

[illegible]

ZPRÁVODAJ

ení uživatelů T_EXu Zpravodaj Československého sdružení uživatelů T_EXu Zpravodaj Československého sdružení uživatelů T_EXu Zpravodaj Československého sdružení uživatelů

Československého sdružení uživatelů T_EXu[illegible]

3-4

2017

OBSAH

Petr Sojka: Úvodník	117
Petr Sojka, Vít Novotný: T _E X na školách? Samozřejmě ano! Příklad užití T _E Xu na Fakultě informatiky Masarykovy univerzity	118
Hans Hagen: LUA in METAPOST	138
Peter Wilson: Mělo by to fungovat VI – Odstavce	155

Zpravodaj československého sdružení uživatelů T_EXu je vydáván v tištěné podobě a distribuován zdarma členům sdružení. Po uplynutí dvanácti měsíců od tištěného vydání je poskytován v elektronické podobě (PDF) ve veřejně přístupném archívu dostupném přes <http://www.cstug.cz/>.

Své příspěvky do Zpravodaje můžete zasílat v elektronické podobě, nejlépe jako jeden archivní soubor (.zip, .arj, .tar.gz). Postupujte podle instrukcí, které najdete na stránce <http://bulletin.cstug.cz/>. Nezapomeňte přiložit všechny soubory, které dokument načítá (s výjimkou standardních součástí T_EX Live), zejména v případě, kdy vás nelze kontaktovat e-mailem.

ISSN 1211-6661 (tištěná verze)

ISSN 1213-8185 (online verze)

Milé čtenářky a čtenáři, příznivci kvalitní typografie sázecím systémem \TeX , s radostí píší úvodník dalšího čísla Zpravodaje, který uzavře podle v minulém čísle avizovaného algoritmu vydávání ročníku 2017. Kromě potvrzení důležitosti snah o dodržení pravidelnosti vydávání Zpravodaje prosincová valná hromada schválila zrušení roční lhůty pro volné vystavení čísel Zpravodaje na webu sdružení. Tím bude Zpravodaj dostupný rychleji větší čtenářské obci. Doufáme tím v další zvýšení kreditu publikování ve Zpravodaji a ζ TUGu jako sdružení. Také apelujeme na stávající členy, aby ve svém okolí šířili povědomí o sdružení a pomáhali tak dobré věci, a naplnění cíle sdružení a tohoto Zpravodaje.

ζ TUG podporuje kvalitní typografii v ČR i SR přes čtvrt století, \TeX nedávno oslavil čtyřicátiny, a jeho autor osmdesátiny. Knuth je oslavil v kruhu svých přátel ve švédském Piteå, kde kromě odborného sympozia zazněla premiéra Knuthova multimediálního díla *Fantasia Apocalyptică*. Detaily akce lze nalézt na <http://knuth80.elfbrink.se/>. Knuth tím rozšířil řadu osob, kteří po větší část života používali \TeX .

V tomto čísle najdete tři články. V prvním líčím své přesvědčení, že \TeX měl a má na školách typu Fakulta informatiky MU své místo, a nemalým způsobem přispívá ke kvalitě produkovaných dokumentů na školách. Jen přínos vzniku pdf \TeX u na FI MU komunitě nelze jednoduše vyčíslit.

Že imperativní programování má smysl nejen v \TeX u, ale i v METAPOSTu, přesvědčuje ve svém článku duchovní otec \LaTeX u Hans Hagen. Možnosti programovatelné grafiky v METAPOSTu tak dostávají nové možnosti a efektivitu.

Číslo uzavírá přeložený článek Petera Wilsona věnovaný možnostem formátování odstavců. Zde si doufám najde použitelné triky každý uživatel \TeX u.

Všem čtenářům přeji pěkné chvíle nad dvojčíslem Zpravodaje. Pro bezproblémovou sazbu v roce 2018 doporučuji opět volit \TeX . Dle osobní zkušenosti je to dobrá, spolehlivá volba kvalitního sázecího systému, nejen na školách. Který software každodenního užití vydržel čtyři dekády? Přeji mu i jeho autorovi aspoň dvojnásobek.

Summary: Introduction

Editorial informs about ζ TUG's new publishing policy and comments on this issue's articles. Go forth and participate in ζ TUG to make the future of \TeX & Friends a reality! *You can!*

*Masarykova univerzita, Fakulta informatiky, Botanická 68a, 602 00 Brno
sojka@fi.muni.cz*

T_EX na školách? Samozřejmě ano! Příklad užití T_EXu na Fakultě informatiky Masarykovy univerzity

PETR SOJKA, VÍT NOVOTNÝ

Studenti Masarykovy univerzity (MU) užívají T_EX velmi široce – píší závěrečné práce, eseje, odborné články. Stejně tak je T_EX využíván při výuce elektronického publikování a literárního programování, pro přípravu vědeckých článků, písemných testů a učebních materiálů a ke generování tiskových výstupů a webových stránek z univerzitních databází informačním systémem. T_EX a příbuzné technologie byly na Fakultě informatiky Masarykovy univerzity (FI MU) systematicky podporovány a využívány po více než dvacet let. V tomto článku popisujeme T_EXové projekty, které se nám na FI MU podařilo realizovat. Mezi tyto spadá návrh fakultní vizuální identity, materiály pro výuku elektronického publikování a nástroje pro databázové publikování. Výsledky projektů hodnotíme a rozebíráme možné budoucí využití T_EXu a příbuzných technologií. Na základě výsledků statistické analýzy využití T_EXu na MU potom argumentujeme, proč je odpověď na otázku v názvu článku kladná.

Why not just hope that in the flow of getting words on a medium we play our humble role and hope we're not forgotten but remembered as inspiration.
(Hans Hagen, [7, p. 32])

Úvod – základní předpoklady

T_EX se zrodil na katedře informatiky Stanfordské univerzity jako nástroj, který byl vyvinut především pro osobní potřebu svého autora. Dává smysl T_EX vyučovat na školách? Takové otázky byly již v minulosti kladeny a zodpovězeny [22, 4, 19] různým způsobem. Za jakých předpokladů a pro jaké účely by měly být T_EX a přidružené programy na školách využívány? To však záleží na typu školy, na řešených problémech a na koncových uživateli.

- T_EX pro výuku algoritmizace jako programovací (makro)jazyk? Nejspíš ne.
- T_EX jako ukázkový příklad paradigmatu literárního programování? Možná.
- T_EX jako nízkoúrovňový sazební systém? Ve vybraných případech v závislosti na typu školy.
- T_EX ve formátu L^AT_EXu jako standardní formát pro tvorbu vědeckých publikací? Nejspíš ano.
- T_EX jako nástroj pro budování komunit? Proč ne.

V textu se pokusíme shrnout čtvrtstoletí osobních zkušeností s \TeX em na Ústavu výpočetní techniky a FI MU v Brně. Veškeré uvedené poznatky jsou podmíněny dobou, místem, typem školy a dalšími faktory, a nelze je tedy vždy zobecnit. Navzdory tomu věříme, že čtenář shledá naše ohlédnutí za putováním s \TeX em zajímavým.

Historia magistra vitae (Latinské přísloví)

Historie \TeX u na českých školách — jenom potěcha, nebo skutečný přínos?

Na úvod si dovolíme krátce shrnout historická fakta.

80. léta \TeX se objevil v Československu na konci osmdesátých let jako nástroj pro šíření zakázaných publikací samizdatovou komunitou [5]. Za tímto účelem bylo třeba obohatit fonty Computer Modern o českou diakritiku [47].

90. léta Rok po sametové revoluci bylo založeno Československé sdružení uživatelů \TeX u ($\text{\textit{CSTUG}}$). Jelikož většina individuálních a hromadných členů pocházela z akademického světa, staly se střední školy a univerzity přirozenými centry znalostí o \TeX u.

Pro zasazení do patřičného historického kontextu – Hàn Thế Thành (obrázek 1) právě přicestoval za prací ze socialistického Vietnamu, začal se učit česky na české škole a přihlásil se na FI MU. Dále byla konsorciem českých univerzit pronajata první internetová 56 kbps ADSL linka z Lince a soubor `latex.tex` o velikosti 290 kB bylo možné snadno prohledávat a editovat i na PC XT s 640 kB operační pamětí a dvěma 5,25" disketovými mechanikami.

\TeX začal nabírat na popularitě a skupinka nadšenců se rozhodla uspořádat \TeX ovou konferenci v Praze [48]. Tak se zrodil Euro \TeX 92, na který dorazilo



Obrázek 1: Hàn Thế Thành studoval na brněnské FI MU v letech 1991–2001.

přibližně třista účastníků z celého světa. \TeX začal být využíván pro přípravu knih a pro databázové publikování [40].

Vznikla nová československá varianta fontů Computer Modern (`csfonts`). Matematické časopisy začaly hromadně přecházet k \TeXu . *Czechoslovak Mathematical Journal*, *Applications of Mathematics*, pražská *Mathematica Bohemica*, brněnské *Archivum Mathematicum* i bratislavská *Mathematica Slovaca* začaly využívat \TeX jako svůj hlavní sazečský nástroj.

Komunita uživatelů \TeXu začala vzkvétat. Skupiny matematiků začaly sázet recenze pro německý časopis *Zentralblatt Math* a (\LaTeX) ové kurzy se začaly vkrádat do školních kurikul, primárně jako *nástroje* pro sazbu matematiky. Jeden takový kurz byl dokonce vyučován na konferenci TUG 1993 v Astonu ve Velké Británii.

V té době pracoval první autor na Ústavu výpočetní techniky Masarykovy univerzity, kde propagoval užití \TeXu . V univerzitním periodiku *Zpravodaj MU* a ve *Zpravodaji ČS TUG* vyšla série článků o \TeXu . MU se stala kolektivním členem TUGu. \TeX byl aktivně podporován a vznikly upravené verze \TeXu s přímou podporou vstupního kódování ISO Latin-2, které byly zavedeny do sdílených univerzitních instalací \TeXu .

První informatická fakulta v České republice – Fakulta informatiky Masarykovy univerzity v Brně (FI MU) – byla založena roku 1994. Prvním děkanem fakulty se stal Jiří Zlatuška, který byl \TeX ovým příznivcem. Fakultní logo bylo navrženo jako *ligatura* FI na motiv Penroseova trojúhelníku, viz obrázek 2. Motto loga pochází z Myšlenek od Blaise Pascala: „Věčné mlčení těch nekonečných prostorů mě děsí.“

\TeX se stal součástí každodenního života na fakultě. Bylo zapotřebí sázet rozvrhy učeben, osob a studijních skupin. \TeX se pro tento úkol ukázal být ideálním nástrojem (viz obrázek 3). \TeX byl využíván pro sazbu takřka všech databázových výstupů fakultní administrativy [26] včetně telefonních seznamů, sylabů vyučovaných kurzů – jak vidno na obrázku 4 – a diplomů.

V roce 1994 byl zpřístupněn kurz o elektronické přípravě dokumentů. Kurz byl navržen jako mix teorie a praxe [18]. Studenti zjišťují, jak se informace přenáší z autorovy hlavy skrz značkovací jazyk (\LaTeX) do cílového média a odsud do mysli čtenáře, odhalují princip separace obsahu od formy a učí se o specifikách tiskových (PDF) a digitálních (XHTML) výstupních formátů. Jelikož tvorba dokumentů a programování mají mnoho společného, studenti využívají verzovací systémy (SVN, Git) a automatizační nástroje (GNU `make`). V \TeXu si studenti osvojují jak praktické dovednosti, jako je příprava závěrečných prací, tak teoretické znalosti, jako jsou algoritmy řádkového zlomu a generování vzorů dělení slov.

Veškeré snahy byly podřízeny tomu, aby byla fakulta bezpečným pískovištěm pro experimentaci s \TeX ovými hračkami a nástroji. Toto přinášelo obecný užitek a umožňovalo studentům snadno učinit \TeX předmětem svého studia [27]. Pro studenty, jako byl Hàn Thé Thành, představoval \TeX jasnou volbu pro sazbu



Obrázek 2: Logo Fakulty informatiky: ligatura FI jako symbol kvalitní typografie byla implementována v METAFONTu [49]. Opticky škálované znaky v pečeti byly rekurzivně propojeny pomocí METAFONTového mechanismu ligatur.

jejich esejí a závěrečných prací. Hàn Thê Thành si zvolil $\text{T}_{\text{E}}\text{X}$ a tehdy čerstvě navržený PDF formát za téma své diplomové práce. $\text{T}_{\text{E}}\text{X}$ byl ve velké míře využíván pracovníky fakulty pro přípravu akademických textů a většina vědeckých publikací byla rovněž připravena v $\text{T}_{\text{E}}\text{X}$ u. Fakultní technické zprávy využívají vlastní $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ový styl, který nastavuje písmo Palatino od Hermanna Zapfa. Palatino slouží jako primární písmo FI MU.

Pro automatizaci sazby delších textů a databázové publikování byly zapotřebí kvalitní vzory dělení slov. Výsledky výzkumu prvního autora [45, 32] byly zveřejněny mimo jiné na TUGu 1995, kde se první autor setkal s Donaldem Knuthem a pořídil fotografii na obrázku 5. Při této příležitosti byl Don pozván do Brna, aby převzal svůj dvacátý čestný doktorát.

Po příjezdu do Brna uviděl Don na zastávce tramvaje jízdní řády vysázené písmem Computer Modern (viz obrázek 7). Byl potěšený, že ovoce jeho „práce z lásky“ se hojně využívá na opačné straně zeměkoule jak teoreticky, tak prakticky. Toto zmínil i ve své inaugurační přednášce, když se stal prvním čestným doktorem FI MU.

Roku 1996 obhájil Hàn Thê Thành svou diplomovou práci [10]; program zvaný `tex2pdf` [31] byl představený $\text{T}_{\text{E}}\text{X}$ ové komunitě na konferenci TUG 1996 v ruské Dubně. Program, který byl následně přejmenován na `pdftex`, si získal pozornost $\text{T}_{\text{E}}\text{X}$ ové komunity a byla připravena první verze uživatelského manuálu programu [15].

Nová hračka potřebovala ke svému otestování uživatele, kteří by ji využívali při své každodenní práci. Udržovali jsme celofakultní instalace, které byly sdílené

Rozvrh pro skupinu 1MI, 1994–95–léto

	6:50-7:35	7:40-8:25	8:30-9:15	9:20-10:05	10:15-11:00	11:05-11:50	12:00-12:45	12:50-13:35	13:45-14:30	14:35-15:20	15:30-16:15	16:20-17:05	17:10-17:55	18:00-18:45	18:50-19:35
Po	P3 1M6-x Cv. teorie množin Hodicez Leo	P3 1M6-x Cv. lineární algebra Slovák Jan			J 1M6 Angličtina začít. Kat. jazyků			I3 Sem. z funkcionálního progr. Hajn Pavel I2 1M6-m,4,5Vt Sem. z funkcionálního progr. Škarvada Libor			M1 1M6 Lineární algebra Slovák Jan				
Út					IA 1M6-a Seminář z programování Ševečková Michaela		IA 1M6-b Seminář z programování Ševečková Michaela			J 1M6-a Angličtina pokr. Kat. jazyků				P4 1M6 Němčina Kat. jazyků	
St				TV 1M6 Kat. TV				J 1M6-b Angličtina pokr. Kat. jazyků Aula 1M6-m Teorie množin Kač oarek Jiří			I2 1M6 Návrh algoritmů I Ochránová Renata				
Čt															

Všecké změny a opravy hláste na studijní oddělení Ivě Hollanové

rozvrh 1MI (A P3) – 28. únor 1995, 9:57

Obrázek 3: Příklad rozvrhu pro studijní skupinu 1MI na FI MU z roku 1995.

napříč rozdílnými operačními systémy a které využívaly stejné texmf stromy. Navíc jsme udržovali a udržujeme historické instalace $\text{T}_{\text{E}}\text{X}$ Live, které si uživatel může zpřístupnit pomocí mechanismu modulů. O dvacet let později je většina historických verzí $\text{T}_{\text{E}}\text{X}$ Live stále nainstalována a připravena k použití; toto umožňuje uživatelům vrátit se v čase a sázet desítky let staré materiály.

Díky snížení nároků na začínající uživatele $\text{T}_{\text{E}}\text{X}$ u, zpřístupnění potřebných nástrojů a místní uživatelské komunitě se $\text{T}_{\text{E}}\text{X}$ rychle stal primárním systémem pro přípravu dlouhých dokumentů, jako jsou knihy a závěrečné práce. Byla navržena $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ová třída fithesis pro přípravu závěrečných prací. Třída byla nainstalována a zpřístupněna studentům spolu s tištěnou příručkou „Začínáme s $\text{T}_{\text{E}}\text{X}$ em na FI“, kterou studenti obdrželi při zápisu ke studiu.

Fakulta organizovala v roce 1996 konferenci Gödel a v roce 1998 konferenci Mathematical Foundations of Computer Science (MFCS). Všecké konferenční materiály (viz obrázek 9) byly připraveny pomocí $\text{T}_{\text{E}}\text{X}$ u z jedné *textové* databáze. V rámci Semináře o Linuxu a $\text{T}_{\text{E}}\text{X}$ u, který byl organizován především samotnými

Doporučení: Je vhodné mít základy algoritimizace, základní znalosti práce s počítačem v unixovém prostředí (vhodné absolvovat například předmět P004 UNIX) a mít ponětí o formálních jazycích.

Úvod. Vymezení předmětu. Cyklus přípravy a ladění dokumentů. Analogie s vývojem programů. ♦ **Značkování.** Logická vs. vizuální struktura dokumentu. Značkovací jazyky, SGML, XML, HTML. Gramatiky dokumentů, DTD. Validace dokumentů, NSGMLS. ♦ **Design.** Principy knižního designu. Specifika designu na WWW. ♦ **Sazba.** Základy typografie, základní typografické pojmy, míry, terminologie. ♦ Písma, typy formáty písem, způsoby reprezentace a designu písem. Rastrovací algoritmy, techniky redukce tvaru písem. ♦ Pravidla sazby. Mikrotypografie. Specifika sazby českých textů. Korektura, značky. ♦ Sázeční systémy. T_EX jako příklad dávkového sázečního systému. WYSIWYG systémy, DSSSL, XSL. ♦ T_EX. Historie. Princip makrojazyka. Algoritmy řádkového a stránkového zlomu použité v T_EXu. *h_z*-systém. Algoritmus dělení slov, ♦ **Předtisková příprava.** Jazyky pro popis stránek. Postscript. Bézierovy křivky. SPDL. Direct Imaging. Archivace montáž. ♦ **Tisk a distribuce.** Výstupní zařízení. Osvět, tisk a vazba. Portable Document Format, Adobe Acrobat. E_TX₂html. pdfT_EX. Publikace databází. Konverze, aktualizace a údržba dokumentů. ♦ **Závěrečné shrnutí.** Sdílení zkušeností, anketa.

Doporučená literatura:

- Knuth, Donald Ervin. *Digital typography*. Stanford : Center for the Study of Language and Information, 1999. xv, 685 s.
- Beran, Vladimír. *Typografický manuál : učebnice počítačové typografie*. 1. vyd. Náchod : MANUÁL, 1994. přeruš. st.
- Brighurst, Robert. *The elements of typographic style*. Vancouver : Hartley & Marks, 1992. 254 s.

Obrázek 4: Syllabus předmětu Elektronická příprava dokumentů vysázený písmem Minion pomocí pdfT_EXu do katalogu kurzů vyučovaných na FI MU v roce 2014.

studenty, pak vyvinuli linuxoví a T_EXoví nadšenci nejen zajímavý výzkumný program, ale mimo jiné i ilustrace z obrázků 8 a 10.


Informační systém fakulty, rovněž vyvinutý zčásti studenty [26], generoval většinu výstupů skrze virtualizovanou T_EXovou instalaci. Data do katalogu předmětů byla od vyučujících shromážděna pomocí webových formulářů a následně byla validována, převedena do L^AT_EXu a vysázena. DTD schéma použité pro validaci vstupu rozeznávalo speciální entity &T_EX; a &L^AT_EX; ©. Vzory dělení slov byly dále zdokonaleny [33] za účelem minimalizace chyb při automatické sazbě. Studenti byli motivováni k aktivnímu zapojení v T_EXových projektech. Mirka Misáková implementovala po vzoru Gutenberga mechanismus zarovnávání textu do bloku pomocí roztahování znaků v METAFONTu jako součást své diplomové práce [21], Jan Pazdziora studoval algoritmy řádkového a stránkového zlomu [25], a Pavel Janík se věnoval digitálním formátům fontů [16]. Většinu programátorské práce na $\mathcal{N}\mathcal{T}\mathcal{S}$ [50] vykonal v Brně absolvent MU, Karel Skoupý [30].



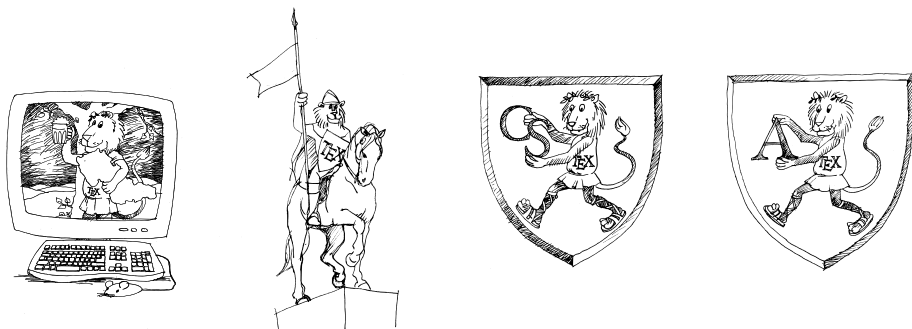
Obrázek 5: Donald Knuth kyne prstem při rozhovoru s Jiřím Zlatuškou během konference TUG 1995 na Floridě; fotografie Petr Sojka.



Obrázek 6: Přednáška Donalda Knutha na Fakultě informatiky Masarykovy Univerzity v Brně roku 1996.

▼ Kohoutovice 1 Pavlovská 2 Talichova (o) 3 Bellova 4 Voříškova 5 Stamicova 6 Glínková 7 Borodínova (o) 8 Libušina třída (o) 9 Libušino údolí (z) 10 Antonína Procházky (z) 12 Pisárky 15 Výstaviště (o) 17 Mendlovo náměstí 19 Tvrdeho (o) 21 Úvoz 23 Komenského náměstí z : zastávka celodenně na znamení o : zastávka od 20 do 5 hodin na znamení		<div>137</div> <div>Odjezdy ze zastávky</div> <div>Kohoutovice</div> 	
		PRACOVNÍ DNY 0 30M 1 30M 2 30M 3 30T 4 00 15 30 42 51 5 05 11 16 22 27 33 38 44 49 55 6 00 06 11 17 22 28 33 39 44 50 55 7 01 06 12 17 23 28 33 40 46 54 8 02 08 14 20 26 32 38 44 50 9 02 14 26 38 50 10 02 14 26 38 50 11 02 14 26 38 50 12 02 14 26 38 50 13 02 14 26 38 50 14 02 08 14 20 26 32 38 44 50 56 15 02 07 13 18 24 29 35 40 46 51 57 16 02 09 15 22 28 35 42 43T 49 57 17 05 06T 13 21 24T 30 37 45 54 18 03 12 14T 21 31 40 49 19 00 12 23 33 45 51T 58 20 10 13T 22 34 46 58 21 10 22 34 47 57T 22 02 10T 22 42 23 09T 13M 30M 51T	
SOBOTA 0 30M 1 30M 2 30M 3 30T 4 00 15 30 45 5 00 15 28 41 54 6 06 18 30 42 54 7 06 18 30 40 50 8 00 08 16 24 32 40 48 56 9 04 12 20 28 36 44 52 10 00 10 20 30 40 50 11 00 10 20 28 36 44 52 12 00 08T 10 20 31 43 54 13 06 08T 17 29 40 52 14 03 15 26 38 49 15 01 12 23 34 45 56 16 06 16 26 36 46 56 17 07 18 29 40 51 18 02 13 24 35 46 58 19 11 23 25T 36 50 20 04 11T 17 30 43 57 21 10 24 37 51 22 04 14 24 43 48T 23 06M 08T 30M 49T		NEDĚLE 0 30M 1 30M 2 30M 3 30T 4 00 15 30 50 5 05 20 35 50 6 05 20 35 50 7 05 20 35 48 8 03 15 27 39 51 9 03 15 27 39 51 10 03 15 27 39 51 11 05 19 31 40 52 12 04 16 28 40 52 13 04 16 28 39 50 14 01 12 23 34 45 59 15 13 22 28 36 45 53 16 02 11 21 30 39 48 57 17 05 11 17 23 31 38 45 51 57 18 03 09 15 23 31 39 47 55 19 03 13 23 27T 36 37T 50 51T 20 04 11T 17 30 43 57 21 10 24 37 51 22 04 14 24 43 48T 23 06M 08T 30M 50T	
M : jede jen na Mendlovo náměstí T : jede do Husovic na Tomkovo náměstí DPmB, Hlinky 151, tel. 4317 1111 Informace o MHD, tel. 4221 0524		Platí od 1.prosince 1995	

Obrázek 7: Jízdní řády brněnské městské hromadné dopravy z doby návštěvy Donalda Knutha v Brně v květnu 1996. Řády využívají písmo Computer Modern.



Obrázek 8: Ikony pro Seminář o Linuxu a T_EXu (SLT '98) připravila Petra Rychlá.

Počátek 21. století Hàn Thê Thành konzultoval další vylepšení pdf_{tex}u [11] s Hermanem Zapfem a provedl sérii mikrotypografických experimentů spolu s Hansem Hagenem, který na MU blokově vyučoval seminář o programování v T_EXu s využitím jazyka LUA a formátu ConT_EXt MkIV. V říjnu roku 2000 dokončil Hàn Thê Thành svou disertační práci [12] a po jedenácti letech studia opustil Brno a vrátil se zpět do Vietnamu, kde finančně zajistil svou rodinu a krátce působil ve Vietnamském akademickém světě [13, 14].

S rostoucím významem elektronických dokumentů rostla i poptávka. V reakci na to vznikly nové učebnice a interaktivní učební materiály [6]. Bylo potřeba vytvářet animace uvnitř PDF dokumentů [34], multiple-choice testy [36] a interaktivní učební materiály v PDF pomocí JavaScriptu [35]. T_EXová notace se těšila takové oblibě, že vyučující matematiky požadovali rozšíření jazyka pro tvorbu online testů o speciální `<math>` element, který by jim umožnil přímo vkládat matematické vzorce v L^AT_EXu. Tato podpora byla zavedena, přičemž vzorečky byly dynamicky sázeny pomocí L^AT_EXu a programu `dvipng` propojených unixovou trubkou. Jako příklad dalších nástrojů týkajících se T_EXu, které byly navrženy a vyvinuty studenty a pracovníky FI, poslouží software pro automatické skenování a hodnocení odpovědních listů vygenerovaných T_EXem [9], rozšířená verze programu `patgen` zvaná `opatgen`, která přidávala podporu přímého zadávání vzorů dělení slov v UTF-8 [2, 1, 39], nebo software pro tvorbu animovaných PDF dokumentů v pdf_{tex}u [8].

Dalším požadavkem byla opětovná využitelnost obsahu připraveného v T_EXu na více výstupních zařízeních. Podařilo se nám ukázat, že pokud jsou obsah a forma naznačovány ve vstupním dokumentu samostatně, lze v T_EXu snadno generovat množství rozdílných výstupů, jmenovitě výstup pro čtení na monitoru, HTML výstup pro zařízení vybavená webovým prohlížečem a (X)HTML/MathML výstup pro zařízení s plnou podporou moderních webových technologií [42] bez



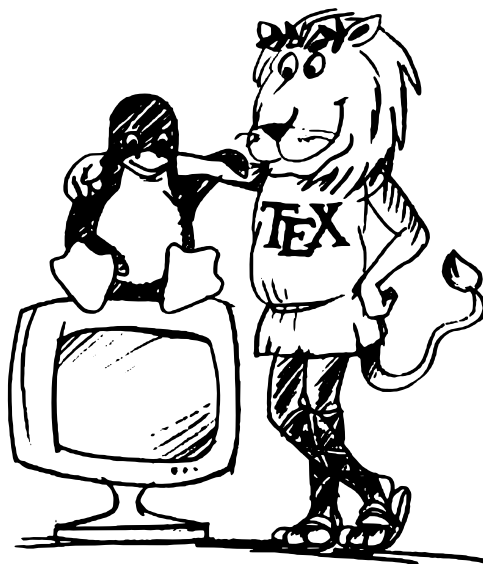
*MFCS/CSL'98 Conference
Welcome Party Invitation
for Mr. Honza Staudek*

*You are wholeheartedly invited to the welcome party
of the MFCS/CSL'98 Conference.*

*The welcome party will take place in the open space
inside the building complex of Faculty of Informat-
ics, Botanická 68a, at 7:30 PM on Sunday, August 23,
1998. Starobrno beer will be served.*

Please, present this invitation card at the entrance.

Obrázek 9: Pozvánka pro účastníky konference MFCS '98 pořádané na FI MU.



Obrázek 10: Logo Semináře o Linuxu a T_EXu (SLT) připravila Petra Rychlá.

použití složitých systémů používaných velkými nakladatelstvími. Náš postup založený na \TeX u využívá většina periodik spolupracujících s matematickou knihovnou DML-CZ [29, 43].

Ve stejné době, kdy \TeX a Knuth nabírali na popularitě, se mnoho firem začalo přesouvat do Brna, které je dnes známé jako Křemíkové údolí střední Evropy. V důsledku toho se brněnské nakladatelství rozhodlo ve spolupráci s absolventem FI MU nechat přeložit knihu *The Art of Computer Programming* (TAOCP) do češtiny a Knuthovy přeložené zdrojové texty vysázet (viz obrázek 11).

Druhé desetiletí 21. století S využitím znalostí sazby v \TeX u se studenti a absolventi FI MU zapojili do množství projektů týkajících se digitálních matematických knihoven, jmenovitě DML-CZ a EuDML. Byl vyvinut postup založený na \TeX u, při kterém je automaticky generována archivní verze dokumentu, která je uložena v rámci digitální knihovny. Časopis *Archivum Mathematicum* vydávaný MU využívá nástroje a postupy vyvinuté pro DML-CZ [44, 38]. Mezi vyvinuté nástroje se řadí efektivní technika komprese PDF dokumentů [41] a algoritmus pro indexaci a vyhledávání v \TeX em připravených matematických dokumentech, který byl vyvinutý v rámci projektu MIAŠ [20] a který byl nasazený v EuDML [46]. Na podnět nevidomých studentů, kteří potřebovali studovat matematiku z učebních materiálů psaných v \TeX u, byla v rámci diplomové práce vyvinuta podpora pro generování výstupu v české variantě Braillova slepeckého písma [17].

Vývoj nástrojů souvisejících s \TeX em byl podpořen programem děkana pro podporu studentských výzkumných a vývojových projektů a nabízen jako téma závěrečných prací. Druhý autor článku pod vedením prvního autora připravil novou verzi \LaTeX ové třídy fithesis [23] pod názvem fithesis3, která přinesla podporu všem devíti fakultám Masarykovy univerzity. Tisíce studentů napříč univerzitou nyní připravují své závěrečné práce v \LaTeX u s možností konzultace problémů skrz diskuzní fórum v univerzitním informačním systému. Studenti jednotlivých fakult se rovněž začínají zapojovat do vývoje třídy; toto vnímáme jako signál, že dochází ke vzniku univerzitních \TeX ových komunit i mimo FI MU.

Další nástroj byl vyvinut v reakci na problémy s prevencí výskytu neslabičných předložek na koncích řádků při sazbě zápisů z akademického senátu zaznamenaných ve značkovacím jazyce Markdown. Druhý autor vyvinul \LaTeX ový balíček `markdown.tex`, který umožňuje zpracovávat dokumenty v jazyce Markdown přímo v \TeX u a mimo jiné řeší i uvedený problém s vazbou neslabičných předložek [24].

FI MU sklídila ovoce své práce v poli elektronické přípravy dokumentů při nedávné změně vizuálního stylu Masarykovy univerzity. Změny při práci s \TeX em byly minimální a díky oddělení obsahu od formy se autorů ve velké míře vůbec nedotkly. Krátce po zveřejnění nového vizuálního stylu se na fakulním GitLab serveru objevily \LaTeX ová třída `muletter` pro přípravu dopisů a šablona pro sazbu posudků závěrečných prací, což rovněž přispělo k hladkému přechodu.

\TeX momentálně slaví čtvrt století podpory a vývoje na MU, během kterého

KLASICKÉ DÍLO V ČEŠTINĚ

AKTUALIZOVÁNO A PŘIZPŮSOBENO SOUČASNÝM
PROGRAMÁTORSKÝM TRENDŮM

Umění programování

1. díl

Základní algoritmy

ADDISON
WESLEY

COMPRESS

DONALD E. KNUTH

Obrázek 11: Český překlad prvního svazku knihy The Art of Computer Programming (TAOCP) vydalo nakladatelství Computer Press roku 2008.

studenti i zaměstnanci významně přispěli k formulování otázek a hledání odpovědí ve světě digitální typografie a především v rámci čtyřicetileté \TeX ové rodiny.

So, maybe instead of ambitious themes, the only theme that matters is: show what you did and how you did it. (Hans Hagen, [7, p. 32])

Kde jsme nyní a co bude následovat – předpovědi

Nelson Beebe předpovídal budoucnost \TeX u před více než deseti lety [3]. Svět, ve kterém žijeme, se ale neustále mění a ačkoliv většina jeho předpovědí stále platí, některé je třeba konfrontovat s realitou a zrevidovat. Pokusili jsme se vyhodnotit dopad užití \TeX ových nástrojů pomocí statistických dat závěrečných prací obhájovaných nejen na FI MU, ale napříč celou univerzitou.

Se vznikem \LaTeX ové třídy `fithesis3` vstoupila podpora přípravy závěrečných prací do nové éry [23]. Byly vytvořeny ukázkové dokumenty třídy pro každou z devíti fakult Masarykovy univerzity a díky využití webové služby pro přípravu dokumentů v \LaTeX u lze nyní začít psát závěrečnou práci na několik kliknutí myši i bez funkční \TeX ové instalace. Očekáváme, že mnoho uživatelů okouzlených kvalitou výstupu si po takové zkušenosti \TeX nainstaluje i na svá zařízení. Webové služby pro přípravu \TeX ových dokumentů umožňují rychlé učení příkladem, online konzultace, poskytování zpětné vazby vedoucím a současnou práci více osob na jednom dokumentu.

Přenositelnost, stabilita, robustnost a jednodušnost stylu podpořená viditelným značkováním, jednoduchý zápis matematických vzorců a estetické kvality výstupu považujeme za hlavní výhody \TeX u oproti WYSIWYG editorům. Obdobně smýšlejí i studenti FI MU, jak vidno z obrázku 12.

Současně s třídou `fithesis3` bylo vyvinuto beamerové téma `fibeamer` pro přípravu prezentací k obhajobě závěrečných prací bez nutnosti zabírat se formou. Téma bylo uveřejněno na \TeX Live a na webových službách pro přípravu \TeX ových dokumentů. K našemu překvapení se téma těší oblibě především na Filozofické fakultě MU (viz obrázek 13).

Na Masarykově univerzitě studuje přibližně 40 000 studentů a veškeré obhájené závěrečné práce jsou archivovány v univerzitním informačním systému. Pomocí heuristik jsme detekovali, které práce ze vzorku 44 875 prací hájených na MU v letech 2010–2015 byly připraveny pomocí \TeX u. Odhadujeme, že počet závěrečných prací připravených \TeX em průběžně stoupal z 5,67 % v roce 2010 na 6,28 % v roce 2014. Jednoduchou extrapolací dostáváme, že do roku 2783 budou veškeré závěrečné práce připravovány v \TeX u ☺.

Závěrečné práce připravené pomocí \TeX u byly hodnoceny známkou A významně častěji a známkami C a D významně méně často, než práce připravené jinými nástroji [23]. Hodnocení prací je sumarizováno v tabulce 1 a na obrázku 14.

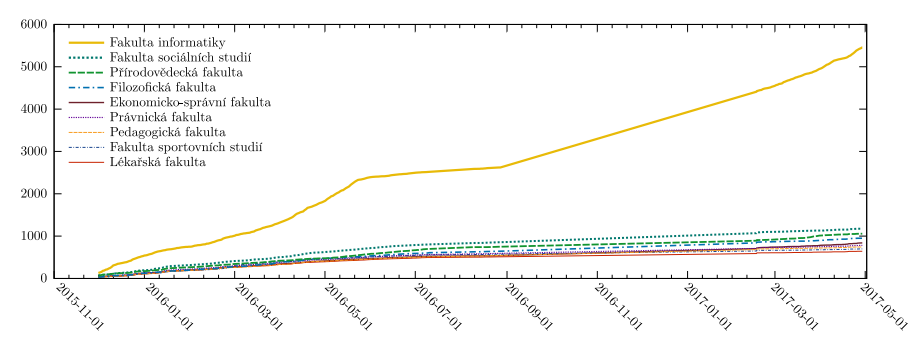
Tabulka 1: Tabulka počtu hájených závěrečných prací s jednotlivými hodnoceními z let 2010–2015. Tabulka vyobrazuje očekávanou (E) a pozorovanou (O) četnost prací s danou známkou připravených T_EXem, kde E je četnost známky vynásobená rel. četností užití T_EXu. $(E - O)^2 / E$ měří neshodu mezi E a O pomocí χ^2 testu.

Hodnocení	Bez T _E Xu	E(T _E Xem)	O(T _E Xem)	$(E - O)^2 / E$
A	15 476	988	1 181	37,858
B	9 999	638	587	4,093
C	7 926	506	381	30,799
D	4 020	257	194	15,248
E	2 783	178	128	13,853
F	1 979	126	145	2,771
Celkem	42 183	2 692	2 692	104,623

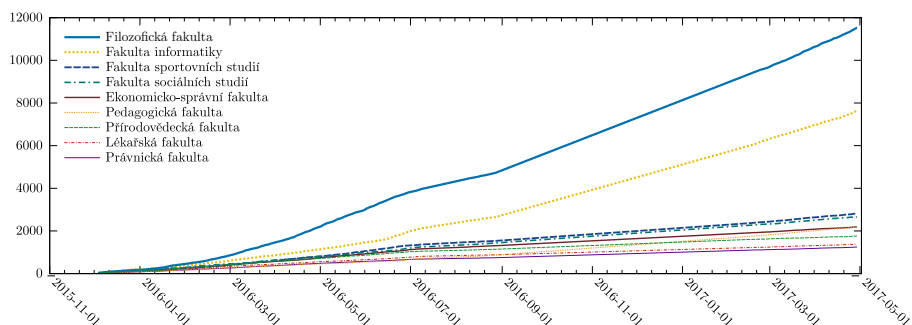
Toto vnímáme jako jasný důkaz toho, že práce připravené T_EXem byly hodnocené lépe, než práce připravené jinými nástroji. V budoucnu bychom rádi ukázali, že známky, které studenti obdrželi za práce připravené v T_EXu, jsou konzistentně lepší, než hodnocení jejich závěrečných zkoušek. Naše hypotéza je taková, že využití T_EXu pomáhá studentům získat za závěrečné práce lepší hodnocení, než jaké odpovídá jejich studijním schopnostem.

Na závěr krátké shrnutí hlavních lekcí z využití T_EXu na MU:

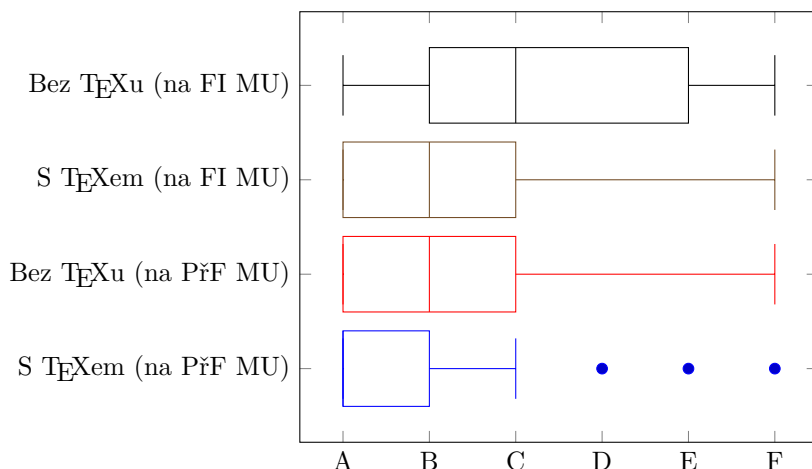
- Udržitelná podpora pro přípravu dokumentů v T_EXu a budování komunity univerzitou hrají významnou roli. Ideální je vytvořit prostředí, ve kterém si



Obrázek 12: Kumulativní počet zhlédnutí šablony I^AT_EXové třídy fithesis3 na webové službě Overleaf.



Obrázek 13: Kumulativní počet zhlédnutí šablony beamerového tématu fibeamer na webové službě Overleaf.



Obrázek 14: Krabicový graf hodnocení závěrečných prací hájených během let 2010–2015 na Fakultě informatiky (FI MU) a Přírodovědecké fakultě (PřF MU) Masarykovy univerzity.

mohou studenti a zaměstnanci společně hrát a experimentovat, pracovat na společných projektech a bavit se.

- Užití \TeX u v každodenním univerzitním životě motivuje studenty k jeho využití pro vlastní projekty a je výhodné pro obě strany – studenti se naučí nové dovednosti a fakultní administrativa i výuka jsou snazší a zábavnější.
- Sazební jádro \TeX u produkuje vizuálně působivé výstupy, které mnohdy předčí alternativní nástroje, a to především pokud je třeba sázet matematické vzorce, jak tomu často je v přírodně-vědních, technických (a matematických) oborech.
- Oproti většině WYSIWYG editorů produkuje \TeX konzistentní výsledky a lze jej snadno použít pro sazbu databázových výstupů a dlouhých dokumentů obsahujících matematiku. Jedná se o bezpečnou volbu, zvlášť pokud univerzita poskytuje oficiální uživatelskou podporu.
- Akademici dobře vědí, že v dnešním proměnlivém světě tvoří oddělení obsahu od formy a \TeX jako pevný bod při přípravě dokumentů silný tandem, jenž umožňuje přípravu dokumentů, které lze s minimálními zásahy opětovně využívat a převádět do nových výstupních formátů.
- Využití \TeX u jako sazebního jádra univerzitního informačního systému se vyplatilo a po desetiletí nebylo zapotřebí měnit technologii.

Neustále se objevují mladí schopní studenti, kteří si hrají s \TeX em, zapojují se do dění v \TeX ové komunitě a pracují na ambiciózních nových projektech. Díky tomu si mohou pracovníci fakulty užít trochu zaslouženého odpočinku:



Reference

- [1] Antoš, D.: PATLIB, knihovna pro manipulaci se vzory. 2001.
URL www.fi.muni.cz/~xantos/patlib/
- [2] Antoš, D.; Sojka, P.: Pattern Generation Revisited. In *Proceedings of the 16th European \TeX Conference, Kerkrade, 2001*, editace S. Pepping, Kerkrade, The Netherlands: NTG, září 2001, s. 7–17.
URL www.ntg.nl/EuroTeX/2001/
- [3] Beebe, N. H. F.: 25 Years of \TeX and METAFONT: Looking back and looking forward — TUG 2003 keynote address. *TUGboat*, ročník 25, č. 1, 2004: s. 7–30, ISSN 0896-3207.
URL tug.org/TUGboat/tb25-1/beebe-2003keynote.pdf
- [4] Cuoco, A.: \TeX in schools: Why not? *TUGboat*, ročník 12, č. 2, červen 1991: s. 303–304, ISSN 0896-3207.
URL tug.org/TUGboat/tb12-2/tb32letters.pdf

- [5] Day, B.: *The Velvet Philosophers*. A&C Black, 1999, ISBN 9781870626422, 344 s.
- [6] Došlá, Z.; Plch, R.; Sojka, P.: *Matematická analýza s programem Maple: 2. Neko-
nečné řady*. CD-ROM, prosinec 2002.
URL www.math.muni.cz/~plch/nkpm/
- [7] Hagen, H.: Children of \TeX . In Przechlewski aj. [28], s. 18–32.
- [8] Holeček, J.; Sojka, P.: Animations in a pdf \TeX -generated PDF. *TUGboat*, ročník 25,
duben 2004: s. 35–41.
- [9] Hrad, M.; Sojka, P.: Automatizace sazby a skenování formulářů. *Zpravodaj CS TUG*,
ročník 12, č. 3–4, 2002: s. 123–139.
- [10] Hàn Thế Thành: *Přenositelný formát dokumentu a sázecí systém \TeX* . Diplomová
práce, Masarykova univerzita, Brno, Fakulta informatiky, duben 1996.
- [11] Hàn Thế Thành: Improving \TeX 's Typeset Layout. *TUGboat*, ročník 19, č. 3, září
1998: s. 284–288, ISSN 0896-3207.
URL tug.org/TUGboat/tb19-3/tb60than.pdf
- [12] Hàn Thế Thành: Micro-typographic extensions to the \TeX typesetting system.
TUGboat, ročník 21, č. 4, prosinec 2000: s. 317–434.
URL tug.org/TUGboat/tb21-4/tb69thanh.pdf
- [13] Hàn Thế Thành: Margin kerning and font expansion with pdf \TeX . *TUGboat*,
ročník 22, č. 3, září 2001: s. 146–148, ISSN 0896-3207.
URL tug.org/TUGboat/tb22-3/tb72thanh.pdf
- [14] Hàn Thế Thành: Micro-typographic extensions of pdf \TeX in practice. *TUGboat*,
ročník 25, č. 1, 2004: s. 35–38, ISSN 0896-3207.
URL tug.org/TUGboat/tb25-1/thanh.pdf
- [15] Hàn Thế Thành; Rahtz, S.: The pdf \TeX user manual. *TUGboat*, ročník 18, č. 4,
prosinec 1997: s. 249–254, ISSN 0896-3207.
URL tug.org/TUGboat/tb18-4/tb57than.pdf
- [16] Janík, P.: *Digitální formáty písma v počítačové sazbě*. Diplomová práce, Masarykova
univerzita, Brno, Fakulta informatiky, březen 2000.
- [17] Jarmar, M.: *Conversion of Mathematical Documents into Braille*. Diplomová práce,
leden 2012, Masarykova univerzita, Brno, Fakulta informatiky (vedoucí práce Petr
Sojka).
URL is.muni.cz/th/172981/fi_m/
- [18] Knuth, D. E.: Theory and Practice. Keynote address for the 11th World Computer
Congress (Information Processing '89), srpen 1989.
- [19] Laube, S.: \TeX in Schools: Just Say Yes! *TUGboat*, ročník 36, č. 3, 2015: s. 188–189,
ISSN 0896-3207.
URL tug.org/TUGboat/tb36-3/tb114laube.pdf
- [20] Liška, M.: *Evaluation of Mathematics Retrieval*. Diplomová práce, leden 2013,
Masarykova univerzita, Brno, Fakulta informatiky (vedoucí práce Petr Sojka).
URL is.muni.cz/th/255768/fi_m/

- [21] Misáková, M.: *Kvalitní typografie v počítačové sazbě*. Diplomová práce, 1998, Masarykova univerzita, Brno, Fakulta informatiky (vedoucí práce Petr Sojka).
- [22] Neuwirth, K.: $\text{T}_{\text{E}}\text{X}$ in Schools: Just Say No. *TUGboat*, ročník 12, č. 1, březen 1991: s. 171–174, ISSN 0896-3207.
URL tug.org/TUGboat/tb12-1/tb31kneuwirth.pdf
- [23] Novotný, V.: Forma odborných závěrečných prací v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u. 2015, Bakalářská práce, Masarykova univerzita, Brno, Fakulta informatiky (vedoucí práce Petr Sojka).
URL is.muni.cz/th/409729/fi_b/
- [24] Novotný, V.: Using Markdown inside $\text{T}_{\text{E}}\text{X}$ Documents. In Przechlewski aj. [28], s. 50–53.
- [25] Pazdziora, J.: Algoritmy řádkového a stránkového zlomu v počítačové sazbě. 1997, Diplomová práce, Masarykova univerzita, Brno, Fakulta informatiky (vedoucí práce Petr Sojka).
URL is.muni.cz/th/2644/fi_m/
- [26] Pazdziora, J.; Brandejs, M.: University Information System Fully Based on WWW. In *ICEIS 2000 Proceedings*, Escola Superior de Tecnologia do Instituto Politécnico de Setúbal, 2000, ISBN 972-98050-1-6, s. 467–471.
URL is.muni.cz/auth/clanky/2000_ICEIS.pl
- [27] Popelková, Z.: Makra pro sazbu rozvrhových tabulek. leden 2001, Bakalářská práce, Masarykova university, Brno, Fakulta informatiky (vedoucí práce Libor Škarvada).
URL is.muni.cz/th/3839/fi_b/
- [28] Przechlewski, T.; Berry, K.; Ludwichowski, J. (editoři): *XXV Międzynarodowa Konferencja Użytkowników Systemu $\text{T}_{\text{E}}\text{X}$: Materiały konferencyjne*, GUST, 2017, ISBN 978-83-939016-4-7.
- [29] Růžička, M.: Automated Processing of $\text{T}_{\text{E}}\text{X}$ -typeset Articles for a Digital Library. In Sojka [37], s. 167–176.
URL dml.cz/dmlcz/702564
- [30] Skoupý, K.: $\text{N}_{\text{T}}\text{S}$: a New Typesetting System. *TUGboat*, ročník 19, č. 3, září 1998: s. 318–322, ISSN 0896-3207.
URL tug.org/TUGboat/tb19-3/tb60nts.pdf
- [31] Sojka, P.; Hàn Thế Thành; Zlatuška, J.: The Joy of $\text{T}_{\text{E}}\text{X}2\text{PDF}$ — Acrobatics with an alternative to DVI format. *TUGboat*, ročník 17, č. 3, 1996: s. 244–251.
URL tug.org/TUGboat/tb17-3/tb52sojk.pdf
- [32] Sojka, P.: Notes on compound word hyphenation in $\text{T}_{\text{E}}\text{X}$. *TUGboat*, ročník 16, č. 3, září 1995: s. 290–296, ISSN 0896-3207.
URL tug.org/TUGboat/tb16-3/tb48soj2.pdf
- [33] Sojka, P.: Hyphenation on Demand. *TUGboat*, ročník 20, č. 3, 1999: s. 241–247.
URL tug.org/TUGboat/tb20-3/tb64sojka.pdf
- [34] Sojka, P.: Animations in PDF. In *Proceedings of the 8th SIGCSE Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2003*, Thessaloniki: Association for Computing Machinery, 2003, ISBN 1-58113-672-2, str. 263.

- [35] Sojka, P.: Interactive Teaching Materials in PDF using JavaScript. In *Proceedings of the 8th SIGCSE Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2003*, Thessaloniki: Association for Computing Machinery, 2003, ISBN 1-58113-672-2, str. 275.
- [36] Sojka, P.: Rapid Evaluation using Multiple Choice Tests and T_EX. In *Proceedings of the 8th SIGCSE Annual Conference on Innovation and Technology in Computer Science Education, ITiCSE 2003*, Thessaloniki: Association for Computing Machinery, 2003, ISBN 1-58113-672-2, str. 265.
- [37] Sojka, P. (editor): *Towards a Digital Mathematics Library*, Birmingham, UK: Masaryk University, červenec 2008, ISBN 978-80-210-4658-0.
URL dml.cz/dmlcz/702564
- [38] Sojka, P.: Digitization Workflow in the Czech Digital Mathematics Library. In *Computer Mathematics*, editace R. Feng; W.-s. Lee; Y. Sato, Springer-Verlag, říjen 2014, ISBN 978-3-662-43798-8, s. 147–156, DOI 10.1007/978-3-662-43799-5_13.
- [39] Sojka, P.; Antoš, D.: Context Sensitive Pattern Based Segmentation: A Thai Challenge. In *Proceedings of EACL 2003 Workshop on Computational Linguistics for South Asian Languages — Expanding Synergies with Europe*, editace P. Hall; D. D. Rao, Budapest, duben 2003, ISBN 1-932432-02-7, s. 65–72.
- [40] Sojka, P.; Červenka, R.; Svoboda, M.: T_EX for database publishing. In Zlatuška [48], s. 53–58.
- [41] Sojka, P.; Hatlapatka, R.: Document Engineering for a Digital Library: PDF recompression using JBIG2 and other optimization of PDF documents. In *Proceedings of the ACM Conference on Document Engineering, DocEng 2010*, Manchester: Association for Computing Machinery, září 2010, ISBN 978-1-4503-0231-9, s. 3–12, DOI 10.1145/1860559.1860563.
- [42] Sojka, P.; Plch, R.: Technological Challenges of Teaching Mathematics in a Blended Learning Environment. *International Journal of Continuing Engineering Education and Life-Long Learning*, ročník 18, č. 5-6, 2008: s. 657–665, ISSN 1560-4624, DOI 10.1504/IJCEELL.2008.022172.
- [43] Sojka, P.; Rákosník, J.: From Pixels and Minds to the Mathematical Knowledge in a Digital Library. In Sojka [37], s. 17–27.
URL dml.cz/dmlcz/702564
- [44] Sojka, P.; Růžička, M.: Single-source publishing in multiple formats for different output devices. *TUGboat*, ročník 29, č. 1, 2008: s. 118–124, ISSN 0896-3207.
URL tug.org/TUGboat/tb29-1/tb91sojka.pdf
- [45] Sojka, P.; Ševeček, P.: Hyphenation in T_EX — Quo Vadis? *TUGboat*, ročník 16, č. 3, září 1995: s. 280–289, ISSN 0896-3207.
URL tug.org/TUGboat/tb16-3/tb48soj1.pdf
- [46] Sylwestrzak, W.; Borbinha, J.; Bouche, T.; aj.: EuDML—Towards the European Digital Mathematics Library. In *Proceedings of DML 2010*, editace P. Sojka, Paris, France: Masaryk University, červenec 2010, ISBN 978-80-210-5242-0, s. 11–24.
URL dml.cz/dmlcz/702569

- [47] Zlatuška, J.: Automatic generation of virtual fonts with accented letters for T_EX. *Cahiers GUTenberg*, ročník 10–11, září 1991: s. 57–68.
- [48] Zlatuška, J. (editor): *Proceedings of the 7th European T_EX Conference, Prague, 1992*, Masaryk University, Brno, září 1992, ISBN 80-210-0480-0.
- [49] Zlatuška, J.: When METAFONT does it alone. *TUGboat*, ročník 16, č. 3, září 1995: s. 227–232, ISSN 0896-3207.
URL tug.org/TUGboat/tb16-3/tb48zlat.pdf
- [50] Zlatuška, J.: *N_TS: Programming Languages and Paradigms*. In *EuroT_EX Proceedings*, Heidelberg: DANTE, 1999, s. 241–246.

Summary: T_EX in Schools? Just Say Yes: The use of T_EX at the Faculty of Informatics, Masaryk University

Students at Masaryk University (MU) use T_EX for many purposes, such as writing theses, essays, and papers. It is also used by the staff for teaching electronic publishing and literate programming, for writing scientific papers, quizzes and teaching resources, and for generating documents and web pages from university databases by the university information system. T_EX and related technologies have been systematically supported and deployed at the Faculty of Informatics of MU (FI MU) for more than two decades. In this paper, we describe the T_EX-related support and projects that we have realized at various levels. These include the design of the Faculty’s visual identity, resources for teaching electronic publishing, and for database publishing directly from the University’s information system. We evaluate the outcomes, and consider some possible future deployments of T_EX-related technologies. With the data analytics of `fithesis3` class support and its use at MU, we give arguments why the answer to the often-asked question in the title is in the affirmative, at least for computer science schools like ours and for authoring math publications.

*Petr Sojka, sojka@fi.muni.cz
Vít Novotný, witiko@mail.muni.cz*

In L^AT_EX, it is now possible to run snippets of LUA code from within METAPOST. The article describes the mechanism, the low-level interface available in L^AT_EX, as well as the high-level interface available in C^ON^TE^XT through example.

Keywords: LUA, L^AT_EX, C^ON^TE^XT, METAPOST, MPLIB

Introduction

Already for a some years I have been wondering how it would be if we could escape to LUA inside METAPOST, or in practice, in MPLIB in L^AT_EX. The idea is simple: embed LUA code in a METAPOST file that gets run as soon as it's seen. In case you wonder why LUA code makes sense, imagine generating graphics using external data. The capabilities of LUA to deal with that is more flexible and advanced than in METAPOST. Of course we could generate a METAPOST definition of a graphic from data but often it makes more sense to do the reverse. I finally found time and reason to look into this and in the following sections I will describe how it's done.

The basics

The approach is comparable to L^AT_EX's `\directlua`. That primitive can be used to execute LUA code and in combination with `tex.print` we can pipe back strings into the T_EX input stream. There is a complication that we have to be able to operate under different so called catcode regimes: the meaning of characters can differ per regime. We also have to deal with line endings in special ways as they relate to paragraphs and such. In METAPOST we don't have that complication so getting back input into the METAPOST input, we can do so with simple strings. For that a mechanism similar to `scantokens` can be used. That way we can return anything (including nothing) as long as METAPOST can interpret it and as long as it fulfils the expectations.

```
numeric n ;  
n := scantokens("123.456") ;  
A script is run as follows:  
numeric n ;  
n := runscript("return '123.456'") ;
```


This primitive doesn't have the word `lua` in its name so in principle any wrapper around the library can use it as hook. In the case of `LUATEX` the script language is of course `LUA`. At the `METAPOST` end we only expect a string. How that string is constructed is completely up to the `LUA` script. In fact, the user is completely free to implement the runner any way she or he wants, like:

```
local function scriptrunner(code)
  local f = loadstring(code)
  if f then
    return tostring(f())
  else
    return ""
  end
end
```

This is hooked into an instance as follows:

```
local m = mplib.new {
  ...
  run_script = scriptrunner,
  ...
}
```

Now, beware, this is not the `ConTEXt` way. We provide print functions and other helpers, which we will explain in the next section.

Helpers

After I got this feature up and running I played a bit with possible interfaces at the `ConTEXt` (read: `METAFUN`) end and ended up with a bit more advanced runner where no return value is used. The runner is wrapped in the `lua` macro.

```
numeric n ;
n := lua("mp.print(12.34567)") ;
draw texttext(n) xsize 4cm withcolor maincolor ;
```

This renders as:

12.34567

In case you wonder how efficient calling `LUA` is, don't worry: it's fast enough, especially if you consider suboptimal `LUA` code and the fact that we switch between machineries.

```
draw image (
  lua("statistics.starttiming()") ;
  for i=1 upto 5000 :
```

```

draw lua ("mp.pair
  (math.random(-74,270),
    math.random(-22,22))" ) ;
endfor ;
lua("statistics.stoptiming()") ;
draw texttext(lua
  ("mp.print(
    statistics.elapsedtime()")
  ) ysize 40 ;
) withcolor maincolor
  withpen pencircle scaled 1 ;

```

Here the part:

```

draw lua ("mp.pair
  (math.random(-74,270),
    math.random(-22,22))" ) ;

```

effectively becomes (for instance):

```
draw scantokens "(25,15)" ;
```

which in turn becomes:

```
draw scantokens (25,15) ;
```

The same happens with this:

```

draw texttext (lua
  ("mp.print
    (statistics.elapsedtime()")
  ) ) ...

```

This becomes for instance:

```
draw texttext(scantokens "1.23") ...
```

and therefore:

```
draw texttext(1.23) ...
```

We can use `mp.print` here because the `texttext` macro can deal with numbers.

The next also works:

```

draw texttext(lua
  ("mp.quoted
    (statistics.elapsedtime()")
  )
) ...

```

Now we get (in METAPost speak):

```

draw texttext(scantokens
  (ditto & "1.23" & ditto) ...

```

Here `ditto` represents the double quotes that mark a string. Of course, because we pass the strings directly to `scantokens`, there are no outer quotes at all, but this is how it can be simulated. In the end we have:

```
draw texttext("1.23") ...
```

What you use, `mp.print` or `mp.quoted` depends on what the expected code is: an assignment to a numeric can best be a number or an expression resulting in a number. This graphic becomes:



The runtime on my current machine is some 0.25 seconds without and 0.12 seconds with caching. But to be honest, speed is not really a concern here as the amount of complex METAPOST graphics can be neglected compared to extensive node list manipulation. With `LUAJITTEX` generating the graphic takes 15% less time.¹

The three print command accumulate their arguments:

```
numeric n ;  
n := lua("mp.print(1) mp.print('+') mp.print(2)") ;  
draw texttext(n) xsize 1cm  
withcolor maincolor ;
```

As expected we get:



Equally valid is:

```
numeric n ;  
n := lua("mp.print(1,'+',2)") ;  
draw texttext(n) xsize 1cm  
withcolor maincolor ;
```

This gives the same result:



¹Processing a small 8 page document like this takes about one second, which includes loading a bunch of fonts.

Of course all kind of action can happen between the prints. It is also legal to have nothing returned as could be seen in the 10.000 dot example: there the timer related code returns nothing so effectively we have `scantokens("")`. Another helper is `mp.quoted`, as in:

```
draw texttext
  (lua
    ("mp.quoted
      ('@0.3f',
        " & decimal n & "
      )"
    )
  )
) withcolor maincolor ;
```

This typesets 3.000. Watch the @. When no percent character is found in the format specifier, we assume that an @ is used instead.

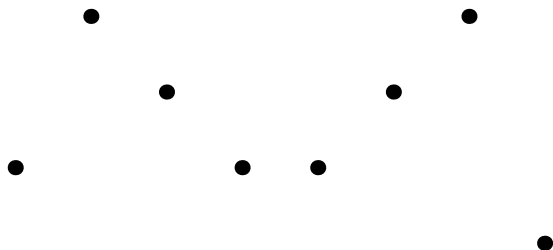
But, the real benefit of embedded LUA is when we deal with data that is stored at the LUA end. First we define a small dataset:

```
\startluacode
table.save("demo-data.lua",
{
  { 1, 2 }, { 2, 4 }, { 3, 3 },
  { 4, 2 }, { 5, 2 }, { 6, 3 },
  { 7, 4 }, { 8, 1 },
}
)
\stopluacode
```

There are several ways to deal with this table. I will show clumsy as well as better looking ways.

```
lua("MP = { }
MP.data = table.load('demo-data.lua')"
) ;
numeric n ;
lua("mp.print('n := ',\#MP.data)" ) ;
for i=1 upto n :
  drawdot
    lua("mp.pair
      (MP.data[" & decimal i & "]" )"
    ) scaled cm
    withpen pencircle scaled 2mm
    withcolor maincolor ;
endfor ;
```

Here we load a LUA table and assign the size to a METAPOST numeric. Next we loop over the table entries and draw the coordinates.



We will stepwise improve this code. In the previous examples we omitted wrapper code but here we show it:

```
\startluacode
  MP.data = table.load('demo-data.lua')
  function MP.n()
    mp.print(#MP.data)
  end
  function MP.dot(i)
    mp.pair(MP.data[i])
  end
\stopluacode
```

```
\startMPcode
  numeric n ;
  n := lua("MP.n()") ;
  for i=1 upto n :
    drawdot
      lua("MP.dot
        (" & decimal i & ")")
      ) scaled cm
    withpen pencircle scaled 2mm
    withcolor maincolor ;
  endfor ;
\stopMPcode
```

So, we create a few helpers in the MP table. This table is predefined so normally you don't need to define it. You may however decide to wipe it clean.



You can decide to hide the data:

```
\startluacode
  local data = { }
  function MP.load(name)
    data = table.load(name)
  end
  function MP.n()
    mp.print(#data)
  end
  function MP.dot(i)
    mp.pair(data[i])
  end
\stopluacode
```

It is possible to use less LUA, for instance in:

```
\startluacode
  local data = { }
  function MP.loaded(name)
    data = table.load(name)
    mp.print(#data)
  end
  function MP.dot(i)
    mp.pair(data[i])
  end
\stopluacode
```

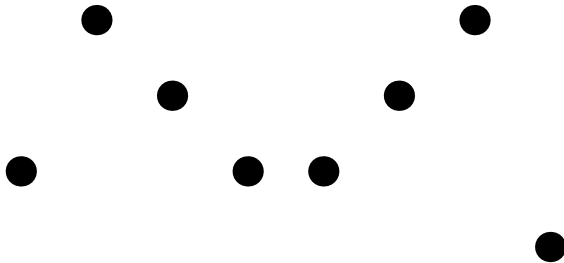
```
\startMPcode
  for i=1 upto
    lua
      ("MP.loaded
        ('demo-data.lua')")
    ) :
  drawdot
    lua("MP.dot(",i,")") scaled cm
```

```

        withpen pencircle scaled 4mm
        withcolor maincolor ;
    endfor ;
\stopMPcode

```

Here we also omit the `decimal` because the `lua` macro is clever enough to recognize it as a number.



By using some METAPOST magic we can even go a step further in readability:

```

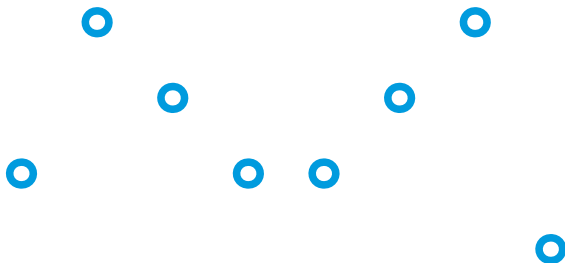
\startMPcode{doublefun}
  cmykcolor maincolor;
  maincolor := (1,.15,0,0);
  lua.MP.load("demo-data.lua") ;

  for i=1 upto lua.MP.n() :
    drawdot lua.MP.dot(i) scaled cm
    withpen pencircle scaled 4mm
    withcolor maincolor ;
  endfor ;

  for i=1 upto MP.n() :
    drawdot MP.dot(i) scaled cm
    withpen pencircle scaled 2mm
    withcolor white ;
  endfor ;
\stopMPcode

```

Here we demonstrate that it also works ok in `double` mode, which makes much sense when processing data from other sources. Watch how we omit the `.lua` prefix: the `MP` macro will deal with that.



So in the end we can simplify the code that we started with to:

```
\startMPcode{doublefun}
  for i=1 upto
    MP.loaded("demo-data.lua") :
    drawdot
      MP.dot(i) scaled cm
      withpen pencircle scaled 2mm
      withcolor maincolor ;
  endfor ;
\stopMPcode
```

Access to variables

The question with such mechanisms is always: how far should we go. Although METAPost is a macro language it has properties of procedural languages. It also has more introspective features at the user end. For instance, one can loop over the resulting picture and manipulate it. This means that we don't need full access to METAPost internals. However, it makes sense to provide access to basic variables: **numeric**, **string**, and **boolean**.

```
draw texttext(lua
  ("mp.quoted
    ('@0.15f',
      mp.get.numeric('pi')-math.pi
    )"
  )
)
  ysize .5cm
  withcolor maincolor ;
```

In double mode you will get zero printed but in scaled mode we definitely get a difference:

-0.000006349878856

In the next example we use `mp.quoted` to make sure that indeed we pass a string. The `texttext` macro can deal with numbers but an unquoted `yes` or `no` is asking for problems.

```
boolean b ;
b := true ;
draw texttext(
  lua
    ("mp.quoted(mp.get.boolean('b')
      and 'yes' or 'no')")
)
ysized 1cm
withcolor maincolor ;
```

Especially when more text is involved it makes sense to predefine a helpers in the MP namespace if only because METAPost (currently) doesn't like newlines in the middle of a string, so a `lua` call has to be on one line.

yes

Here is an example where `LUA` does something that would be close to impossible, especially if more complex text is involved.

```
string s ;
s := "TEX" ; % ""
draw texttext
  (lua
    ("mp.quoted
      (characters.lower
        (mp.get.string('s'))
      )
    )"
  )
)
ysized 1cm
withcolor maincolor ;
```

As you can see here, the whole repertoire of helper functions can be used in a `METAFUN` definition.

The library

In ConT_EXt we have a dedicated runner, but for the record we mention the low level constructor:

```
local m = mplib.new {  
  ...  
  script_runner = function(s) return  
    loadstring(s)() end,  
  script_error = function(s)  
    print(s) end,  
  ...,  
}
```

An instance (in this case `m`) has a few extra methods. Instead you can use the helpers in the library.

<code>m:get_numeric(name)</code>	returns a numeric (double)
<code>m:get_boolean(name)</code>	returns a boolean (<code>true</code> or <code>false</code>)
<code>m:get_string (name)</code>	returns a string

<code>mplib.get_numeric(m,name)</code>	returns a numeric (double)
<code>mplib.get_boolean(m,name)</code>	returns a boolean (<code>true</code> or <code>false</code>)
<code>mplib.get_string (m,name)</code>	returns a string

In ConT_EXt the instances are hidden and wrapped in high level macros, so there you cannot use these commands.

ConT_EXt helpers

The `mp` namespace provides the following helpers:

<code>print(...)</code>	returns one or more values
<code>pair(x,y)</code> and <code>pair(t)</code>	returns a proper pair
<code>triplet(x,y,z)</code> and <code>triplet(t)</code>	returns an RGB color
<code>quadruple(w,x,y,z)</code> and <code>quadruple(t)</code>	returns an CMYK color
<code>format(fmt,...)</code>	returns a formatted string
<code>quoted(fmt,...)</code> and <code>quoted(s)</code>	returns a (formatted) quoted string
<code>path(t[,connect][,close])</code>	returns a connected (closed) path

The `mp.get` namespace provides the following helpers:

<code>numeric(name)</code>	gets a numeric from METAPOST
<code>boolean(name)</code>	gets a boolean from METAPOST
<code>string(name)</code>	gets a string from METAPOST

Paths

In the meantime we got several questions on the ConT_EXt mailing list about turning coordinates into paths. Now imagine that we have this dataset:

```
10 20 20 20 -- sample 1
30 40 40 60
50 10
```

```
10 10 20 30 % sample 2
30 50 40 50
50 20
```

```
10 20 20 10 # sample 3
30 40 40 20
50 10
```

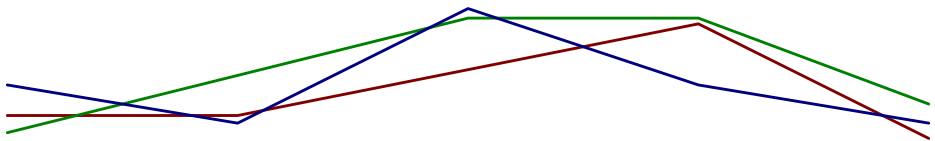
In this case I've put the data in a buffer so that it can be shown here as well as used in a demo. Watch how we can add comments. The following code converts this into a table with three subtables.

```
\startluacode
  MP.myset =
    mp.dataset
      (buffers.getcontent("dataset"))
\stopluacode
```

We use the MP (user) namespace to store the table. Next we turn these subtables into paths:

```
\startMPcode
  for i=1 upto
    lua("mp.print(mp.n(MP.myset))") :
  draw
    lua("mp.path
      (MP.myset[" & decimal i & "]
      )"
    )
    xysized (HSize,10ExHeight)
    withpen
      pencircle scaled .25ExHeight
    withcolor basiccolors[i]/2 ;
  endfor ;
\stopMPcode
```

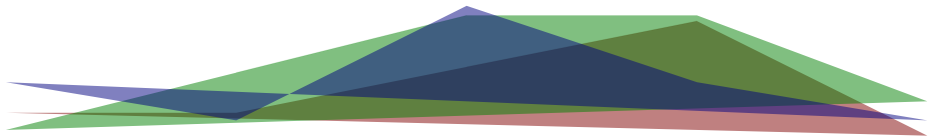
This gives:



Instead we can fill the path in which case we also need to close it. The `true` argument deals with that:

```
\startMPcode
  for i=1 upto
    lua("mp.print(mp.n(MP.myset))") :
  path p ; p :=
    lua("mp.path
      (MP.myset
        [\" & decimal i & \"],
        true
      )"
    )
    xysized (HSize,10ExHeight) ;
  fill p
    withcolor basiccolors[i]/2
    withtransparency (1,.5) ;
  endfor ;
\stopMPcode
```

We get:



The following makes more sense:

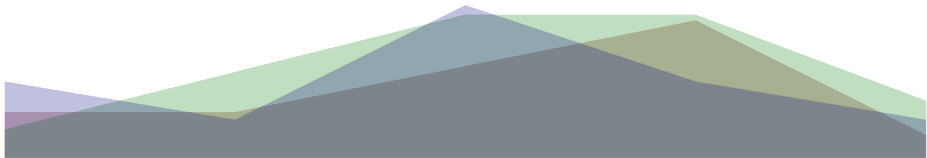
```
\startMPcode
  for i=1 upto
    lua("mp.print
      (mp.n(MP.myset))"
    ) :
  path p ;
  p := lua("mp.path
    (MP.myset[\" & decimal i & \"])")
  )
  xysized (HSize,10ExHeight) ;
  p :=
```

```

(xpart llcorner boundingbox p,0)
-- p --
(xpart lrcorner boundingbox p,0)
-- cycle ;
fill p
  withcolor basiccolors[i]/2
  withtransparency (1,.25) ;
endfor ;
\stopMPcode

```

So this gives:



This (area) fill is so common that we have a helper for it:

```

\startMPcode
for i=1 upto
  lua("mp.size(MP.myset)") :
fill area
  lua("mp.path
    (MP.myset[" & decimal i & "])"
  )
  xysized (HSize,5ExHeight)
  withcolor basiccolors[i]/2
  withtransparency (2,.25) ;
endfor ;
\stopMPcode

```

So this gives:



This snippet of METAPOST code still looks kind of horrible so how can we make it look better? Here is an attempt, First we define a bit more LUA:

```

\startluacode
local data =
  mp.dataset
    (buffers.getcontent("dataset"))

```

```

MP.dataset = {
  Line = function(n) mp.path(data[n]) end,
  Size = function() mp.size(data) end,
}
\stopluacode

```

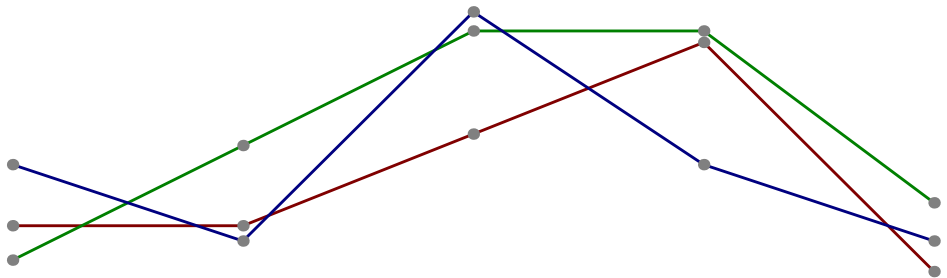
We can now make the METAPOST look more natural. Of course this is possible because in METAFUN the lua macro does some extra work.

```

\startMPcode
  for i=1 upto
    lua.MP.dataset.Size() :
  path p ;
  p := lua.MP.dataset.Line(i)
    xysized (HSize,20ExHeight) ;
  draw
    p
    withpen pencircle scaled .25ExHeight
    withcolor basiccolors[i]/2 ;
  drawpoints
    p
    withpen pencircle scaled ExHeight
    withcolor .5white ;
  endfor ;
\stopMPcode

```

As expected, we get the desired result:



Once we start making things look nicer and more convenient, we quickly end up with helpers like the once in the next example. First we save some demo data in files:

```

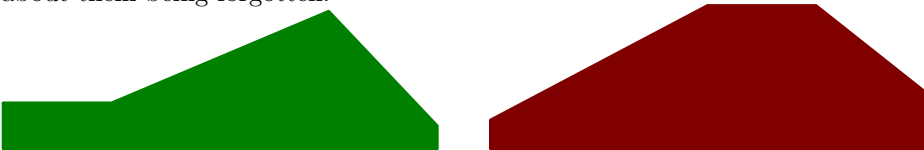
\startluacode
  io.savedata("foo.tmp","10 20 20 20 30 40 40 60 50 10")
  io.savedata("bar.tmp","10 10 20 30 30 50 40 50 50 20")
\stopluacode

```

We load the data in datasets:

```
\startMPcode
lua.mp.datasets.load("foo","foo.tmp");
lua.mp.datasets.load("bar","bar.tmp");
fill area
lua.mp.datasets.foo.Line()
xysized (HSize/2-EmWidth,10ExHeight)
withpen
  pencircle scaled .25ExHeight
withcolor green/2 ;
fill area
lua.mp.datasets.bar.Line()
xysized (HSize/2-EmWidth,10ExHeight)
shifted (HSize/2+EmWidth,0)
withpen
  pencircle scaled .25ExHeight
withcolor red/2 ;
\stopMPcode
```

Because the datasets are stores by name we can use them without worrying about them being forgotten:



If no tag is given, the filename (without suffix) is used as tag, so the following is valid:

```
\startMPcode
lua.mp.datasets.load("foo.tmp") ;
lua.mp.datasets.load("bar.tmp") ;
\stopMPcode
```

The following methods are defined for a dataset:

method	usage
Size	the number of subsets in a dataset
Line	the joined pairs in a dataset making a non-closed path
Data	the table containing the data (in subsets, so there is always at least one subset)

Due to limitation is METAPOST suffix handling the methods start with an up-percase character.

Remark

The features described here are at this moment still experimental but the interface will not change. There might be a few more accessors and for sure there will be more LUA helpers provided. As usual I need some time to play with it before I make up my mind. It is also possible to optimize the METAPOST-LUA script call a bit but I might do that later.

When we played with this interface we ran into problems with loop variables and macro arguments. These are internally kind of anonymous. Take this:

```
for i=1 upto 100 : draw(i,i) endfor ;
```

The `i` is not really a variable with name `i` but becomes an object (capsule) when the condition is scanned, and a reference to that object when the body is scanned. The body of the for loop gets expanded for each step but at that time there is no longer a variable `i`. The same is true for variables in:

```
def foo(expr x, y, delta) =  
  draw (x+delta,y+delta)  
enddef ;
```

We are still trying to get this right with the LUA interface. Interesting is that when we were exploring this, we ran into quite some cases where we could make METAPOST abort due some memory or stack overflow. Some are just bugs in the new code (due to the new number model) while others come with the design of the system: border cases that never seem to happen in interactive use while the library use assumes no interaction in case of errors.

In ConT_EXt there are more features and helpers than shown here but these are discussed in the METAFUN manual.

Lua v MetaPostu

V L^AT_EXu lze nyní spouštět skripty v jazyce LUA z interpretru jazyka METAPOST. Článek na příkladech popisuje samotný mechanismus, nízkoúrovňové rozhraní na úrovni L^AT_EXu i vysokoúrovňové rozhraní pro formát ConT_EXt.

Klíčová slova: LUA, L^AT_EX, ConT_EXt, METAPOST, MPLIB

Hans Hagen, pragma@wxs.nl

Abstrakt

Článek ukazuje možnosti nastavení \TeX ových odstavcových parametrů pro různá i méně typická zarovnání odstavce.

Klíčová slova: \LaTeX , balíček `shapepar`, odstavce, zarovnání.

Whose waves do glister by the Queen's bright beams.
Which makes them murmur as they passe away.

Poems and Fancies

MARGARET CAVENDISH

Cílem tohoto seriálu je ukázat čtenáři krátké kousky kódu, které mohou vyřešit některé z jeho problémů. Doufám, že situaci ještě více nezkomplikuji v důsledku mých chyb. Opravy, poznámky a návrhy na změny budou vždy vítány.

Poslední dobou se ve skupině `comp.text.tex` vyskytla spousta otázek o formátování odstavců.

for life's not a paragraph
and death I think is no parenthesis.

Since feeling is first

E. E. CUMMINGS

1. Běžné formátování odstavců

Běžný odstavec vysázený v \LaTeX u vypadá následovně.

Běžný odstavec je v \LaTeX u vysázený stejně jako tento. První řádek je zleva odsazený, další řádky jsou zarovnány do bloku a poslední řádek je zprava nezarovnaný.

Odsazení prvního řádku odstavce je dáno hodnotou `\parindent`. Pokud chceme mimořádně začít odstavec bez odsazení prvního řádku, použijeme na začátku odstavce příkaz `\noindent`.

Z anglického originálu *Glisterings* [8] přeložil Jan Šustek.

Další běžně používaná zarovnání odstavců jsou zarovnání doleva, zarovnání doprava a zarovnání na střed.

V některých případech je vhodné vytvořit odstavec s více než jedním odsazeným řádkem. \TeX pro tyto případy nabízí dva příkazy. (Balíček `hanging` [7] toto řeší formou \LaTeX ového prostředí.)

Šířku odsazení určuje příkaz `\hangindent<šířka>`, zatímco počet odsazených řádků určuje příkaz `\hangafter<číslo>`. Pokud je `<číslo>` kladné, pak budou odsazeny řádky `<číslo> + 1`, `<číslo> + 2` a další, zatímco pokud je `<číslo>` záporné, pak je odsazeno prvních `-<číslo>` řádků. Pokud je `<šířka>` kladná, pak jsou řádky odsazeny zleva o `<šířka>`, pokud je záporná, pak jsou řádky odsazeny zprava o `-<šířka>`. Například při nastavení

```
1 \hangindent=3pc \hangafter=-2
```

dostaneme následující výsledek

První dva řádky jsou odsazeny
zleva o 3pc a další řádky odsazeny
nejsou. První řádek je navíc odsazen ještě
o `\parindent`. Tyto příkazy můžete použít jak
přímo v \TeX u, tak také v \LaTeX u.

Příkazy `\hangindent` a `\hangafter` je nutné použít opakovaně pro každý odstavec, který má být odsazený.

\LaTeX disponuje interním makrem `\@hangfrom`, které používá pro několik účelů, například pro sazbu nadpisů. Zde uvádím čitelnější verzi tohoto makra.

```
2 \makeatletter % pokud není v souboru .cls nebo .sty
3 \newcommand*{\hangfrom}[1]{%
4   \setbox\@tempboxa\hbox{#{1}}}%
5   \hangindent \wd\@tempboxa
6   \noindent\box\@tempboxa}
7 \makeatother % pokud není v souboru .cls nebo .sty
```

Makro `\hangfrom{<text>}` použité na začátku odstavce způsobí, že druhý a další řádek odstavce budou odsazeny o přirozenou šířku parametru `<text>`. Použití si ukážeme na příkladu.

```
8 \hangfrom{$\Longrightarrow$\quad}Tento odstavec má řádky
9   odsazené podle šířky argumentu makra \p{hangfrom}.
10  To se může někdy hodit.
```

\Rightarrow Tento odstavec má řádky odsazené podle
šířky argumentu makra `\hangfrom`. To
se může někdy hodit.

T_EX umožňuje vysázet odstavce i velmi netypických tvarů.

Libovolný tvar odstavce
můžeme v T_EXu nastavit pří-
kazem `\parshape`. Pro L^AT_EXové
uživatele Donald Arseneau vy-
tvořil balíček `shapepar` [1],
který umož-ňuje speci-
ální tvary odstavce na-
stavit příjemnějším způso-
bem. Takové tvary odstavce
však doporu-čuji používat
pouze velmi zřídka a pouze
v případě, že ta-kový tvar od-
stavce opravdu musíme použít. Tento
odstavec byl vysázen s pomocí
balíčku `shapepar` s nastavením
tvaru odstavce `\nutshape`.

Balíček `shapepar` má předdefinováno několik tvarů odstavce a obsahuje makra,
která automaticky vygenerují příslušné parametry odstavce.

For precept must be upon precept; precept upon
precept; line upon line; line upon line; here a little,
and there a little.

Isiah

CH. 28, v. 10

2. Konkrétní formátování odstavců

Kromě příkazů `\hangafter`, `\hangindent` a `\parshape` má T_EX další čtyři para-
metry, které ovlivňují tvar běžných odstavců. Délkový registr `\parindent` nasta-
vuje odsazení prvního řádku odstavce. Mezery velikosti `\leftskip` a `\rightskip`
jsou vloženy na začátek a na konec každého řádku. Mezera velikosti `\parfillskip`
je vložena na konec posledního řádku odstavce. Poslední tři uvedené příkazy mají
charakter pružné délky a jejich změnou dostaneme konkrétní zarovnání odstavce.

Předpokládejme, že jsme si nastavili následující délkové registry.

```
11 \newskip\Zeroskip \Zeroskip=0pt plus 0pt minus 0pt
12 \newskip\Flushglue \Flushglue=0pt plus 1fil
```

L^AT_EX definuje makra `\raggedright`, `\raggedleft` a `\centering`, která nastavují následující hodnoty mezer `\leftskip`, `\rightskip` a `\parfillskip`.

makro	(implicitně)	<code>\raggedright</code>	<code>\raggedleft</code>	<code>\centering</code>
<code>\leftskip</code>	<code>\Zeroskip</code>	<code>\Zeroskip</code>	<code>\Flushglue</code>	<code>\Flushglue</code>
<code>\rightskip</code>	<code>\Zeroskip</code>	<code>\Flushglue</code>	<code>\Zeroskip</code>	<code>\Flushglue</code>
<code>\parfillskip</code>	<code>\Flushglue</code>	<code>\Zeroskip</code>	<code>\Zeroskip</code>	<code>\Zeroskip</code>

S uvedenými mezerami si můžeme pohrát a docílit toho, že vnitřní řádky odstavce budou formátovány jinak než první nebo poslední řádek.

Následující makro `\poslednivpravo` zarovná odstavec do bloku a poslední řádek zarovná doprava.

```

13 \newcommand*{\poslednivpravo}{%
14   \leftskip=0pt plus 1fil
15   \rightskip=-\leftskip
16   \parfillskip=\leftskip
17   \parindent=0pt}

```

Tento odstavec není moc zajímavý. Je zarovnaný vlevo i vpravo. Výjimkou je poslední řádek, který je zarovnaný vpravo.

Makro `\posledninastred` zarovná odstavec do bloku a poslední řádek zarovná na střed.

```

18 \newcommand{\posledninastred}{%
19   \leftskip=0pt plus 1fil
20   \rightskip=0pt plus -1fil
21   \parfillskip=0pt plus 2fil\relax}

```

Všechny řádky tohoto odstavce jsou zarovnané vlevo i vpravo. Výjimkou je první řádek, který je odsazený, a poslední řádek, který je zarovnaný na střed.

Pokud nechceme mít první řádek odstavce odsazený, použijeme na začátku textu odstavce `\noindent`.

Makro `\prvnivlevopotomvpravo` nastaví, že první řádek odstavce bude zarovnaný vlevo, zatímco ostatní řádky odstavce budou zarovnány vpravo. Tento tvar odstavce se mi však zdá nepřírozený.

```

22 \newcommand*{\prvnivlevopotomvpravo}{%
23   \leftskip=0pt plus 1fill

```

```

24 \rightskip=0pt plus 1fil
25 \parfillskip=0pt
26 \everypar{\hskip 0pt plus -1fill\relax}%
27 \parindent=0pt}

```

Tento odstavec je podivně zarovnaný. Jeho první řádek je zarovnaný vlevo. Další řádky včetně posledního jsou zarovnané vpravo.

Množství způsobů různých zarovnání odstavce je dáno množstvím kombinací, které lze vytvořit z mezer (Donald Knuth používá pojem „glue“) vkládaných na kraje řádků. Detailní popis datového typu glue je v *T_EXbooku* [5, kap. 12], případně v knize Victora Eijkhouta [3, kap. 8].

V definici makra `\prvnivlevopotomvpravo` má mezera `\leftskip` pružnost `1fill`, což je nekonečněkrát více než pružnost `\rightskip`, která je `1fil`. To znamená, že se řádek zarovná doprava. V posledním řádku odstavce má mezera `\parfillskip` hodnotu `0pt` a to nijak neovlivní konec řádku. První řádek odstavce má pružnost `-1fill` z `\everypar` a pružnost `1fill` z `\leftskip`. Tyto pružnosti se pokrátí a zůstane pružnost `1fil` z `\rightskip`. To znamená, že první řádek se zarovná doleva.

Nikos Platis [6] potřeboval, aby bylo posledních několik slov odstavce zarovnaných doprava a pokud by vpravo nebyl dostatek místa, pak aby se tato slova přesunula na další řádek a zarovнала doprava.

Krátký řádek.	Text vpravo
---------------	-------------

Delší řádek než v prvním případě.	Text vpravo
-----------------------------------	-------------

Existuje více řešení tohoto problému. V Nikosových pokusech se jako nejlepší ukázalo řešení, které zaslal Dirk Schlimm a které původně navrhl Donald Knuth v *T_EXbooku* [5, str. 106]. Po převedení do *L^AT_EXu* uvedené řešení vypadá následovně.

```

28 \newcommand*{\vpravo}[1]{\{%
29 \unskip\nobreak\hfil\penalty50
30 \hskip2em\hbox{\}\nobreak\hfil#1%
31 \parfillskip=0pt\finalhyphendemerits=0\par}}

```

Makro `\vpravo` se používá takto.

```

32 Krátký řádek. \vpravo{Text vpravo}

```

Dalším častým požadavkem je, aby poslední řádek odstavce nebyl moc krátký. Pro tyto účely je možné použít makro `\nemockratky`, které zajistí, že poslední řádek bude alespoň `\minimum` dlouhý.

```

33 \newdimen\delka
34 \newdimen\minimum
35 \minimum=2em
36 \newcommand*{\nemockratky}{%
37   \delka=\hsize
38   \advance\delka -\minimum
39   \leftskip=0pt plus 0pt minus 0pt
40   \rightskip=\leftskip
41   \parfillskip=\delka minus \delka
42   \parindent=2em}

```

Poslední řádek tohoto odstavce by neměl být kratší než délka, kterou si nastavíme.

Poslední řádek tohoto odstavce by neměl být kratší než délka, kterou si předtím nastavíme.

U krátkých odstavců, jako jsou tyto, účinek makra tolik nevynikne. Lépe je to vidět u delších odstavců.

Peter Flynn [4] odpověděl na níže uvedenou otázku uživatele Marka a naprogramoval makra pro zarovnání, které nazval „pružný okraj“.

Ahoj, rád	...a nějaký text tady.
bych měl na	Zkoušel jsem
jednom řádku	tabularx a TabularC ,
text	ale tato prostředí
zarovnaný	nejsou dostatečně
vlevo	přesná na to, aby byl
i vpravo,	text zarovnán až
například	k okraji. Máte nějaký
nějaký text	nápad? Díky, Mark
tady...	

Tento odstavec byl vytvořen makrem

```

43 \pruzny{0.3}{0.5}{Ahoj, rád bych měl...}%
44   {\dots a nějaký text...}

```

kde `\pruzny` je drobně upravené Peterovo makro. První dva argumenty určují poměry celkové šířky řádku pro levý a pravý sloupec. Jejich součet musí být menší než jedna. Zbývající dva argumenty jsou texty, které se mají sázet.

```

45 \newcommand{\pruzny}[4]{%
46   \par\noindent\hbox to\columnwidth{\vtop{%

```

```

47 \hsize=#1\columnwidth\flushleft#3\par}\hss
48 \vtop{\hsize=#2\columnwidth\flushright#4\par}}}
```

Četl jsem právnické dokumenty, kde byl každý odstavec zakončen čarou, aby nebylo možné později připsovat další text. _____

Při sazbě této ukázky byla použita následující makra.

```

49 \let\origpar\par
50 \newcommand*{\parrule}{%
51   \hrule height 2.2pt depth -1.8pt\relax}
52 \newcommand*{\caranakonci}{%
53   \unskip\nobreak\space
54   \leaders\parrule\hskip\Flushglue
55   \vadjust{ }\parfillskip=0pt\origpar}}
```

Pokud nastavíme

```
56 \let\par\caranakonci
```

pak budou všechny další odstavce zakončeny čarou. Je ale třeba být opatrný, protože L^AT_EX pracuje s mnoha věcmi jako s odstavci, takže se můžeme dočkat různých překvapení. Pokud se budeme chtít vrátit k normálně zakončeným odstavcům, napíšeme

```
57 \let\par\origpar
```

Můžeme také využít skupiny.

```

58 \begingroup
59 \let\par\caranakonci
60 Odstavec s čarou na konci...
61
62 Další odstavec s čarou...
63
64 \endgroup
65 Normální odstavec...
```

Běžně se odstavec ukončuje prázdným řádkem nebo příkazem `\par`. Pokud budeme chtít zakončit čarou pouze jediný odstavec, použijeme namísto prázdného řádku makro `\caranakonci`.

3. Ruské formátování odstavců

Procházel jsem staré články, abych protřídil ty, které už nejsou užitečné. Narazil jsem přitom na staré vydání časopisu Baskerville, které obsahovalo článek o sazbě odstavců v ruském stylu [2]. Zdá se, že podle ruské typografické tradice musí být poslední řádek odstavce buď dlouhý alespoň `\parindent` a být ukončený mezerou alespoň `\parindent`, nebo musí být zarovnaný na levý i pravý okraj.

Tento požadavek není možné splnit jednoduchým nastavením parametrů odstavce.

V uvedeném článku byla popsána dvě řešení. První řešení je od Petera Schmitta. Základní princip je zakončit odstavec posloupností `glue + hbox + glue`, kde prázdný `hbox` má šířku `\parindent`, šířka `glue + hbox` se pohybuje od `\parindent` do `\hsize - \parindent` a šířka `hbox + glue` se pohybuje od `\hsize - \parindent` do `\hsize`, přičemž `\hsize` znamená šířku sazby. Podle pravidel \TeX se řádek může zlomit buď před `glue + hbox` nebo ihned za `hbox`. V obou případech takto vznikne další, prázdný řádek odstavce, který se musí odstranit.

```
66 \def\ruskyPS{\ifhmode \unskip
67   \hskip-2\parindent minus -2\parindent
68   \hskip\hsize minus\hsize
69   \hbox{\hskip\parindent}%
70   \hskip0pt \hbox{\strut}%
71   \hskip-\parindent
72   \hskip\hsize plus\parindent
73   \vadjust{\nobreak\vskip-\baselineskip}%
74   \parfillskip0pt
75   \origpar
76   \fi}
```

Jak toto zarovnání vypadá v praxi, je ukázáno v rámečku. Použilo se nastavení

```
77 \parindent=2em
78 \let\par\ruskyPS
```

Druhé řešení pochází od Donalda Arseneaua.

```
79 \def\ruskyDA{\ifhmode \unskip
80   \strut\vadjust{}\nobreak
81   \discretionary{}%
82     {\hbox{\hskip2\parindent
83              \vrule depth 273sp
84              width 0sp height \ht\strutbox}}%
85     {\hbox{\hskip\parindent}}%
86   \hskip-2\parindent minus 2\parindent
```


<p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4 5</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4 5 6</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4 5 6 7</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4 5 6 7 8</p> <p>Poslední řádek tohoto odstavce se sází podle ruské typografické tradice. 0 1 2 3 4 5 6 7 8 9</p>

```

87 \hskip\hsize minus\hsize
88 \kern0pt\parfillskip0pt
89 \origpar
90 \ifdim\prevdepth=273sp
91   \nobreak
92   \vskip-2\baselineskip
93   \hbox{\strut}%
94 \fi
95 \fi}

```

Toto řešení funguje podobně jako Peterovo řešení, na rozdíl od něj se však nemusí vytvořit přebytný prázdný řádek. Makra vloží do textu neviditelnou čáru konkrétní velmi malé hloubky. Pokud se tato čára dostane na další řádek, můžeme ji testem na `\prevdepth` odstranit.

Seznam literatury

- [1] Arseneau, Donald. `shapepar.sty`, 2002. Dostupné na CTAN v adresáři `latex/macros/generic/shapepar`.
- [2] Carlisle, David; Schmitt, Peter. *Russian paragraph shapes*. Baskerville, 6(1):13–15, February 1996.
- [3] Eijkhout, Victor. *T_EX by topic, A T_EXnician's Reference*. Addison-Wesley, 1991. Dostupné na www.wijkhout.net/tbt.

- [4] Flynn, Peter. *Re: simultaneous justification in latex*. Příspěvek ve skupině `comp.text.tex`, 20. 9. 2006.
- [5] Knuth, Donald E. *The T_EXbook*. Addison-Wesley, 1984.
- [6] Platis, Nikos. *Justify at right margin or in next line*. Příspěvek ve skupině `comp.text.tex`, 21. 8. 2006.
- [7] Wilson, Peter. *The hanging package*, April 2004. Dostupné na CTAN v adresáři `latex/macros/contrib/hanging/`.
- [8] Wilson, Peter. Glisterings. *TUGboat*, 28(2):229–232, 2007.

Summary

This paper demonstrates possibilities of setting the T_EX paragraph parameters for different, even less usual, paragraph justifications.

Keywords: L^AT_EX, shapepar package, paragraph, justification.

*Peter Wilson, herries.press@earthlink.net
18912 8th Ave. SW
Normandy Park, WA 98166 USA*

Zpravodaj Československého sdružení uživatelů T_EXu
ISSN 1211-6661 (tištěná verze), ISSN 1213-8185 (online verze)

Vydalo: Československé sdružení uživatelů T_EXu vlastním
nákladem jako interní publikaci
Obálka: Antonín Strejc
Ilustrace na obálce: Petra Rychlá
Počet výtisků: 310
Uzávěrka: 9. 12. 2017
Odpovědný redaktor: Jan Šustek
Redakční rada: Pavel Haluza, Lukáš Novotný, Vít Novotný,
Michal Růžička a Jan Šustek (šéfredaktor)
Technická redakce: Vít Novotný
Evidenční číslo MK: E 7629
Tisk: ASMETI, Klášterní 1187, 735 11 Orlová
Adresa: ČS_{TUG}, Nejedlého 373/1, 638 00 Brno
Email: cstug@cstug.cz

Zřízené poštovní aliasy sdružení ČS_{TUG}:

bulletin@cstug.cz, zpravodaj@cstug.cz
korespondence ohledně Zpravodaje sdružení

board@cstug.cz
korespondence členům výboru

cstug@cstug.cz, president@cstug.cz
korespondence předsedovi sdružení

gacstug@cstug.cz
grantová agentura ČS_{TUGu}

secretary@cstug.cz, orders@cstug.cz
korespondence administrativní síle sdružení, objednávky CD a DVD

cstug-members@cstug.cz
korespondence členům sdružení

cstug-faq@cstug.cz
řešení otázky s odpověďmi navrhované k zařazení do dokumentu ČS_{FAQ}

bookorders@cstug.cz
objednávky tištěné T_EXové literatury na dobírku

ftp server sdružení:
ftp://ftp.cstug.cz

www server sdružení:
http://www.cstug.cz

CONTENTS

Petr Sojka: Introduction	117
Petr Sojka, Vít Novotný: T _E X in Schools? Just Say Yes: The use of T _E X at the Faculty of Informatics, Masaryk University	118
Hans Hagen: LUA in METAPOST	138
Peter Wilson: It Might Work. VI – Paragraphs	155