.

<http://www.foxitsoftware.com/>

Při každém prvním spuštění prohlížeč nabízí další nedotazování se. To si stačí zaškrtnout a potvrdit. Byl to ideální stav. Omezením mi bylo, že PDF musí být v adresáři s dávkovými soubory (jak si spouští další dávkové soubory), ale to byl operativní zásah přesunutí jednoho souboru.

Klíčová změna na \TeX ové úrovni byla upravit příkaz `\href` o `run` na:

```
\href{run:davka.bat}{Klikni!}
\href{run:sh davka.sh}{Klikni!}
```

Dá se použít i cesta mimo `hyperref` balíček, pro starší verze pdf \TeX u je řešení zmíněno zde (`\pdfannotlink`), případně pro novější verze zde (`\pdfannot`).

```
\def\url#1{\hbox{\pdfannot height 10pt depth 3pt
  {/Border[0 0 0]/Subtype/Link/A <<
  /Type/Action/S/Launch /F (#1)>>}\relax
  \color{blue}#{1}\color{black}}}
\url{d.bat}
```

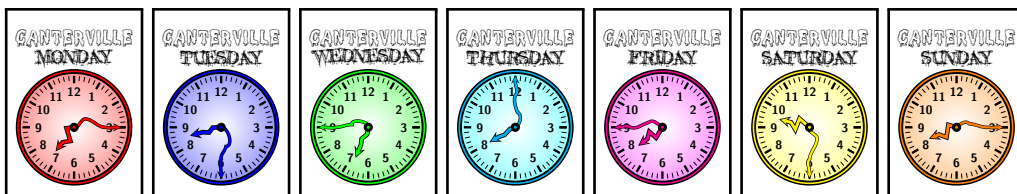
Řešení přes balíček `hyperref` je pro nás dostatečné, jádro je stejné a problém byl na úrovni prohlížeče, nikoliv obsahu PDF. Jak to tedy vypadá, když vše funguje? Náhled pod pokličku divadelního ochotníka je ve spodní části na předchozí straně.

Po termínu odevzdání článku, během pročitání dokumentace nového balíčku `ocgx` (z 2. pol. 2012), jsem otestoval i PDF-XChange Viewer, i zde bylo dosaženo úspěchu! Program Evince pod Windows by asi vyžadoval absolutní cestu k souborům (to si nelze dovolit), neb se automaticky přepíná do domovského adresáře.

Závěrem další vzorky

Zajímavá řešená situace se objevuje u kulis. Například řešíme sazbu sedmi ručičkových hodin s nastavitelným časem a texty (papír versus skutečné hodiny). Možná budou potřeba vlastní pohlednice (ještě řešíme její přední stranu). Pro děti do dílniček jsem řešil sazbu tangramů (viz samostatný článek).

Čas ukáže, co vše bude ještě potřeba, a na co se \TeX ista, spíš bych měl napsat $\text{Ti}\kern{0.2em}\text{Z}$ ista, bude hodit.



Kontaktní adresa

Ing. Pavel Stríž, Ph.D., Nakl. Martin Stríž, U Škol 940, 685 01, Bučovice, Česká republika,
E-mailová adresa: pavel@striz.cz

ABSTRAKTY NERECENZOVANÝCH UKÁŽOK A PREZENTÁCIÍ

Podujatie *Otvorený softvér vo vzdelávaní, výskume a v IT riešeniach* nie je len konferencia v klasickom, akademickom ponímaní, ale aj stretnutie ľudí, ktorých spája záujem o otvorený softvér. Popri recenzovaných príspevkoch bolo na podujatí prezentovaných mnoho ďalších nerecenzovaných príspevkov a ukážok využitia otvoreného softvéru a otvorených informačných technológií. Stručné abstrakty týchto ukážok a prezentácií sú uvedené v závere tejto časti. Súčasťou konferencie sú rôzne sprievodné akcie zamerané nielen na softvér, ale aj na hardvér. Ďalšou neoddeliteľnou súčasťou konferencie sa stávajú odborné semináre pre učiteľov základných a stredných škôl. Tohto roku má veľmi silné zastúpenie na konferencii Sekcia

OpenGIS — Otvorený hardvér, softvér, dáta a ľudia v geografických informačných systémoch.

OpenGIS sa stala už tradičnou sekciou konferencie OSSConf, ktorá sa v rámci vednej disciplíny geoinformatika venuje najmä geografickým informačným systémom (GIS), prezentuje príklady použitia otvoreného hardvéru a softvéru pre zber, spracovanie, publikovanie priestorových (geografických) informácií a údajov (databáz).

Na základe vyhodnotenia minuloročnej hromadnej pripomienky k novele Zákona o geodézii a kartografii č. 215/1995 Z.z. sme zistili, že pracovníci škôl a vedecko-výskumných pracovísk, zástupcovia profesijných združení či občianskych iniciatív sa stávajú významnými používateľmi priestorových údajov a žiadajú štát o bezplatné poskytovanie takýchto podkladov.

Podľa nás v geokomunitách, najmä v akademickom a neziskovom sektore, existuje najväčší potenciál pre tvorbu nadstavbových geografických vrstiev a analýz nad štátnymi referenčnými geodátami. Verejné inštitúcie za finančné prostriedky daňových poplatníkov vytvárajú, no čoraz viac obstarávajú takéto údaje a mapy. Verejnosti však nie sú poskytované bezplatne a častokrát sa o ich existencii a kvalite (metaúdaje) občania dozvedajú iba sporadicky. Ak sa konečne niečo podarí uverejniť, býva takáto geografická informácia už stará a nepoužiteľná. Nová aktualizácia štátom garantovaných údajov je finančne a časovo náročná záležitosť.

Pred rozporovým konaním s Úradom geodézie, kartografie a katastra SR k hromadnej pripomienke sme si položili aj nasledovné dve otázky: Existujú u nás doma alebo vo svete konkrétne aplikácie z geokomunit, ktoré vznikli z dostupných/uvoľnených štátnych geodát? Keby teraz naša verejná správa poskytla bezplatne mapy, geodáta resp. tieto podklady cez mapové služby, bol by stále na strane geokomunit záujem ich využiť? Odpovede sme vtedy nevedeli zistiť.

Tohtoročná sekcia preto okrem odborných prednášok o otvorenom softvéri prináša aj prezentácie zástupcov mapovacích komunit (sú to ľudia - dobrovoľníci, ktorí sú súčasťou crowdsourcingu), otvorenú diskusiu o mapových službách, potrebe topografických a katastrálnych OpenData, vzdelávaní učiteľov v oblasti OpenGIS.

Žiaden z prihlásených príspevkov tejto sekcie nebol recenzovaný, a preto sa v závere zborníka nachádzajú iba ich abstrakty.

Budaj Martin: Mapové znaky v jazyku MetaPost.

Čupka Matúš: Aktívne občianstvo v mapách.

Kisztner Juraj: Monitorovacie systémy v spojení s GIS.

Klačko Michal: TrashOut.

Ofúkaný Miloslav: Od hromadnej pripomienky ku QGIS-u.

Páleník Michal: Freemap Slovakia a OpenStreetMap v rokoch 2012 a 2013.

Piroh Jaroslav: Využitie platformy Google Earth pre propagáciu regiónov.

Růžička Jan, Dubrovský Vojtěch: Podpora hodnocení biocenter.

Růžičková Kateřina: Využití GRASS GIS ve výuce.

Tomašovič Peter: Blender, 3D modelovanie geografických objektov.

Tesař Martin: Vyhledávač tras pro mtbmap.cz.

Miloslav Ofúkaný

MAPOVÉ ZNAKY V JAZYKU METAPOST

Budaj, Martin, (SK)

Štruktúrovaný popis mapových znakov používateľmi GIS (či už pomocou značkovacieho jazyka alebo nastavením atribútov v používateľskom rozhraní aplikácie) sa stal štandardným prostriedkom na definovanie vizuálnej podoby vlastných mapových znakov pomocou ich skladania z jednoduchších grafických prvkov. Príspevok predstavuje alternatívny prístup, v ktorom je grafická podoba znaku popísaná používateľom v jazyku MetaPost. MetaPost je programovací jazyk primárne určený na popis a tvorbu vektorovej grafiky. Tento prístup umožňuje napríklad tvorbu komplexných čiarových znakov, ktorých grafická podoba je prispôbena tvaru alebo atribútom čiary (napríklad čiarový znak s premenlivou šírkou na znázornenie vodného toku); návrh znakov s prvkami náhodnosti (na oživenie máp zvyčajne tvorených z dokonale pravidelných znakov); tvorbu graficky komplikovaných znakov (vrátane ad-hoc bodových aj čiarových kartodiagramov) a najmä správne zobrazenie často prehliadaných detailov mapových znakov (častým prehréškom býva napríklad nedokonalé napojenie grafických elementov, čiastočné orezanie vzorky vyplňajúcej plochu a pod.) vďaka možnosti použitia výpočtov analytickej geometrie v jazyku MetaPost. Predstavená je tiež knižnica Symlib, ktorá uľahčuje integráciu jazyka MetaPost do iných GIS.

MONITOROVACIE SYSTÉMY V SPOJENÍ S GIS

Kisztner, Juraj, (SK)

Projekt spracováva problematiku spojenia geografických informačných systémov a monitorovacieho systému Zabbix. V práci je vytvorená aplikácia, ktorá pridáva nové komponenty do monitorovacieho systému Zabbix na prácu s priestorovou zložkou monitorovaných zariadení. Ukazuje reverzné spracovanie databázy a web front-endu Zabbix-u a prepojenie výsledkov monitorovania s priestorovou databázou PostgreSQL/PostGIS. Následne sa práca zaoberá vytvorením pilotnej mapovej aplikácie, ktorá slúži na zobrazenie monitorovaných zariadení spolu s priestorovou lokalizáciou.

AKTÍVNE OBČIANSTVO V MAPÁCH

Čupka, Matúš, (SK)

Na Slovensku sa v poslednej dobe objavila nová vlna občianskeho aktivizmu. Tá sa týka rozvoja zelenej dopravy, záchranu kultúrnych pamiatok, starostlivosti

o životné prostredie a podobne. Všetky tieto aktivity potrebujú pre svoje efektívne fungovanie rôzne mapové služby, ktoré slúžia na zaznamenávanie problémov, nedostatkov, riešení a ďalších informácií dôležitých pre koordináciu aktivít. Nie každý nástroj je však na to vhodný a nie každý je zdarma. Prostredníctvom tohto príspevku prezentujeme nástroje na mapovanie či už voľne dostupné, alebo financované občianskym či súkromným sektorom, ktoré pri mapovaní potrieb pomáhajú. Portály ako OdkazPreStarostu, TrashOut alebo Mapujeme získali svojich priaznivcov. Dominuje však využívanie bezplatných Google Maps. Súčasťou príspevku bude tiež praktická ukážka mapovania aktivít neformálnym združením Zelená hliadka, ktoré sa aktivizuje najmä v Bratislave.

TRASHOUT

Klačko, Michal, (SK)

Projekt TrashOut sa zaoberá lokalizáciou nelegálnych skládok cez mobilné zariadenia (Android, iPhone, Windows Phone). Dáta sú ukladané v cloude a verejne dostupné na webe trashout.me. Proces lokalizácie je jednoduchý: pozostáva z troch základných krokov. Lokalizácia skládok je zabezpečená na základe GPS súradníc prijatých z mobilného zariadenia. Dáta sú následne štatisticky spracované podľa administratívnych oblastí a reprezentované na webe. Celý proces funguje za využitia Google Maps a Google geocoding API.

OD HROMADNEJ PRIPOMIENKY KU QGIS-U

Ofúkaný, Miloslav, (SK)

Úrad geodézie, kartografie a katastra Slovenskej republiky (ďalej úrad) predložil 23. októbra 2012 na medzirezortné pripomienkové konanie novelu Zákona o geodézii a kartografii č. 215/1995 Z.z. (ďalej zákon). Spoločnosť pre otvorené informačné technológie, Občianske združenie Freemap Slovakia a geokomunitný portál GEOINFORMATIKA.sk predložili 12. novembra spoločnú hromadnú pripomienku k tejto novele zákona. Chceli sme podporiť efektívnejší a jednoduchší prístup k digitálnemu obsahu verejnej správy a docieľiť to, aby sa čo najviac používali štátom garantované referenčné geografické dáta (otvoril sa k nim prístup). Hromadná pripomienka bola doručená úradu 13. novembra so zoznamom 896 podporovateľov, za čo nesmierne ďakujeme. Vďaka aktivite týchto ľudí za 2 dni získali naše stanoviská status zásadnej pripomienky, spôsobili publicitu celej témy a dodali nášmu pôsobeniu legitimitu. Začiatkom decembra 2012 sa uskutočnilo na úrade rozporové konanie ku všetkým bodom hromadnej pripomienky.

Zásadná podpora pri rokovaní bola zo strany Splnomocnenca vlády SR pre rozvoj občianskej spoločnosti Mgr. Filipa Vagača, ktorý zastrešuje iniciatívu OpenData v štátnej správe. Teší nás ústretový prístup úradu, keď v rámci rozporového konania bol poskytnutý priestor na vysvetlenie a argumentáciu ku všetkým našim pripomienkam. V troch hlavných cieľoch hromadnej pripomienky (bezodplatné prístupnenie údajov, strojovo spracovateľné formáty údajov, uvoľnenie pravidiel používania údajov) sa na rozporovom konaní podarilo dosiahnuť významný pokrok, a preto ho považujeme za veľmi úspešné. Napriek tomu, že úrad následne navrhol celú novelu zrušiť, dosiahnuté závery boli obojstranne akceptované a budú základom pre naše budúce rokovania pri príprave nového znenia zákona. Aby sme v budúcnosti mohli ďalej zlepšovať prácu s komunitou, potrebujeme poznať napr. jej sektorovú štruktúru podľa geografického rozmiestnenia. Pre vizualizáciu v mape sme použili databázu podporovateľov hromadnej pripomienky a výsledky sme anonymne spracovali v prostredí opensoure softvéru Quantum GIS (QGIS).

FREEMAP SLOVAKIA A OPENSTREETMAP V ROKOCH 2012 A 2013

Páleník, Michal, (SK)

Úvod prednášky sa venuje predstaveniu projektov Openstreetmap (OSM), Freemap, vrátane dôvodov ich vzniku a hlavne licencie pod ktorou existujú. Predstavíme si aj novinky, ktoré sa vo svete OSM udiali za posledný rok. Popri obligatórnom náraste prispievateľov a dát, je to najmä ešte stále aktuálna zmena licencie, konferencie a mapping party, novinky v editoroch (JOSM, iD), novinky na freemap.sk aj na oma.sk, spolupráca so školami, tlačaná cyklomapa Bratislavy ako i progressbarové mapovacie aktivity. Večerná časť sa venuje mapovaniu používajúc JOSM.

MAPOVANIE V PRAXI: OPENSTREETMAP

Páleník, Michal, (SK)

Cieľom workshopu je priblížiť editovanie OpenStreetMap v programe JOSM. Predstavíme si základné elementy databázy OSM, naučíme sa editovať body (napr. reštaurácia), cesty (napr. rieky a ulice) ako i relácie (napr. turistické značky). Odporúčame prácu na vlastnom notebooku.

VYUŽITIE PLATFORMY GOOGLE EARTH PRE PROPAGÁCIU REGIÓNOV

Piroh, Jaroslav, (SK)

Cielom príspevku je ukázať možnosti využitia voľne dostupných nástrojov na propagáciu regiónov. Vo vystúpení bude naživo prezentovaná aplikácia „Štrbské Pleso na Google Earth“, ktorá je vystavaná na platforme Google Earth API a predstavuje moderný spôsob integrácie rôznych dátových zdrojov v interaktívnej mape. Aplikácia využíva populárne mapové okno Google Earth, ktoré je doplnené o rad nástrojov. Pomocou nich si môže užívateľ zapínať a vypínať témy typické pre konkrétny región a vyberať si jednotlivé objekty z pripravených ponúk. Po vykonaní výberu aplikácia automaticky zameria a priblíži vybraný objekt. Pre objekty, ktoré sú uvedené v ponukách, boli vyhotovené realistické 3D modely, čím sa dosiahol efekt virtuálnej reality. Modely objektov si môže užívateľ aplikácie interaktívne prezeráť a voľne sa pohybovať medzi nimi. K týmto objektom je prilinkovaný rad ďalších verejne dostupných informácií, ktoré sa otvárajú po kliknutí na ikony reprezentujúce wikipédiu, panorámy, cestovné poriadky, lokálne informácie, živé kamery a ďalšie. Druhá časť vystúpenia bude zameraná na tvorbu niektorých obsahových komponentov pre túto aplikáciu (3D modely objektov a panoramatická fotografia).

PODPORA HODNOCENÍ BIOCENTER

Růžička, Jan, (CZ), Dubrovský Vojtěch (CZ)

Projekt sa zabýval uľahčením procesu mapovania biocenter s využitím webových GIS technológií. Na základe stanovených cieľov bola vytvorená webová mapová aplikácia, ktorá umožňuje zaznamenávať prostorové údaje. Nad uloženými údajmi sú prováděny automatizované výpočty sloužící k vytěžení dalších informací. V mapové aplikaci je možné prostorové a popisné údaje prohlížet i vkládat. Pro správu mapovaných území a uživatelských účtů slouží přehledné webové rozhraní.

VYUŽITÍ GRASS GIS VE VÝUCE

Růžičková, Kateřina, (CZ)

GRASS GIS je ve výuce na Institutu geoinformatiky VŠB – TU Ostrava využíván již dlouhá léta. V rámci předmětu Modelování a simulace v geovědách jsou s jeho pomocí prezentovány rozdíly mezi deterministickými a stochastickými modely. GRASS se zde používá s podporou jednoduchého skriptování v Bash tak,

aby bylo možno simulovat také dynamiku některých modelovaných procesů. Je tak možné studentům přiblížit například princip fungování celulárních automatů. Dále jsou zpracovávány aplikačně zaměřené úlohy pro modelování vodní eroze, modelování stoku vody a modelování šíření požárů. I když se nejedná o program, na kterém by byla primárně postavena výuka geoinformatiky, je pro daný účel snáze použitelný než mnohé komerční programy.

BLENDER, 3D MODELOVANIE GEOGRAFICKÝCH OBJEKTŮV

Tomašovič, Peter, (SK)

Blender je v súčasnosti unikátnym nástrojom zameraným nielen na 3D modelovanie, ale obsahuje aj nástroje na textúrovanie, renderovanie, úpravu a „post-processing“ animácií a obrázkov, nástroje na tvorbu špeciálnych efektov, 3d tlač, herný engine a tvorbu hier pre zariadenia s OS Android. Cieľom prezentácie je ukázať možnosti modelovania geografických objektov využitím programu Blender. Pri modelovaní budov budú využité základné princípy práce v 3D prostredí – od navigácie v 3D prostredí až po tvorbu jednoduchých budov využitím základných editačných funkcií. Pri tvorbe prírodných objektov s veľkou premenlivosťou budú využité doplnky programu Blender, ktoré boli vytvorené komunitou a postupne boli dopracované a implementované v oficiálnom vydaní programu. Na modelovanie terénu bude použitý doplnok ANT Landscape a na modelovanie stromov bude použitý doplnok „Sapling“.

DEBIAN A SLOVENČINA

Fedorik, Slavko, (SK)

Distribúcia Debian Linux je v linuxovom svete stálicou už od roku 1993 a za ten čas sa stala základom pre mnoho viac alebo menej úspešných klonov, ako napríklad Ubuntu, či Linux Mint. A hoci Debian nenájdete často na strojoch bežných používateľov, práve tieto klony z neho robia najpoužívanejšiu distribúciu aj medzi bežnými používateľmi. Distribúcia Debian má ja malý prekladateľský tím, ktorý sa stará o to, aby bola dostupná v slovenčine napríklad webová stránka, ale v neposlednom rade aj popisy balíkov a ich konfiguračných dialógov, ktoré sú využívané aj v spomínaných klonoch, a tak poskytujú našincovi zrozumiteľné prostredie. Vo svojom príspevku chcem priblížiť možnosti spolupráce na prekladoch pre distribúciu Debian, a tým prizvať k spolupráci ďalšie osoby, najmä študentov.

VYHLEDÁVAČ TRAS PRO MTBMAP.CZ

Tesař, Martin, (CZ)

Cílem této práce je implementovat vyhledávač tras pro jízdu na horském kole a turistiku. Jako datové podklady jsou využita rozsáhlá geografická data projektu OpenStreetMap. Vyhledávač cílovým uživatelům umožňuje libovolně zadávat své preference a požadavky a optimalizuje tak výsledky vyhledávání každému na míru. Součástí práce je zobrazení interaktivních mapových vrstev, které zpřístupňují doplňkové informace o jednotlivých geografických prvcích. Výsledky jsou začleněny na webu mtbmap.cz.

REVERSE ENGINEERING: OVLÁDAČ DO JADRA LINUX PRE NEZNÁME ZARIADENIE

Rintel, Ľuboško, (SK)

Podpora hardvéru v jadre Linux sa v posledných rokoch rozšírila tak výrazne, že je širšia než u akéhokoľvek iného operačného systému. Spolupracuje s ním drtivá väčšina hardvéru, od bežného vybavenia osobného počítača, cez spotrebnú elektroniku až po špecializované vedecké prístroje. Je obtiažne nájsť zariadenie, ktoré nie je ním aspoň čiastočne podporované. Príspevok na základe skúsenosti s digitizérom televízneho signálu popisuje, ako pridať podporu pre zariadenie, ktorému chýba a neexistuje pre neho žiadna dokumentácia. Z praktického hľadiska preberá architektúru zbernice USB, analýzu (reverse-engineering) protokolu, nástroje, rozhrania jadra, či kódovanie signálu. Nepredpokladá znalosť problematiky. Technického poslucháča prevedie praktickými problémami, neprogramátorovi poskytne vhľad do použitých technológií.

NašeObce.sk – PORTÁL, KTORÝ JE O SLOVENSKU PRE SLOVÁKOV

Proksa, Ondrej, (SK)

Internetový portál NašeObce.sk unikátnym spôsobom poskytuje štatistiky o rozlohe, hustote, nadmorskej výške, prvej historickej zmienke, obyvateľoch, organizáciách v samosprávnych jednotkách Slovenska a efektívne ich prenáša do mapy. Cieľom projektu NašeObce.sk je zobrazovať všetky verejne dostupné informácie o mestách, obciach, okresoch, krajoch a organizáciách na Slovensku. Prostredníctvom tabuliek, grafov a máp sa zobrazujú štatistiky o území, čím sa vytvára komplexný nástroj na porovnanie a priestorovú analýzu. NašeObce poskytujú

aj spoluprácu s obcami a samosprávami pomocou jednoduchých bannerov, ktoré si obce umiestňujú na svoje oficiálne internetové stránky. Tieto bannery priamo odkazujú na ich obce na portále NašeObce.sk.

NOVINKY VE FEDOŘE 19

Eischmann, Jiří, (CZ)

Fedora 19 vychází na začátku července a opět přináší celou řadu novinek nejen v oblasti desktopu, ale především v oblasti serveru. Jak je známo z Fedory vychází Red Hat Enterprise Linux, takže je dobré novinky ve Fedoře sledovat, protože mnoho z nich je budoucností enterprise nasazení. Tento příspěvek by měl poskytnout přehled k tomu, co Fedora 19 přináší a jaké jsou plány do budoucna.

INFORMACE O ANTIPLAGIÁTORSKÉM PROJEKTU

Rybička, Jiří, (CZ)

V letech 2010–2013 se Mendelova univerzita účastnila mezinárodního projektu Impact of Policies for Plagiarism on Higher Education Across Europe (IPPHEAE, číslo 510321-LLP-1-2010-1-UK-ERASMUS-EMHE), jehož základním cílem bylo podchytení plagiátorství, jeho pramenů a jeho řešení na univerzitách v Evropě. V rámci projektu proběhla závěrečná mezinárodní konference v Brně ve dnech 12.–13. června 2013, na níž byly prezentovány jak výsledky výzkumu, tak i řada dalších příspěvků přesahujících i evropský region.

BITCOIN – AKO A PREČO VLASTNE FUNGUJE?

Rusnák, Pavol, (SK)

Táto prednáška priblíži virtuálnu menu Bitcoin a hlavne jej kryptografický aparát, ktorý celý komplexný aparát poháňa. Je určená hlavne pre tých, ktorí už o Bitcoine niekedy počuli, ale doteraz nenašli odvahu pátrať po technických detailoch „pod kapotou“.

PROJEKTY CZ.NIC

Řdurechová, Katarína, (SK)

Predstavenie projektov CZ.NIC, hlavne vývojového a výskumného pracoviska – CZ.NIC Labs.

KNOT DNS A BEZPEČNOST

Vavruša, Marek, (CZ)

Knot DNS je autoritativní DNS server vyvíjený v laboratořích CZ.NIC. V příspěvku se podíváme nejen na zajímavosti a novinky z jeho vývoje, ale i v poslední době často diskutovanou bezpečnost a nové způsoby obrany před amplifikačními útoky.

DATOVKA: ANDROID APLIKACE PRO OBSLUHU DATOVÝCH SCHRÁNEK

Strbačka, Martin, (CZ)

Představení dalšího z projektů CZ.NIC. V přednášce bude nahlédnuto pod pokličku vývoje Datovky (play.google.com/store/apps/details?id=cz.nic.datovka), aplikace pro obsluhu datových schránek (mojedatovaschranka.cz) pro mobilní zařízení s operačním systémem Android.

SKÚSENOSTI S NASADENÍM SYSTÉMU NA SPRÁVU VERZIÍ DO PROCESU ODOVZDÁVANIA A VYHODNOCOVANIA ŠTUDENTSKÝCH PROGRAMÁTORSKÝCH ZADANÍ

Biñas, Miroslav, (SK)

Proces odovzdávania a vyhodnocovania študentských zadaní je štandardnou súčasťou vzdelávania. Zo stavu, kedy študenti svoje projekty odovzdávali v tlačenej podobe ako zdrojové kódy alebo ich nosili na CD/DVD nosičoch, sme v stave, kedy ich študenti odovzdávajú elektronicky. E-mailové schránky na takýto spôsob odovzdávania nie sú vždy úplne ideálne, ale vedia byť postačujúcim riešením. Mnohé e-learningové systémy poskytujú pre študentov možnosť odovzdať svoje zadanie a učiteľom zasa prostredie na jeho ohodnotenie, ktoré študenti následne môžu vidieť. Tento príspevok sa venuje skúsenostiam s nasadením systému na správu verzií do procesu odovzdávania a vyhodnocovania študentských programátorských zadaní na Technickej univerzite v Košiciach. Skúsenosti boli získané počas trvania kurzu Programovanie v druhom semestri bakalárskeho štúdia.

OTVORENÝ SOFTVÉR VO VZDELÁVANÍ, VÝSKUME A V IT RIEŠENIACH
Zborník príspevkov medzinárodnej konferencie OSSConf 2013,
konanej 2.–4. júla 2013 v Žiline

Prvé vydanie 2013

Elektronická sadzba programom pdfL^AT_EX Rudolf Blaško

Vydal SOIT, Bratislava, Žilinská univerzita v Žiline

Tlač C–PRESS, s. r. o., Košice

Rozsah 150 strán

Náklad 100 ks

ISBN 978-80-970457-3-9

ISBN 978-80-970457-3-9



9 788097 045739