

OBSAH

DONALD KNUTH — Computer Literacy Bookshops Interview ...	141
Karel Horák: Jak si poradit s velkými METAFONTovými obrázky	154
Karel Dostál: O jedné vraždě a jejím možném odčinění	163
Jan Kybic: Program IMARK: Nástroj usnadňující vytváření rejstříků v $\text{T}_{\text{E}}\text{X}$ u a $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u	174
Jan Kybic: BMV — Jak na PostScript pod Linuxem	179
Ladislav Bittó: Písanie cs textov trochu ináč	181
Zápis z valného shromáždění $\mathcal{C}\mathcal{S}\text{TUG}$ u	188

Z elektronické konference $\mathcal{C}\mathcal{S}\text{T}_{\text{E}}\text{X}$ u

(otištěno bez vědomí autora Petra Novotného)

$\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ vs plain. Je to těžké. Já $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ (většinou) používám (a to verzi 2.09). Přesto k němu mám spoustu výhrad. Je velice ochuzen `tabular`, nemá ani zlomek potenciálu `\halign`'u (jak uděláte `\halign to\hsize` v $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u?)... S environment listem zápasím odjakživa. Ve skipech se prostě nevyznám (a to mám Lamportův manuál!), čas od času tomu chybí `\item`... Občas si musím pomoci tak, že něco vysázím do `\box<n>`, a ten pak používám pro další operace... Při sazbě oboustranného tisku prostě neudržím tiskové zrcadlo (říká se tomu takhle nějak?) — prostě aby mi na průsvitu řádky lícovaly (a byly stejně daleko od sebe atd.) — musí se řešit fůra „gumových“ mezer.

Je pravda že v $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ u se píše mnohem snáz — ale vylézá z toho celkem nudný sterilní tisk občas okořeněný nekorektním `\hskip`'em nebo `\vskip`'em... Proti plainu to je krok směrem k Venturám a WordPerfectům — tedy k prefabrikovaným nástrojům, kde rozhodující je, jak si to představoval autor systému. (A neříkejte mi, že to jde změnit. Jde, ale člověk už toho musí umět fakt dost.)

Použiji s vašim svolením (nedokonalou) analogii. Když si chci dát večeři, tak si buď skočím do hospody (PageMaker, Ventura), kde si dám to, co mají; nebo si koupím nějaký polotovar a dotvořím v mikrovlnce ($\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$); anebo si nakoupím suroviny a dám si tu práci, že si uvařím sám (plain). Samozřejmě poslední alternativu zvolím, budu-li mít nějaké hosty, na kterých mi bude fakt záležet. Ale normálně, když o nic nejde, si klidně můžu dát housku se salámem z ubrousku (T602). Doufám, že jsem nikoho neurazil.

DONALD KNUTH — **Computer Literacy**
Bookshops Interview

*Interview vedený firmou Computer Literacy Bookshop, Inc., dne 7. prosince 1993**

Donald E. Knuth patří podle mnohých do světové špičky v *computer science*. První tři svazky díla *The Art of Computer Programming*, které ani za 30 let neztratily na významu, mu přinesly Turingovu cenu ACM v roce 1974 a národní medaili za vědu v roce 1979. Knuth také vyvinul převratné aplikace v počítačové sazbě (T_EX a METAFONT) a vývoji software (WEB) a má na svém kontě více než 100 publikací.

Nyní, kdy je emeritním profesorem Stanfordovy univerzity, vrhl Knuth opět veškerou svou energii do psaní. Dan Doernberg se jej v tomto rozhovoru z prosince 1993 ptal na to, co v poslední době dělal a na co se můžeme těšit.

CLB: Právě jste publikoval knihy jak o systému CWEB, tak i o *Stanford GraphBase*, dvou oblastech vašeho vlastního výzkumu. Začněme CWEBem, který je spojením jazyka C a T_EXu a slouží k dokumentování programů.

Knuth: Systém CWEB je nadstavbou C, a je to zdaleka nejlepší na světě známá programovací metoda. Musím být prostě upřímný a říci, že je to senzační věc. Kniha *CWEB System of Structured Documentation* je definitivní uživatelskou příručkou a vysvětluje toho víc, než může kdokoliv potřebovat.

CLB: Řekl jste, že CWEB zvyšuje řádově produktivitu programátora — jak to?

Knuth: No, možná ne řádově, možná jen dvojnásobně. Lidé, kteří používali CWEB, zjistili, že píší lepší programy, které jsou přenositelnější, lze je snáze ladit, udržovat... a dají se také rychleji napsat.

CLB: Používá se CWEB jen ve Stanfordu, anebo také v průmyslu?

* Originální verzi rozhovoru lze získat z WWW
<http://www.clbooks.com/nbb/knuth.html>.

Knuth: Používá se na celém světě. Na řadě systémů jsme měli WEB, původní verzi pro Pascal, a řada lidí, kteří se jím infikovali, se pak stále rozrůstala. \TeX byl napsán ve WEBu. Silvio Levy v roce 1987 provedl konverzi na CWEB. Ten byl dlouho experimentální, ale teď mohu říci „Experiment vyšel!“ CWEB je mnohem lepší než WEB, protože C je mnohem hezčí jazyk pro systémové programování a řadu jiných věcí. Nenapadá mne, proč by měl někdo, kdo se opravdu zajímá o programování, tímto pohrdnout ve prospěch jiného systému.

CLB: Snadno použitelné, rychlé, všechno bezvadné?

Knuth: Správně, a taky cítíte po napsání programu ohromné uspokojení!

CLB: I když napíšete špatný program?

Knuth: Skoro... no... jo! Zeptejte se Jill, jak párkrát za týden vyjdu ze své pracovny a říkám „To programování v CWEBu je taková zábava!“ Je to fakt, nemůžu se toho nabažit.

Duševní rozpoložení, v němž se ocitáte, když píšete program v CWEBu, je o hodně lepší než starý přístup. Představujete si, že píšete pro lidskou bytost, lidské bytosti vysvětlujete, co by měl počítač udělat, a ne že jste jen v roli toho, kdo má počítači nějak sdělit, co má dělat. Máte-li to ale vysvětlit jiné osobě, tak si to musíte sám daleko lépe sesumírovat. Tento postup pomáhá i v případě programu, který za hodinu zahodíte. CWEB je prostředek, který doporučuji používat, i když třeba píšete program jen pro sebe, pro své oči.

CLB: CWEB se zdá být hodně blízko modelům strukturovaného programování v 70. letech...

Knuth: Správně, jde o další krok. U strukturovaného programování někteří radili vytvářet programy shora dolů, a jiní zase zdola nahoru. S WEBem nebo CWEBem můžete kus udělat zdola nahoru a kus shora dolů, podle toho, co vám přijde vhodné z hlediska programu nebo té části, v níž zrovna jste.

Metodologie strukturovaného programování byla výborná... jenže k opravdovému pochopení ji nesmíte chápat jako kuchařku s předpisy, ale jako způsob porozumění vztahu mezi vyšší a nižší perspektivou pohledu na program. Uděláte to tak, že na celý program nahlížíte jako na síť, spoustu malých kousků, které jsou samy jednoduché a mají jednoduchá propojení s jinými malými kousky. Schéma WEBu právě podporuje tento způsob chápání složitého celku pomocí jednoduchých malých částí a propojení mezi nimi.

Ty části můžete vytvářet v takovém pořadí, které vám psychologicky nejvíce vyhovuje. Někdy je můžete udělat zdola nahoru. Zdola nahoru znamená, že nějak tušíte, že se asi bude pro něco hodit určitý podprogram, takže ho napíšete hned, když jste na to připraven nebo dokud máte náladu. Při tomto programování zdola nahoru se vaše tužka stává stránku od stránky mocnější, protože na straně devět vyvinete nástroje, které můžete použít na straně deset. . . vaše tužka je silnější.

Pokud jde o programování shora dolů, tam začnete zkraje a řeknete si „Tohle napíšu napřed a pak toto, a potom toto. . .“, ale později je na vás, abyste podrobně zorepsal, oč v každé části běží — může být, že po sto stránkách budete lapat po dechu, když konečně přijdete na to, jak to vlastně máte všechno udělat!

Na několika prvních stranách vypadá programování shora dolů obyčejně velmi pěkně, ale pak je čím dál těžší udržet všechny nitky v chodu. Programování zdola nahoru také často vypadá nějakou dobu pěkně, vaše tužka sílí, ale to právě zas svádí k různým trikům. Pokud oba přístupy nějak psychologicky vhodně zkombinujete, pak to funguje, dokonce i když končíte.

Takhle jsem to dělal s $\text{T}_{\text{E}}\text{X}$ em, což je hodně velký program: víc než 500 stran zdrojového kódu v knize. V celém tom programu, u každé jeho řádky byla vždy jen jediná věc, kterou jsem musel udělat jako příští. Neměl jsem žádný velký výběr; každý krok se zakládal na tom, co už jsem do té doby udělal. Žádná metodologie by mne nenaučila, jak psát software tohoto druhu, pokud bych se jí přísně držel. Ale když jsem si představil, že ten program vysvětluji dobrému, schopnému programátorovi, tedy všechno, co v něm je, našel jsem jen jednu přirozenou cestu, jak to udělat. Uspořádání, v němž se kód nachází v knize, odpovídá tomu, jak jsem jej psal.

CLB: Do jaké míry jste sledoval nebo sledujete ony „svaté války“ — debaty o metodologiích softwareového inženýrství?

Knuth: Nesledoval jsem všechny nuance těchto věcí, ale dominantních myšlenek jsem si všiml. Až za mnoho let po jiných lidech jsem vlastně zjistil, co jsou to CASE nástroje. Myslím si, že nebylo dobře dělat z toho takové náboženství. V oněch dobách se způsob programování řídil „politickou korektností“.

Podobná věc se stala v matematickém společenství ve dvacátých letech, kdy se říkalo, že dobrý matematik musí dokazovat věty určitým způsobem. Nehodilo se používat jisté prostředky důkazu, o nichž někteří soudili, že vás mohou zavést do paradoxů. Bylo to něco jako pěstování

matematiky s jednou rukou uvázanou za zády. Politicky korektní strukturované programování podobným způsobem překázelo lidem v napsání dobrých programů, i když přesně věděli, co dělají, a to jen proto, že jejich postup zrovna nezapadal do současného chápání správnosti. *Computer science* je jako každý jiný obor; postupuje v módních vlnách. Některé z trendů jsou dobré, jenže skoro s každou dobrou myšlenkou se tak nějak zachází jinak, než jak by to mělo být.

Například si vezměte generátory náhodných čísel. Patnáct let jsme neměli žádnou teorii o tom, jak generovat náhodná čísla. Pak někdo dokázal takovou drobnost o jedné z metod: když se vzal průměr sériových korelací přes celou periodu čítající miliardu čísel, pak ten průměr byl nula, což bylo dobré. Z ničeho nic všichni všeho nechali, vyhodili své staré rutiny a přešli k této nové metodě, poněvadž to byla jediná metoda, k níž byla vůbec nějaká teorie. Ukázalo se, že to je hrozný generátor náhodných čísel; teorii ušlo, že průměr přes první polovinu je +1 a přes druhou polovinu -1! Vždy v celé historii si lidé vybírali metody a nedošla jim jejich omezení.

CLB: Co to bylo za metodu?

Knuth: No, ve většině knihoven podprogramů se jí říkalo RANDU. Touto dobou už by měla být dostatečně vymýcena; nicméně, jak někdo uvidí podprogram s názvem RANDU, pryč s ním!

CLB: Vedlo vás ke spojení WEBu s Céčkem to, že je dnes tolik programátorů používá, anebo se vám osobně C líbí a píšete v něm?

Knuth: Myslím si, že C má spoustu velmi důležitých vlastností. Například způsob, jímž C zachází se směrnicí, byl znamenitou inovací; vyřešil řadu problémů se strukturováním dat, s nimiž jsme se dříve potýkali, a vylepšil také vzhled výsledných programů. C ovšem není dokonalý jazyk, nakonec dokonalý není žádný z existujících jazyků, ale má podle mého spoustu kladů, a když se vám něco nelíbí, tak se tomu můžete vyhnout. Takže C jako jazyk se mi určitě líbí, zvláště proto, že je dobře sladěno s operačním systémem (pokud například používáte UNIX).

Po celý život si vždy vybírám ten jazyk, který je nejlépe sladěn s debuggerem a operačním systémem, které zrovna používám. Pokud bych měl pro nějaký jazyk X lepší debugger a X se dobře snášel s operačním systémem, používal bych tento jazyk.

Extrémní případ nastal, když jsem rok pracoval v laboratoři, kde operační systém navrhl Ned Irons. Byl to systém pro jednoho z raných Crayů a Irons také napsal kompilátor pro jazyk zvaný IMP. IMP měl řadu hrozných vlastností. Jednak to byl rozšiřitelný jazyk, a tak jej všichni neustále

rozšiřovali. Program, který v pondělí fungoval, vám v úterý fungovat přestal, čili po zřícení programu se vždycky muselo napřed zkontrolovat, zda je v pořádku strojový kód programu. Za druhé, IMP byl jazykem ohromně zhuštěným. Kupříkladu tam, kde v Pascalu píšete „if X>0 then ...“, v IMP řeknete „X+=>“. Jinými slovy, programy byly velmi krátké. Připadalo vám sice, že píšete elegantní programy, protože měly jen pár znaků, jenže druhý den už jste je nemohl vůbec přecíst! Cenou za úspornost tak bylo, že jste z té změti značek na stránce byl docela vedle...

CLB: V současné době zřejmě kladete důraz na „gramotné programování“, ale přitahoval vás někdy APL coby matematicky orientovaný jazyk?

Knuth: To je něco jiného. APL je pro lidi, kteří mají problémy k řešení a nestarají se tolik o efektivitu; chtějí pěkně a elegantně zformulovat řešení svého problému, i když toto řešení nemusí být nutně snadnou záležitostí pro počítač. Je to jazyk pro specifikaci problémů, nikoliv jazyk pro systémové programování... APL-WEB existuje.

Chtěl bych ale ještě něco říci k IMP. Třetí věcí proti němu bylo to, že v případě chyby se buď kompilátor dostal do nekonečné smyčky, anebo se zastavil na první chybě, oznámil „ERROR ERROR ERROR“ a skončil. Sám jste musel přijít na to, v čem je chyba. Žádný skvělý jazyk nebo kompilátor to tedy nebyl.

Nicméně... stejně jsem si tenhle jazyk zvolil, protože bezvadně zapadal do onoho operačního systému. Názvy polí se volily tak, že byly snadno vidět v debuggeru, a mohl jste zjistit, jakým způsobem se přiděluje paměť, věděl jste, co se děje, a konec konců jste mohl program zvládnout tak, aby běžel spolehlivě, poněvadž IMP byl sladěn s operačním systémem. Mohl jste psát v lepším jazyku, ale odpovědi byste jen tak nezískal a celá práce by vám trvala o pár týdnů déle. Já jsem používal IMP.

CLB: Používal se IMP ve Stanfordu?

Knuth: Tohle bylo ve výzkumné laboratoři v Princetonu. Rok před mým příchodem do Stanfordu jsem tam dělal na výzkumném projektu klasifikované analýzy šifer.

CLB: Řekněte nám, prosím, o vaší druhé knize, která se týká stanfordské *GraphBase*.

Knuth: Kniha o *GraphBase* je určena pro dva druhy lidí. Jeden její účel je výzkumný; lidé, zabývající se studiem algoritmů pro kombinatorické problémy, potřebují standardní sadu zkušebních dat, na nichž se mohou navzájem poměřovat, a také jako výkonnostní normu. Během přípravy

svazku IV řady *The Art of Computer Programming* jsem se rozhodl, že zpřístupním všem příklady a data, které používám. Byla tu potřeba standardních výkonnostních norem, a taky by to všechno mělo být uspořádáno tak, aby se to mohlo použít tisícem různých způsobů. Takže... mám teď sbírku tisíců standardních datových sad; kdokoli v Polsku může mít data přesně stejná, a zrovna tak kdokoli v Kalifornii nebo v Číně. Je to velmi přenositelné a lze to stáhnout z Internetu.

Druhým důvodem pro odbočku ke *GraphBase* je to, že jde o příklad programování v *CWEBu* — je to vlastně 32 příkladů programování v *CWEBu*. Jsou to krátké programy ilustrující ten styl programování, jež já dávám přednost. Příklady jsou cosi jako drobné eseje, kratičká vyprávění formou počítačových programů, které se snad hezky čtou.

CLB: Jaké je vaše současné hardwareové a softwareové prostředí?

Knuth: Pro své programování používám *CWEB*, velice využívám editor *Emacs*, a pro kreslení technických ilustrací používám výtečný jazyk vysoké úrovně, který se nazývá *META O T*. Je to nový jazyk, autorem je John Hobby a myslím, že by měl být už brzy zveřejněn. Je založen na *METAFONTu*. 75% kódu pochází ode mne z *METAFONTu*, ale je to upraveno tak, že z toho vychází *PostScript*. Moc se mi líbí.

Také používám program *Mathematica*. Lidé od *Maple* se mě pokoušejí přesvědčit, abych přešel k *Maple*, jinému výtečnému systému. Momentálně se mi zamlouvá *Mathematica*, poněvadž nemusíte psát značky násobení; stačí říct „2X“ místo „2*X“. Taktéž má *Mathematica* výjimečně dobrý manuál.

CLB: Takže se vám líbí *Wolframův* styl psaní?

Knuth: Zejména rejstřík... V té knize se jen tak neztratíte. U prvního vydání jsem se díval do rejstříku, když jsem měl řešit nový problém, a skoro vždycky mě odkázal na správnou stránku. Snad ve třech nebo čtyřech případech tam hledané slovo nebylo, což jsem si vždy v knize označil tužkou, až se k tomu dostanu příště. V druhém vydání to už bylo všechno opraveno, a to jsem to ani nikomu nehlásil.

CLB: Rád bych se zeptal na vaše letmé dojmy o několika výzkumných směrech a také zda jste v nich něco studoval nebo udělal. První z nich jsou genetické algoritmy. Co říkáte té obecné myšlence, že algoritmus nemusí určovat člověk, ale může se nechat na stroji...

Knuth: Plánuji s tím řadu experimentů, jakmile se dostanu ke svazku IV. Máme tu genetické šlechtění, pak simulované žhání (*simulated annealing*), a lidé vymysleli ještě jiné strategie. V knize o stanfordské *GraphBase* mám metodu, které říkám „stratifikovaná lačnost“. Všechny tyto

postupy soupeřů o stejný druh problémů a já bych rád zkusil hodně příkladů; některé mohou být lepší tady než jinde, takže bych pro to rád získal nějaký cit. Určité problémy jsou jako stvořené pro neuronové sítě. . . genetické algoritmy budou nejspíš vycházet dobře v úlohách kolem rozpoznávání řeči a prý také v odhadech burzovních pohybů nebo v něčem takovém. Jaksi čím je problém blíže přírodě, tím více lze počítat s tím, že genetický algoritmus bude fungovat, zatímco čím je bližší teorii čísel nebo něčemu umělému, tím spíše se dá čekat pomoc od nějakého jiného přístupu. Je dost těžké porozumět vlivu velikosti úlohy na tyto metody; na malých úlohách mohou běžet božsky a pak najednou zcela zkrachují, když se úloha jen trošičku zvětší. . . anebo to může být obráceně.

CLB: Vypadá to tak, jako byste měl před sebou pár let disciplinovaného testování na vašich sadách dat.

Knuth: Stanfordská *GraphBase* mi dává neomezený zdroj úloh, na nichž mohu dělat já i jiní. Pročítám si, co jiní lidé prohlásili o svých metodách, ale také si je všechny zkuším. Můj původní přínos v *The Art of Computer Programming* spočívá v tom, že vezmu metody od dvou různých autorů a analyzuji metodu A z pozice autora B a metodu B z pozice autora A. Každý to popsal jen ze své strany, tak zkuším vyplnit mezery. . .

CLB: A co objektově orientované programování? Je to jen módní heslo anebo se vám opravdu zamlouvá?

Knuth: O programování jsem vždycky přemýšlel v tomto duchu, ale nepoužíval jsem jazyky, které pomáhají dodržovat disciplínu. K disciplíně jsem se vždy přinutil sám i v jiných jazycích. Programovací jazyky vás teď mohou chytit, když uděláte chybu, a ulehčují vám omezovat viditelnost informací mezi jednotlivými částmi programu navzájem. Ve svých programech napsaných ve starších jazycích jsem nepoužíval to, co jsem používat neměl; musel jsem se sám ukázněvat, abych se těmito pravidly řídil. Bylo to možné, a tak jsem to dělal. Nešlo o to, že bych nějaké programy nemohl napsat. . . ale ty nové prostředky určitě pomáhají.

Problém, který s nimi dneska mám, je v tom, že. . . C++ je moc komplikované. Momentálně je pro mne nemožné psát přenositelný kód, o němž bych věděl, že poběží na mnoha různých systémech, jedině snad pokud bych se vyhnul všem exotickým prvkům. Kdykoli se tvůrcům jazyka C++ naskytly dva konkurenční postupy pro řešení nějakého problému, řekli si „OK, dáme je tam oba“. Tím pádem se jazyk stal pro

moje gusto příliš barokním. Každý uživatel C++ si ale stejně vybírá jen nějakou oblíbenou podmnožinu, takže se nic neděje. CWEB plně podporuje C++ zrovna tak jako C.

CLB: Co si myslíte o takových oborech, jako je teorie chaosu a fraktály? Nedeterminovanost v nich obsažená se zdá být poněkud v rozporu s oblastmi, na něž jste se v minulosti soustředil.

Knuth: Zpočátku jsem se trochu zabýval fraktály a podobnými věcmi a zdá se mi, že je to vynikající nová abstrakce. Lidi mohou budovat modely, které by je dříve ani nenapadly a které přitom skutečně vystihují řadu věcí v přírodě, majících tu vlastnost, že vypadají stejně i při změně měřítká. Jak známo, když zvětšujete pobřežní čáru, vypadá stále jako pobřežní čára, a spousta jiných věcí má tento charakter. Příroda disponuje rekurzivními algoritmy, které používá pro vytváření mraků a ementálu a tak. Takže teď máme matematické postupy, s jejichž pomocí můžeme porozumět procesům, které sahají za hranice diferenciálních rovnic, na něž byli lidé zvyklí v předchozích staletích. Dostali jsme do rukou úplně nový nástroj, ale moje intuice mi toho o těchto metodách mnoho neříká. Jsem si vědom omezenosti své intuice; dovedu některé problémy dobře vyřešit, ale znám jiné lidi, kteří z fleku vyřeší něco, co mně sebere spoustu času. . . . Není to můj šálek čaje.

CLB: Do jaké míry jste kdy sledoval rozvoj umělé inteligence? Váš třetí program v životě byl „kámen–nůžky–papír“, který se učil z vlastních chyb, a Stanford je odjakživa jednou z předních institucí přes výzkum umělé inteligence. . . .

Knuth: No, umělá inteligence hodně souvisí se svazkem IV; badatelé v tomto oboru užívají kombinatorické postupy, které já studuji, takže vím o spoustě relevantní literatury. Mým úkolem je porovnání literatury o umělé inteligenci, která vyšla z kruhů elektrotechnického inženýrství, s tou, která pochází z jiných disciplín; každá komunita přistupuje k problémům trochu odlišně. Zkouším tyhle věci číst, očistit je od žargonu a sjednotit myšlenky. Z umělé inteligence pocházely odjakživa ty nejtěžší aplikace a nejatraktivnější problémy v *computer science* — umělá inteligence byla nejplodnějším zdrojem metod v *computer science*. Vedla k řadě důležitých zlepšení, jako jsou datové struktury a práce se seznamy. . . . umělá inteligence toto ohromně stimulovala. Mnohá z nejlepších paradigmat pro ladění a vývoj programů, všechny systémy počítačové algebry, rané studie počítačové grafiky a počítačového vidění, atd., to vše mělo velmi silné kořeny v umělé inteligenci.

CLB: Nepatříte tedy mezi ty, kdož bagatelizují vše, co bylo v této oblasti uděláno. . .

Knuth: Ne, kdepak. Stalo se totiž to, že spousta lidí věřila, že umělá inteligence se stane všelékem. To máte, jako když nějaká firma udělá jen 15 % zisku, zatímco jí analytici předpovídali 18 %, a její kurs hned spadne. Bylo to jen nesplněné očekávání, které vzniklo z nabubřelých představ, že jedno paradigma vše vyřeší. Je to tak asi se všemi věcmi, které jsou teď na výsluní; lidi si časem uvědomí, že nejsou vyčerpávající odpovědi. Mnohé problémy jsou tak těžké, že se nám nikdy nepovede najít opravdu velkolepé řešení. Lidé jsou zklamaní, když nenajdou elixír mládí. . .

CLB: Kdybyste měl těsně před promoci na univerzitě nebo byl doktorem a neměl žádnou „zátěž“, jaký druh výzkumu byste chtěl dělat? A vybral byste si vůbec znovu výzkum?

Knuth: Myslím, že nejbáječnější počítačový výzkum se teď dělá zčásti v robotice a zčásti v biochemických aplikacích. Robotika, kupříkladu, to je něco fantastického. Dělat stroje, které se skutečně pohybují a navzájem komunikují. Stanford má teď velkou laboratoř robotiky a naším plánem je umístit do nové budovy stovku robotů, kteří budou kráčet po chodbách. Chceme tím stimulovat studenty. Do nové budovy se nastěhujeme tak za dva za tři roky. Jenom to, že tam ty roboty uvidíte, vás přivede k hezkým projektům. Z takových projektů též vznikají mnohé dobré matematické a teoretické otázky. A grafické nástroje vysoké úrovně, tady je taky spousta skvělých věcí. Jo, to by se mi moc líbilo dělat. . . máme jen jeden život, to se ví, ale stejně. . .

CLB: Proč jste se zmínil o biochemii?

Knuth: Jsou tam milióny a milióny nevyřešených problémů. Biologie je tak digitální a přitom tak neskutečně složitá. Průšvih s biologií je ten, že práce biologa je nezáživná. Vaše experimenty vás stojí tři roky práce a pak během jedné noci vypnou proud a všechno vám to chcípne! Musíte začít znova. V počítači si můžeme vytvořit vlastní světy. Biologové zaslouží velký obdiv za to, že se tím dokáží prokousat.

Těžko mohu zodpovědně prohlásit, že po padesáti dalších letech explozivního růstu *computer science* tu ještě zbude hodně fascinujících nevyřešených problémů, že se nebude do značné míry pouze pracovat na drobných vylepšeních dobře prozkoumaných věcí. Může být, že všechny elementární věci a všechny opravdu dobré věci už budou v té době objeveny. Možná, že ne, ale rozhodně nemohu předpokládat růst bez konce. U *computer science* si nemohu být tak jistý, jako jsem si jist u biologie.

Ta má před sebou 500 let práce na vzrušujících problémech, v takovém je to stadiu.

CLB: Používání Internetu se teď bouřlivě rozvíjí, každý se připojuje. . .

Knuth: Jednoho dne taky zkusíme zjistit, kdo to vlastně všechno platí!

CLB: Vy sám jej teď používáte? Víím, že tomu tak bylo v minulosti.

Knuth: Po dobu patnácti let jsem používal elektronickou poštu na ARPANETu a Internetu. V lednu 1990 jsem s tím nakonec přestal, protože mě přehrabování toho smetí připravovalo o příliš mnoho času. Nemám emailovou adresu. Lidé, kteří mi zkoušejí posílat nevyžádané elektronické zprávy, dostávají slušnou odpověď „Profesor Knuth skončil se čtením elektronické pošty; můžete mu napsat na takovou a takovou adresu.“

Email se nedá vypnout! Pošlete někomu zprávu, ten ji pošle s poděkováním zpět a vy mu zas odepíšete „Děkuji za vaše poděkování. . .“

Email je báječný pro určitý druh lidí, je absolutně nepostradatelný pro jejich povolání a umožňuje jim dělat jejich práci lépe. Řekl bych, že pro lidi, jejichž úkolem je být na vrcholu dění, je email skvělý. Avšak mou rolí je sledovat dění zespodu. Sleduji různé myšlenky, pozorně o nich přemýšlím a snažím se je přepsat načisto. . . Probírám se zvolna věcmi již dříve udělanými a zkusím tento materiál zorganizovat. Ale nemám ponětí o tom, co se děje tento měsíc.

Takže teď nečtu elektronickou poštu, ale občas ji stejně používám. Řekněme když cestuji do Izraele a musím něco zařídit na poslední chvíli. Když navštívím na pár dnů jinou univerzitu nebo výzkumné středisko, musím odtud poslat email. Naučil jsem se používat emailové prostředky v Emacsu, ale nechci se v tom stát přeborníkem.

CLB: Máte spoustu zájmů nesouvisejících s počítači a matematikou — hudbu, náboženství, literární tvorbu. Je pro vás hudba tvůrčím ventilem, prostředkem pro rekreaci, nebo duchovním ventilem?

Knuth: Momentálně jde o rekreaci. Rád si zvu domů přátele, s nimiž hraje čtyřruční klavírní skladby. Kdybych to mohl dělat každý týden, tak bych to dělal. Doufám, že budu žít dost dlouho na to, abych po ukončení svého životního díla *The Art of Computer Programming* mohl komponovat hudbu. To je jen sen. . . ovšemže by to třeba byla mizerná hudba.

CLB: Už jste ale nějaké skladby napsal, že?

Knuth: Ano, ale byla to většinou aranžmá motivů jiných lidí. Tedy, když jsem byl na vyšší škole, napsal jsem krátkou hudební komedii nazvanou „Říše Nešťastníků“. Vzpomínáte si, jakým hitem byli Nešťastníci

(*Nebbishes*) koncem padesátých let? „Říše Nešťastníků“ byla jen asi desetiminutovou parodií, ale mělo to všechno původní hudbu a libreto.

CLB: Máte ty noty ještě někde na půdě?

Knuth: Jo. . . vlastně ne, myslím, že jsem je ztratil. Mám jich jen část. Doufám, že na ně znovu narazím. Procházím teď své složky a dělám si počítačový rejstřík všeho, co doma mám.

CLB: Nezdá se, že byste měl domácnost bez papírů!

Knuth: Kdepak!

CLB: Zkoušel jste hudebně počítačovou technologii MIDI, anebo jste se od ní záměrně držel stranou?

Knuth: Baví mě to. Minulé vánoce jsem koupil pro svého syna syntezátor a hrál jsem na něj sám dlouhé hodiny. Moc se mi to líbilo. Kdysi před lety, bylo to v domě Marvina Minskyho, jsem hrál na syntezátor Kurzweil, což byla imitace koncertního křídla. Ne tak dávno odjel jeden můj přítel na tři roky do Anglie a nechtělo se mu brát s sebou své křídlo, a tak si koupil šestihlasou Yamahu. Když jsem ho navštívil, mohl jsem po tři nádherné dny procházet všechny kusy, které jsem se na piano naučil, a hrál jsem je jakoby na vibrafon, nebo na cemballo, nebo jinými hlasy. Jeho „piano“ má zvuk cemballa, ale klávesnice je citlivá na sílu stisku, takže můžete hrát nahlas a potichu, což na opravdovém cemballu nejde. Tyhle syntezátory jsou skvělé.

CLB: Kdy jste odešel ze Stanfordu do výslužby?

Knuth: Letos. Byl jsem na dvouroční dovolené, než jsem mohl oficiálně odejít do výslužby. Neoficiálně jsem odešel v roce 1990, ve stejný den, kdy jsem se vzdal emailu. Své plány jsem ohlásil ještě o tři roky dříve. Ujasnil jsem si, že mým hlavním životním cílem je dokončení *The Art of Computer Programming*; promyslel jsem to do budoucnosti a vyšlo mi, že by to chtělo dvacet let práce na plný úvazek. Kdybych pokračoval ve všem ostatním, co jsem tehdy dělal, stalo by se z toho čtyřicet nebo padesát let práce. Nemohl jsem se nikam pohnout, měl jsem čím dál větší zpoždění. Tak jsem řekl „dost“. Přirozeně mi bylo proti srsti vzdát se mnoha z těch věcí, které dělám s velkým potěšením. Ale některých věcí jsem se milerád zbavil, například psaní návrhů. Těch jsem se vzdal s nadšením!

CLB: Vy jste musel psát nějaké návrhy?? Domníval jsem se, že jste od toho byl nějak izolován.

Knuth: Máte obdivuhodný smysl pro humor! Teď už to nemusím dělat; ale jako profesor jsem musel najít sponzory, když jsem chtěl zajistit

pro své studenty slušné vybavení, nebo přijímat hosty v rámci aktivních výzkumných programů, publikovat zprávy, atd. Žebrání o peníze stojí spoustu úsilí. System Development Foundation mi nabídla milión dolarů, abych mohl dodělat \TeX a vrátit se k *The Art of Computer Programming*.

CLB: Vzal jste ty peníze?

Knuth: Jistě, ale dokončení \TeX u stejně ještě trvalo mnoho, mnoho let. Řekl jsem si, že jediná naděje na dokončení *The Art of Computer Programming* spočívá v přechodu na psaní na plný úvazek, na život poustevníka a v tom, že začnu lidem říkat „ne“. V prvních letech to šlo dost těžko. Teď už si připadám opravdu výkonný a psaní jde dobře. Příjemný ustálený stav.

Tak jednou za měsíc mám přednášku ve Stanfordu, když se tam dostanu. Nazývá se „Počítačové hrátky“. Chci v tom pokračovat dvacet let a každý měsíc pohovořit o něčem, co mne zrovna zaujalo, o pěkných myšlenkách, které jsem posbíral. . . Vytahuji problémy, s nimiž si nevím rady, aby je někdo udělal za mne. Když teď nedokážu pohnout s problémem za dvě hodiny, nezbyvá mi než toho nechat a požádat někoho, aby se tomu věnoval; jinak budu zase zaostávat. Jak píšu knihu, musím přecházet od námětu k námětu a jednomu každému věnuji pozornost tak tři týdny.

CLB: Jste ponejvíce znám svými publikacemi a výzkumem; líbilo se vám učit a být ve styku se studenty?

Knuth: Měli jsme nejlepší studenty na světě. Se studenty se stále setkávám při svém přednáškovém cyklu, až na to, že už neznám jejich jména. To je potíž.

CLB: Nemáte žádné studentské praktikanty?

Knuth: Představte si, že při „Počítačových hrátkách“ nadhodím otevřený problém, a řekněme, že mezi posluchači se najde nějaký student, který ten problém vyřeší, udělá z toho disertaci a tu během příštích dvou (nebo snad dvou a půl) týdnů dokončí a ukáže mi ji. V takovém případě by mě to ještě zajímalo, ještě bych si to přečetl a s radostí bych se mu podepsal pod obhajobu. . . , ale to je jediný možný způsob. Dvacet osm je celkový počet doktorandů, kteří u mne promovali, a více jich asi mít nebudu. . . jedině, když něco bleskově vznikne v rámci mých „Počítačových hrátek“.

CLB: Doktorské práce v reálném čase! Jakých změn jste si všiml během těch let u studentů přicházejících na obor *computer science*?

Knuth: Došlo k velké změně, kterou si neumím vysvětlit. V sedmdesátých letech se většina našich studentů velice zajímala o hudbu. Ptávali jsme se jich hned, jak přišli: „Na jaký nástroj hrajete?“ Měli jsme spousty komorních seskupení a podobně. Dneska skoro nikoho z našich studentů hudba nezajímá. Nevím, jestli je to proto, že se na *computer science* zapisuje jiný typ lidí, anebo zda to platí pro všechny dnešní studenty, nebo co. Když se dnes zeptáte studentů *computer science*, jakého mají koníčka, nejspíš vám většina z nich řekne „cyklistiku“. Nedávno jsem měl jednoho, který hrál na foukací harmoniku, ale jinak v té skupině nebyli skoro žádní muzikanti.

CLB: Nějaké změny v úrovni studentů?

Knuth: V úrovni ne... Ale o matematice toho ví méně, než bývalo dříve zvykem. Musíme se tak více věnovat doškolování, dokonce i na takové škole, jakou je Stanford.

CLB: A co změny v oboru samotném... při takovém pokroku a počtu zúčastněných lidí, liší se dnešní *computer science* hodně od té dřívější?

Knuth: No, jsou tu všechna ta média a vizuální věci, to je velký rozdíl proti dřívějšímu. Také je tu konkurence; je to teď o hodně obtížnější, než to bylo za mých časů. Když jsem začínal, dalo se přijít s něčím novým daleko snáze než teď, kdy tu máte tisíce a tisíce chytrých lidí, kteří všichni dělají skvělé věci. Onehdy mohlo být takových deset výborných doktorských disertací za rok; dneska prostě není šance držet s tím vším krok.

Ať jste v kterémkoli oboru *computer science*, ukazuje se všeobecně, že se těžko drží krok. Jednotlivé obory se stále zužují, protože nikdo už není schopen pokrýt celé území. Každý si musí vybrat dvě malé části *computer science* a nastudovat je; jestliže jeden zná části A a B, další B a C a jiný zas C a D, zůstává obor rozumně dobře propojen, i přesto, že se stále rozšiřuje.

CLB: Nahlížíte na sebe jako na jednoho z posledních „renesančních lidí“ v oboru *computer science*?

Knuth: Nemám tak široký záběr, jak byste si mohl myslet — nepracuji na více věcech zároveň. Řekl bych, že mám schopnost rychle studovat; umím se v momentu stát expertem přes něco. Po dobu třiceti let sbírám materiál, takže mohu číst literaturu o jakémkoli tématu v „dávkovém režimu“ — nemusím přeskakovat mezi mnoha různými tématy. Dokážu absorbovat daný předmět lokálně a být v tom chvilku dobrý... ale nechtějte pak po mně, abych předvedl to, čím jsem se zabýval před pár měsíci! A taky mám spoustu lidí, kteří mi pomáhají opravovat mé chyby.

CLB: Poslední otázka, pro vás ta nejprotivnější. . . Jaký je váš současný plán na dokončení všech sedmi svazků *The Art of Computer Programming*?

Knuth: Chystám se vydávat dvakrát ročně fascikly o zhruba 128 stranách. Nejdříve shromáždíme čtyři z nich a pak teprve vypustíme první dva; budeme si nechávat něco v záloze! První fascikly čkejte v roce 1995 nebo 1996; budou to testovací beta verze opravdových knih. Počítám, že bych mohl dokončit svazek IV (části A, B a C) v roce 2003, svazek V v roce 2008, pak přijít s novými vydáními svazků I, II a III, potom dělat na VI a VII. Bude i jedna verze jakožto „čtenářský výtah“ ze svazků I až V.

CLB: Jaká by byla vaše kariéra a život, kdybyste nebyl ohlášil ten sedmisvazkový soubor?

Knuth: Počkat, nejdříve jsem to neohlašoval. Myslel jsem si, že píšu jedinou knihu. Ale i kdybych to nebyl udělal, počítám, že bych stejně hodně psal. Nějak se zdá, že po celou tu dobu jsem vždy nacházel zálibení v pokusech vysvětlovat věci. Na střední škole jsem vydával studentský plátek; na vyšší škole jsem redigoval časopis. Vždycky jsem si hrál se slovy.

(překlad: *Ladislav Lhotka*)

Následující příspěvek byl původně napsán v angličtině pro EuroTEX 94. Vždycky jsem si myslel, že je pro autora snadné napsat pak totéž i česky, takže do posledního okamžiku jsem doufal, že mě to nebude stát víc energie, než kolik je potřeba na přetukání. To, že článek vychází česky teprve teď, kdy hlavní důvody pro jeho napsání částečně pominuly, je bohužel jen moje ostuda.

Jak si poradit s velkými METAFONTovými obrázky

KAREL HORÁK

Před nějakým časem referoval o podobném problému na stránkách tohoto občasníku kolega Zdeněk Wagner [2]. Řešení, které je v jeho článku navrženo, předpokládalo rozklad obrázku na několik menších obdélníků

a poté jejich následné složení dohromady při sazbě pomocí $\text{T}_{\text{E}}\text{X}$ ových boxů. V podstatě šlo o stejný princip, jaký používá program `bm2font` F. Sowy: to je rozumné řešení, protože samotný $\text{T}_{\text{E}}\text{X}$ je natolik přesný, že se nemusíme obávat nějakých nepřesností, jež by čtenáři prozradily, že obrázek sestává z menších částí.

S podobnými problémy při tisku větších `METAFONT`ových obrázků jsem se setkal i já, i když ne v takové míře, jak to kolega Wagner tehdy popisoval. Jak je asi všeobecně známo, dokázaly $\text{emT}_{\text{E}}\text{X}$ ovské programy tisknout i hodně velké obrázky (skoro bych tvrdil, že větší, než často bylo možno samotným `METAFONT`em vzhledem k problémům s pamětí vůbec vyrobit). Jakkmile ovšem bylo třeba tisknout otočeně či zrcadlově, byl konec pohody, protože v takových případech šlo tisknout jen charaktery, jejichž počet pixelů nebyl větší než cca 230 000, což při kvalitě běžné laserové tiskárny nestačilo ani na obrázky velké 5 × 5 cm (`bm2font`, aby vyhověl většině běžných ovladačů tiskáren, se omezuje jen na palcové charaktery). Tuto hranici jsem u svých obrázků překračoval dost často.

Tyto problémy se přirozeně znásobily přechodem z 300 na 600 dpi. Do té doby jsem uvedený problém (např. při tisku otočeně pro sestavení brožurky anebo při zrcadlovém tisku na pauzák sloužící jako ofsetová předloha) řešil víceméně technicky: pomocí $\text{emT}_{\text{E}}\text{X}$ ového ovladače `dvidot` s parametrickým souborem `pcx.dot` (původně $\text{emT}_{\text{E}}\text{X}$ obsahoval k tomu účelu samostatný program `dvimsp`, až později jej E. Mattes učinil součástí ovladače jehličkových tiskáren) jsem kritický obrázek převedl na `pcx` soubor a ten zas hned pomocí programu `bm2font` do bitové mapy ve tvaru `pk` ($\text{emT}_{\text{E}}\text{X}$ sice umí tisknout pomocí příkazu `\special` i grafické soubory typu `pcx`, ale to bychom byly tam kde předtím, protože na ně nefungují obvyklé transformace). Na tom mi trochu vadilo, že při novém začlenění takto upravených obrázků do textu se zdrojový text zaneřádl dočasně existujícími soubory (navíc svým charakterem silně závislými na zvoleném výstupním zařízení), které ovšem nestálo zato archivovat, protože se daly kdykoli znovu vyrobit popsáním způsobem. Později jsem proto začal občas dávat přednost tomu tisknout takto celé stránky, které obsahovaly problematické obrázky (což šlo díky schopnější verzi programu `bm2font`, které při rozlišení 300 dpi nevadila ani strana formátu B5). Celý postup jsem snadno zautomatizoval jak ve své původní dávce, tak i v souboru `textbat.bat`, když jsem začal používat program mnu Petra Olšáka.

Po přechodu k LaserJetu 4 se 600 dpi se ovšem začal tento postup zdát málo efektivní, zvláště když jsem před konečným tiskem dospěl k nut-

nosti drobných oprav a musel tak uvedený postup několikrát opakovat. Na tom asi není nic překvapivého, když si uvědomíme, že najednou musel `bm2font` zpravovávat obrázky čtyřikrát větší. To byl patrně hlavní důvod, proč jsem začal raději uvažovat o jiném řešení.

Dospěl jsem tedy podobně jako kdysi kolega Wagner k přesvědčení, že bude potřeba obrázky nějak rozsekat na menší „podobrázky“. První užitečný nápad pramenil ze známé skutečnosti, že \TeX zajímají pouze metrické informace skryté v souboru TFM a je mu úplně jedno, jak vypadají použité bitové mapy, které jediné při tisku činí potíže. Bude tedy nejjednodušší vyrobit všechny části budoucího obrázku jako charaktery se shodným referenčním bodem a s vlastní šířkou nulovou, protože \TeX je pak postupně vysází všechny do téhož místa, jednotlivé části obrázku se budou překrývat, a nemusíme se tak starat o to, aby zvolený rozklad byl rozkladem i v matematickém smyslu, tedy na navzájem disjunktní části. Naopak, některé části se v takovém případě mohou klidně i opakovat. Samozřejmě že nejpohodlnější je zůstat u obdélníků, ale nemusíme se už tolik starat o jejich (nenulovou) společnou hranici.

Zůstaneme tedy u obdélníků, jak ale mají být veliké? To samozřejmě závisí především na schopnostech zvoleného ovladače. Ve starší verzi Mattesových ovladačů ($1.4x$, $x \leq s$) byla problematická hranice někde kolem 480 pixelů (tedy pokud jsme chtěli tisknout s použitím některé ze sedmi netriviálních transformací — při normálním tisku jsem nikdy na problémy nenarazil, snad jedině že natažení fontu do tiskárny trvalo trochu déle než u malých písmenek). Pokud tedy zvolené rozměry našeho obrázku (šířka `{width}`, výška `{height}` a hloubka `{depth}`) odpovídají skutečným rozměrům toho, co nakonec METAFONT vyrastruje, pak určitě můžeme volbu odpovídajících obdélníků vcelku snadno automatizovat: následující výpočet v rámci METAFONTového programu dá postačující počet $n_x \times n_y$ obdélníků, které daný obrázek pokryjí,

```
n_x[charcode]:=ceiling(w/480);
n_y[charcode]:=ceiling((h+d)/480);
```

Hotový obrázek pak můžeme uschovat do proměnné typu `picture` (v našem případě to bude `pic_[charcode]`), ten přidáme k příslušnému vyplněnému obdélníku a to, co bude mít obdélník s naším obrázkem společného, uchováme příkazem `cull currentpicture keeping (2,2)`:

```
picture pic_[];
```

```

beginchar(1,45mm#,45mm#,mm#);
z1=origin;z2=(w,y1);z3=(.7w,h);
pickup pencircle scaled .2mm;
  draw z1--z2--z3--cycle;
cullit;
pic_[charcode]:=currentpicture; clearit;
n_x[charcode]:=ceiling(w/480);
n_y[charcode]:=ceiling((h+d)/480);
endchar;

```

Zde jsme pro jistotu použili příkaz `cullit`, abychom měli jistotu, že uschovávaný obrázek (`currentpicture`) nebude mít „násobné“ pixely. Následujícím příkazem `clearit` vyčistíme obsah aktuální proměnné `currentpicture`, což znamená, že samotný charakter `\char1` nebude obsahovat jediný černý bod, jediný pixel. Nakonec v následujícím cyklu vygenerujeme všechny potřebné charaktery s částmi našeho obrázku, přičemž všechny budou mít (pro $\text{T}_{\text{E}}\text{X}$) nulovou šířku:

```

for $=0 upto n_x1*n_y1-1:
beginchar(incr charcode,0,0,0);
fill unitsquare scaled 480
  shifted(480*($ mod n_x1-.5,floor($/n_x1)));
addto currentpicture also pic_1;
cull currentpicture keeping (2,2);
endchar;
endfor

```

Až dosud jsme se nemuseli příliš starat o počet vzniklých znaků, ale v okamžiku, kdy je máme v dokumentu vysadit, už by se tato informace asi hodila. Ovšem při znalosti počtu $n_x \times n_y = n$ výsledných pravoúhelníků lze použít $\text{T}_{\text{E}}\text{X}$ ovskou smyčku (`\loop`). Předpokládáme-li opět, že $n = 9$, jak tomu bylo v našem ilustračním příkladu pro rozlišení 600 dpi, stačí uvést

```

\newcount\N \N=1
\loop
  \ifnum\N<9\char\the\N\advance\N 1
\repeat

```

Naštěstí ale můžeme T_EX k témuž výsledku přivést i jinak, bez znalosti skutečného počtu vzniklých obdélníků. Využijeme-li nové ligační schéma T_EXu 3.1... a odpovídajícího METAFONTu 2.7..., kde D.E. Knuth poprvé zavedl možnost „okrajových“ (boundary) ligatur [1], můžeme v METAFONTovém zdrojovém textu napsat

```
boundarychar:=255;
```

a pak přidat následující schéma (pořád ještě předpokládáme, že předchozí METAFONTový obrázek byl rozdělen na $3 \times 3 = 9$ částí):

```
ligtable 1: 255 =:| 2;
ligtable 2: 255 |=:| 3;
ligtable 3: 255 |=:| 4;
ligtable 4: 255 |=:| 5;
ligtable 5: 255 |=:| 6;
ligtable 6: 255 |=:| 7;
ligtable 7: 255 |=:| 8;
ligtable 8: 255 |=:| 9;
ligtable 9: 255 |=:| 10;
ligtable 10: 255 |=: 1;
```

což se dá zjevně ještě zestručnit:

```
ligtable 1: 255 =:| 2;
for $=0 upto n_x1*n_y1-2:
  ligtable $+2: 255 |=:| $+3;
endfor
ligtable n_x1*n_y1-1+2: 255 |=: 1;
```

takže pak v T_EXovém souboru stačí napsat

```
\font\fig figures
{\fig\char1 }
```

kde mezera tvořící hranici tohoto triviálního slova zároveň způsobí vysazení `\char2\char3... \char9\char10\char1`. Ve skutečnosti není ovšem mezera potřebná, jak jsem si zprvu myslel, T_EX „pozná“ konec „slova“ i bez ní. To jsem si však uvědomil až mnohem později, protože psát na

tomto místě mezeru jsem byl zvyklý už dávno (pro obrázkové charaktery mezeru (`\space`) obvykle nedefinujeme, proto je její výskyt v uvedené závorce bezvýznamný. Na druhou stranu podléhám nyní pocitu, že právě zmíněný zvyk psaní mezery byl počátečním impulsem ke vzniku popísaného řešení.

Tak, teď už je snad vše jasné, takže se můžeme podívat na definitivní implementaci odpovídajících METAFONTových maker. Podíváme-li se na důležitá makra `beginchar` a `endchar` v METAFONTbooku, snadno též napíšeme obecnou proceduru, která vše včlení do makra `endchar` prostřednictvím `extra_endchar`.

```
%%%%%%%%% divide.mf %%%%%%%%%%
boundarychar:=255;
picture pic_[];

max_charcode:=0;
def char_i(expr c)=
  i:=c; CHars[i]=1;
  max_charcode:=max(c,max_charcode);
enddef;

extra_beginchar:=
  extra_beginchar&"char_i(charcode)";

def divide_pic=
  cullit;
  pic_[charcode]:=currentpicture; clearit;
  n_x[charcode]:=
    ceiling(charwd*hppp/480)+1;
% show n_x[charcode];
  n_y[charcode]:=
    ceiling((chardp+charht)*hppp/480)+1;
  n_d[charcode]:=
    ceiling(chardp*hppp);
% show n_y[charcode];
enddef;

extra_endchar:=
  extra_endchar&"divide_pic";
```

```

def do_pictures=
  def divide_pic= enddef;
  charcode:=max_charcode;
  for ii=0 upto charcode: save Charcode;
  Charcode:=ii;
  if known CHars[ii]:
  ligtable ii: 255 =:| charcode+1;
  for $=0 upto n_x[ii]*n_y[ii]-1:
  beginchar(incr charcode,0,0,0);
  fill unitsquare scaled 480
  shifted(n_d[ii]*down+480*
    ($ mod n_x[ii]-.5,floor($/n_x[ii])););
  addto currentpicture also pic_[ii];
  cull currentpicture keeping (2,2);
  ligtable charcode: 255
  if $<n_x[ii]*n_y[ii]-1: |=:| charcode+1
  else: |=: Charcode fi;

  endchar;
endfor    fi
endfor
enddef;

```

Máme-li teď nějaké větší obrázky, se kterými jsou při tisku potíže, stačí načíst soubor `divide.mf` a použít před koncem souboru `end` příkaz `do_pictures`; . Ale i toho se můžeme zbavit, když v souboru `divide.mf` vhodně změníme definici konce programu:

```

inner end, bye; let saveEnd=end; let saveBye=bye;
def end= do_pictures;
  saveEnd;
enddef;
let bye=end;
outer end, bye;

```

Samořejmě při tom všem musíme ještě trochu dát pozor na jiná chytrá makra, která s oblibou používáme — v mém případě např. skvělý `labtex` Alana Hoeniga, kterým od `EUROTEXu'92` soustavně umísťuji popisky

do obrázků. Ten totiž zas používá kerningové informace na přenos souřadnic popisovaných bodů prostřednictvím TFM z METAFONTového do T_EXového souboru. Takže stačilo přesunout tento úkol z prvních tří znaků 0, 1, 2 na poslední 253, 254, 255 (začínat kvůli tomu číslování vlastních obrázků vždy až od 3 mi nepřipadalo příliš přirozené, zvláště když jsem celé popisované řešení považoval jen za dočasné).

Jiným takovým případem mohou být chytrá makra Olina Ulrycha `incpic`, která používají jiný bezrozměrný souřadný systém a proměnnou `current_transform`, aby zprostředkovala vztah mezi skutečnými a uživatelskými jednotkami. V tomto případě postačí doplnit příkaz `notransforms` před použitím `do_pictures`.

Možné problémy a závěrečné poznámky

Především je třeba si uvědomit, že ukládání obrázků do proměnné `picture` má své meze, neboť rychle zaplňuje paměť, kterou má METAFONT k dispozici, a i při používání Mattesova `mf386` na konečnou kompilaci málokdy svedeme takto víc než 20 obrázků najednou. A s méně mocnými verzemi METAFONTu asi přiměřeně méně.

Stejně jako u programu `bm2font` závisí rozsekávání obrázku víc na počtu pixelů než na jeho skutečných rozměrech, takže konečná rozlišovací schopnost, pro kterou obrázky připravujeme, hraje důležitou roli. Tak, jak jsou makra napsána, dostaneme pro různou rezoluci různé metrické údaje (a samozřejmě také úplně jiné rozložení charakterů ve fontu). Samozřejmě se tomu můžeme vyhnout tím, že dělení obrázků zvolíme v závislosti na nejvyšším rozlišení, které používáme, a nebudeme je pro nižší rozlišení měnit. Také můžeme používat pro různá rozlišení různé adresáře a měnit přístup T_EXu do nich změnou příslušné systémové proměnné.

Uvedený postup se samozřejmě může občas hodit i v jiných případech. DOSovská verze Rokického `dvips`, která teď stejně jako emT_EXovské programy používá Mattesův ovladač `emx`, si sice už tak často nestěžuje, že není schopna zpracovat větší charakter kvůli limitu paměti 64 kB, nicméně myslím, že si stále ještě neporadí s tak velkými charaktery, které je METAFONT schopen vytvořit... Ovšem spíš se může tento postup hodit při rastrování větších ploch METAFONTem, kdy si METAFONT začne velmi brzy stěžovat na nedostatek paměti a začne být neúměrně pomalý.

Téměř přesně po roce, kdy se objevila první z nových verzí ovladačů Eberharda Mattese řady 1.5, které dovedou download na šestisetbodové tiskárny HP, se ve verzi 1.5h zřejmě podařilo odstranit nedostatky při tisku velkých charakterů při použití transformací `/tr1, 3, 4, 6`, které

jsem až dosud musel obcházet právě popsaným způsobem. U této zatím poslední verze přestala konečně zlobit i hláška `internal error 3073: XMS error: 164`. Po intenzivním ročním používání tak snad budu moci odložit `divide.mf` ad acta. V každém případě mi však tato netradiční možnost využití ligačních schopností \TeX u a METAFONTu přinesla radost a potěšení z toho, jak je ten \TeX s METAFONTEM pěkně vymyšlen!

A nejlepší (zejména pro příznivce METAFONTu) zpráva nakonec. Přesto, že popsaný postup překonává některé potíže při používání METAFONT k rýsování obrázků, některá omezení kompilátoru tohoto skvělého programovacího jazyka jsou zřejmě nepřekonatelná. Tedy pokud zůstane METAFONT jenom tím programem, jak ho (pro návrh písma ovšem) navrhl jeho autor, Donald Knuth. Tedy přesněji řečeno, pokud bude METAFONT schopen pouze generovat bitové mapy až do určitého počtu pixelů. Naštěstí jeden z Knuthových žáků, John Hobby, dokázal vnutit jazyk METAFONTu také PostScriptu a vytvořil program `META O T`, který rozumí příkazům METAFONTu, ale místo bitové mapy produkuje PostScriptový výstup. A od ledna je tento program také *in public domain*, tedy volně dostupný a šířitelný. Zatím jen ve své UNIXové verzi, ale John Hobby zároveň slibuje v dohledné době připravit i verzi `META-`

`O T`u pracující pod DOSem. Takže můžeme dál využívat skvělého a geometricky názorného Knuthova jazyka a nestarat se už tolik o závislost na rozlišení příslušného výstupního zařízení. Až se budu moci s `META-`

`O T`em seznámit na vlastní kůži, rád poskytnu lepší odpověď na otázku v titulu tohoto článku, než se mi dosud mohlo podařit.

Literatura:

- [1] Donald E. Knuth: *The new versions of \TeX and METAFONT*, *TUGboat* 10 (1989), 325–328.
- [2] Zdeněk Wagner: *METAFONT a velké znaky*, *\TeX bulletin* (1993), 77–83.

O jedné vraždě a jejím možném odčinění

KAREL DOSTÁL

Motto:

Prondu — prondu.

Neprondu — neprondu.

ESKYMO WELZL

Tak nám zabili... pomlčku, řečenou pauzu,

a já se s tím nehodlám smířit. Pomlčka totiž není zbytečný luxus, má své místo na slunci, své opodstatnění, není to nic redundantního, žádný starý křáp na vyhození do popelnice.

Pomlčka — praví se v Pravidlech českého pravopisu — „*naznačuje větší přestávku v řeči nebo od sebe výrazně odděluje části projevu...*“ A dále: „*Možnosti užít pomlčky v platnosti citové jsou velmi bohaté a mnohotvárné a nelze je schematicky vypočítat.*“ A do třetice: „*Od pomlčky je třeba odlišovat (podtrženo námi) spojovací čárku (spojovník).*“

O spojovníku (rovněž zvaném tiset, divis), což je grafický znak v podobě vodorovné čárky podstatně kratší než pomlčka, praví se tamtéž, že je kladen „*bez mezer mezi slova nebo jejich části. Užívá se, chceme-li vyjádřit, že jím spojené výrazy tvoří těsný... celek. Spojovník se graficky i funkčně odlišuje od pomlčky.*“ Aspoň tolik úředně.

Ale teď už k vraždě samé. První krok k ní byl učiněn zhruba před sto lety s vynálezem a rozšířením psacích strojů. Ve snaze o ekonomičnost jejich konstrukce byl sveden nelítostný boj o každou klávesu, bez které se bylo možno obejít. Mezi prvními oběťmi této řeže se ocitl například středník, v běžném textu ostatně málo používaný a tudíž zdánlivě ožehavý. (Bylo jej navíc možno zkonstruovat vyklepnutím dvojtečky, vrácením vozíku o jednu mezeru a doklepnutím čárky.) Na řadu pak přišly obě kulaté závorčky (o hranatých ani nemluvě, neboť mají význam jen pro specializovaný text), které splynuly vjedno se šikmou zlomkovou čarou. U komfortněji vybavených moderních strojů to dávno neplatí, ale v úspornějším režimu přenosných strojů bývá tomu tak dosud. Boj pokračoval: Jedničku nahradilo písmeno (1), štíhlou nulu baculatější, ale přesto podobné (0). Uvozovky, jak anglické (“ — ”), tak francouzské

(« — »), či německé („ — “), (« — «) sestávaly podobně jako kulaté závorčky ze dvou znaků a musely tedy vyklidit pole znaku jedinému — (“), který zastal funkci předních i zadních uvozovek. (Tady je ostatně nejméně čeho želet, protože při mírumilovné koexistenci různých stávajících druhů uvozovek mezi ně mohla zapadnout ještě jedna další modifikace docela dobře.) Až potud můžeme — byť s kapkou trpkosti v duši — svoje námitky víceméně potlačit. Horší a důsažnější už je, že se někdejším tvůrcům klávesnice jevil zcela zanedbatelným nějaký milimetrový rozdíl v délce pomlčky a spojovníku. Ti by nám navíc mohli protiargumentovat tím, že klaviatura kromě spojovníku (-) je vybavena i podtrhovací čarou (⏟), která je rozeznatelně delší (aby při podtrhávání vytvořila souvislou linku) a pouhým posunutím válce o jeden zub nazpět se octne v žádoucí poloze pomlčky (—). Kdo by se s tím ale, řekněte sami, páral? Zvláště když se z obecného povědomí pomlčka postupně takřka vytratila?

Z řečeného tedy vyplývá, že můžeme mluvit o vraždě z úsporných důvodů. Tehdy se koneckonců zdálo, že není tak dalece čeho litovat. Navíc se očím, až dotud uvyklým na nepravidelnost, různorodost a někdy i velmi problematickou kvalitu manuskriptu musel na stroji psaný text nutně jevit jako vrchol dokonalosti, na níž marno hledat chybičku. A k tomu všemu: jakmile se strojopis určený širší veřejnosti dostal do povolaných rukou sazeče písma, typografa, ukázalo jediné znaménko okamžitě svou dvojdomost a proměnilo se zpětně tu v pomlčku, tu ve spojovník, jak třeba. Žádný sazeč v tom na vteřinku nezaváhal. To víte, fachmani! Frajírkové!

Takže vražda se vyklubala na světlo boží až po nějakých bratru počítáno sto letech, s příchodem computerové éry, kdy klasická horká sazba s příchutí magie, v níž se vyznali jen zasvěcení, kteří pohrdajíce metricou soustavou zatvrzele měřili na body a cicera, se takřka přes noc ocitla ve starém haraburdí.

Ke keyboardům počítačů usedli nesazeči, netypografové, a pustili se mezi jiným bez skrupulí, zato s tím větší suverenitou, tímto erbovním štítem neznalých, i do zhotovování novinové, ba ještě náročnější knižní „sazby“. A ačkoli vybavení každého slušnějšího computeru hravě umožňuje realizovat všechny divy a zázraky klasické horké sazby, ba dokáže i daleko víc než sazba ruční, „sazba“ počítačová rázem poklesla zpátky na úroveň strojopisného textu. Mnozí stačili mezi strojopisným a počítačovým textem postřehnout jeden jediný rozdíl — že totiž počítač dokázal bez problémů produkovat stejně dlouhé řádky. To v nich samo o sobě navozovalo pocit tak dobře vykonané práce, že se o víc nestarali.

A v pohodě, nikým nekorigováni, nikým nekritizováni, nikým nepoučení vnašeli do svého díla nejen různé návyky (a zlovyky) z psaní na psacím stroji, ale i některé pochopitelné nezalosti. Podívejme se na ně blíže.

Například stejný prostor pro každou literu či znak u psacího stroje způsobil, že za objemově nepatrnou tečkou či čárkou vznikla nadměrná mezera. A tak ji většina pisarů zrušila vůbec a psali 1.srpna, A.S.Puškin atp. Působilo to snad i estetičtěji než při použití mezeríku a volný prostor kolem tečky sám navozoval iluzi mezery. Z počítače, který chytře poskytuje každému grafickému znaku jen tolik „vzduchu“, kolik mu přináleží, vám ovšem při takovém postupu vyleze nevábná (a nelogická) slepenina, která nerespektuje nutnost mezer mezi jednotlivými slovy, přičemž „sloven“ se tu samozřejmě rozumí i řadová číslovka či zkratka křestního jména. (Nadbytečná mezera se naopak občas objevuje před znaky (!), (?) či (:), a rovněž u uvozovek či závorek, ačkoliv se všechny tyto znaky logicky přimykají k předchozímu nebo následujícímu slovu a nemají povahu samostatného znaku.)

Nežádka se pod tlakem dřívější zkušenosti dává bezdůvodně přednost zlomkovým čarám ve funkci závorek před kulatými. Lomítka se objevují ve specifické funkci rovněž u textů, v nichž recenzent básnické sbírky cituje verše a radí je pro úsporu místa in continuo. Jednotlivé verše jsou pak od sebe odděleny lomítkem, před nímž a za nímž musí být mezera, aby nedocházelo k mylné interpretaci veršů jako textu v závorkách. Před novou strofou pak přijde lomítka zdvojené. Tedy: *„...a strašní jezdcí Zjevení / mávají praporem // Je lehké listí vavřínů / a těžký padlých stín // Já vím Já vím...“*

Jinou takovou vadou na krásu, plynoucí rovněž z předchozí — rozdílné — zkušenosti s psacím strojem, je nerozlišování mezi literou (l) a číslovkou (1), či mezi verzálkou (O) a útlejší nulou. Počítač ovšem něco takového neodpouští a je vždy dobré si připomenout, kterou z příslušných čtyř kláves máme stisknout. (Mimochodem, tomuhle nerozlišování bych se zas moc nedivil. Jednak zvyk je železná košile a jednak tu nerozlišuje ani můj elektronický psací stroj s písmem PRESTIGE1012: ačkoli mi jeho klaviatura nabízí čtyři různé možnosti a místa je tu tedy dost, autoři české verze typového kolečka obsadili tato čtyři pole pouhými dvěma vždy identickými znaky — (l) a (O). Inu, i mistr tesař...)

Další nezalost řemesla se projeví tam, kde např. číslovka 10 000 se dělí do dvou řádků a osiřelé tři nuly se před žasnoucím čtenářem objevují na začátku následujícího řádku. To se zkratka někdejšímu sazeči nemohlo

stát. Věděl, že tři nuly, samy o sobě postrádající smyslu, musí nahradit slovním výrazem *tis.*, případně *tisíc*, anebo dát celou číslovku až do následujícího řádku. Věděl také, že nemůže — alespoň v knižní sazbě — dávat neslabičnou předložku *v*, *k*, *s* atd. na konec řádku. A že tam naopak může klidně ponechat spojku *a* či *i*. Neboť spojka, na rozdíl od předložky, nepatří jen k následujícímu slovu, ale i k předcházejícímu; spojuje je, jak už její název napovídá. (Setkáváme se naopak v některých novinách i s extrémní hyperkorektností, kdy se na konci řádku neobjeví ani zmíněné větné spojky. Pachatelé takových hříšků byli zřejmě poučeni, bohužel jen napůl. Jde tu o nerozlišování mezi předložkami a spojkami a dost se divím, že se v tomto směru nepoučí pouhým minutovým nahlédnutím do Pravidel alespoň páni korektoři, ač to mají takřka v popisu práce. Ke všemu je to počínání kontraproduktivní a navíc neestetické — způsobí někdy i víc než centimetrovou mezeru mezi slovy; k čemu už je taková hyperkorektnost dobrá?) Osobně bych se — přinejmenším u novinové sazby s poměrně krátkými řádky — přimlouval za více tolerance i v umisťování neslabičných předložek; zdá se mi, že jsme tu někdy (ba i platná oborová norma) papežšější než papež.

K užitku ale přijde povědomost o opakování spojovníku v případech jako u spojky *-li*, např. výraz *je-li*. Pokud první část výrazu vyjde na konec řádku (*je-*), musí mít druhá část na začátku řádku následujícího podobu (*-li*). Jinak bychom nechtěně vytvořili slovesný tvar *jeli*. Obdobně je tomu v případech jako *Joliot-Curie* ap. Zhusta se naopak setkáváme s nadužíváním spojovníku tam, kde *-li* je pouhou součástí složené spojky jedině. Místo náležitého *nežli*, *zdali*, *jestli* atd. se pak setkáváme s příšerkami *než-li*, *zda-li*, *nebo-li* atd. ad libitum. To už vůbec nemluvíme o ošklivce *je-li* (s pomlčkou). Nejde o vymyšlený příklad: mám před sebou corpus delicti v podobě dvou knížek nakladatelství Odeon (1992), považovaného vždy za příkladné. Před počítačovou sazbou vzalo jeho renomé zasvé. Autor Bohuslav Brouk: *Autosexualismus a psychoerotismus*, a tentýž: *Lidská duše a sex*. V prvním případě zhotovily sazbu Tiskařské závody Praha a *je-li* s pomlčkou se tu užívá důsledně. Podobně i ve složeninách typu *médico-légale* nebo *Kraft-Ebing*. (Příčemž *Kraft-Ebing* s pomlčkou navozuje představu, že jde o dva pány.) V případě druhém (sazba firma AMOS Praha) se u *je-li* pomlčka se spojovníkem libovolně střídá, o složeninách platí totéž co v prvním případě. Počítačový sazeč tedy na rozdíl od písaře na stroji pomlčku k dispozici má, nezná však její specifické poslání a začasť jí proto neaplikuje vůbec (další příklady

netřeba uvádět, stačí zalistovat každou druhou knihou z posledních let), anebo jí používá chaoticky, bez znalosti její funkce.

Jiný takový nešvar plynoucí ze strojopisné zkušenosti je podtrhávání slov či celých vět k jejich vyznačení. Počítač nám podobně jako klasický sázecí stroj dává k vyznačování hned tři možnosti: písmo **tučné** (polotučné), *ležaté* (kurzivu) a **p r o s t r k a n é** (laicky řečeno proložené). Můžeme dále vyznačovat VERZÁLKAMI či KAPITÁLKAMI; k čemu ještě podtrhávání, které nejen nepěkně vypadá, nejen je netypografické, ale navíc při větší hustotě řádků ztěžuje čitelnost?

S tlumeným rachotem klasických linotypů a monotypů vymizel z našeho světa dobrý mrav umět si poradit s umístěním uvozovek. A je to přitom až na některé zřídkavé případy docela prosté: Uvozovkám **za** větou tečka (nebo čárka) předchází. *Rekl.*: „*Mám se dobře.*“ — „*Mám se dobře,*“ *řekl.* Je-li však v uvozovkách pouze výraz nebo část věty, uvozovky předcházejí interpunkční znaménko: *Byl z toho „jurodivý“.* — *Starí říkávali „všude dobře, doma nejlíp“.* A obdobně je tomu se závorkami. Nedávejte, pěkně prosím, u věty v závorkách tečku až za závorku. Patří přece k té větě, a ne kamsi do volného prostoru. A víte snad, co zla napáchá nesprávně umístěná závorka v matematice! Ještě k těm uvozovkám: existuje zlovyk dávat vždycky (někdy je to oprávněné) do uvozovek výraz nezvyklý, nespisovný nebo ironicky míněný: *Jsem z toho „paf“.* — *Byl to „pěkný“ ptáček.* Pokud někdo neví, co je to redundance, dobře mu to vysvětlíte právě na tomto příkladě.

Za nepřilíš vhodné považuji rovněž, když jsou vkládány do uvozovek názvy uměleckých (a dalších) děl, výrobků ap. Pro příklad stačí, že v odborné práci musíme jmenovat v rychlém sledu za sebou třeba desítku děl a hejna kumulovaných uvozovek pak naruší, „rozhodí“ klidný, vyrovnaný obraz celé stránky. Obvykle raději vystačíme s tím, že název vyznačíme verzálkou (jako každé vlastní jméno): *V Babičce B. Němcové...* Možná dvojí interpretace přichází jen tehdy, octne-li se název na začátku věty: *Babička B. Němcové...* Celý problém ovšem zmizí, použijeme-li pro názvy děl kurzivu. (Povšimněme si tu běžné slovníkové praxe, která si počíná právě takto.) Uvozovky bych použil leda v případě, kdy chceme naznačit podřazenost (nadřazenost) jednoho díla druhému, část celku např.: Povídka „Rebeka“ ze sbírky J. Škvoreckého *Hořkej svět* ap. — Nemluvě o možném ironizujícím účinku uvozovek např. v názvu deníku „Pravda“. Co už může být pochybnějšího než pravda v uvozovkách!

Už jako typografický učedník v Břejchově venkovské oficíně jsem nabyl ponětí o tom, jak si počínat při dělení do dvou řádků u takových

ošemetností, jako je *10 m*, *100 %* či *10–20*. Pomlčka (a ne spojovník!!) v posledním příkladě má význam (až); jen si to přečtete nahlas. Dělíme tedy *10 | až 20*, případně *10 až | 20*. Značky *m* či *%* nabudou při dělení slovní podoby: *10 | metrů*, *100 | proc.*, atd. A samozřejmě jsem musel vědět, že *100 %* (s mezerou) je sto procent, kdežto *100%* (bez mezery) znamená stoprocentní. Nemá proto smysl psát *100%ní*. (Pravidla v příkladech *10 m*, *100 %* atd. doporučuji je do dvou řádek nedělit, ale naše praktické východisko nenabízejí, ačkoli oborová norma z r. 1975 je uvádí. Přitom se v úzkých novinových sloupcích lze někdy dělení jen těžko vyhnout.)

A už tehdy jsem šetřil apostrofy, jakož i vším ostatním, co bylo nadbytečné. Když jsem četl: *On to zved'*, dýchl na mě Jaroslav Vrchlický s naftalínovou vůní 19. století. (A že by totéž bez apostrofu bylo někomu nesrozumitelné, to se snad nedomnívá ani ten, kdo apostrof vymyslel.) Nu a v případech jako *žes to viděl, dobřes udělal* nemá apostrof už vůbec co pohledávat; to byl železný zákon, který — ač často nerespektován — platí dodnes. (Viz Pravidla.)

Taky od těch dob vím, že číslovky 1–4 vypisujeme v běžném, zejména beletristickém textu slovy. Ale i vyšším číslovkám to v takovém prostředí lépe sluší: mít dovolenou *čtrnáct* (a ne *14*) dní, či si koupit noviny za *pět korun* (místo za *5 Kč*, natož *Kč 5,-*) je jaksi lidštější, méně zdeůřední. A že byste doběhl do cíle *2.*, a ne *druhý*? Tak vidíte.

S řadovými číslovkami jsou i jiné potíže: *5 tého* nebo *5-tého září* napíše autor bez uzardění, ale sazeč profesionál tam dá suverénně tečku — *5. září* —, případně volí slovní podobu *pátého*. Rovněž odmítne zpotvoření typu *5 ti*, *5-ti*, *5-ti denní*. Hodlá-li už autor číslovku skloňovat, ať nelení a píše *pěti*, *pětidenní* (pěkně dohromady, jde o jedno slovo!). Může sice napsat *5násobný*, ale beze všech *ti* a *-ti* a opět bez mezery.

Později na filozofické fakultě mě — zaplaťpánbu — učili kromě jiného psát *Chci ti*, *Karle*, *něco povědět* — nikdy vokativ bez čárek! A taky že si nemám plést *jako by* a *jakoby*, protože to první má ve větě význam *jako kdyby* a je provázeno slovesem: *Jako by (kdyby) to neslyšel*. A to druhé má význam spíše přirovnávací, obejde se bez slovesa a tulí se nejraději k přídatnému jménu: *Byl tichý, jakoby zaražený*.

Naučili mě tam i pár dalším věcem, takže jsem dodneška konsternován, když se například někdo vyznává z lásky k hvězdám, zářících na nebi. Tu postojím v tichém zadumání neschopen slova a pohybu a přemítám, čemu to vlastně páni učitelé naši mládež v hodinách českého jazyka učí, že jim nezbývá čas na skloňování? Měl bych asi být otrlejší, vždyť takřka

denně se v rádiu či televizi mluví ne o *pěti stech*, nedíví se tam *pěti stům*, neděsí se *pěti set*, ale jen a jen *pětseti* mrtvých. . .

Tohle vše už ale přesahuje naše nejvlastnější téma a sazeč jaksí nemá v popisu práce dávat autorům lekce z českého jazyka a pravopisu. Mohl by však vědět a mít právo oproti nedbalému autorovi uplatňovat holý fakt, že neexistuje spojka *a nebo*, že není žádné *nebo-li*, nejezdíme *Felicií* a vždy bereme něco *s sebou*, nikoli *sebou*, byť se to takovými neblahostmi ve zpracovávaném textu jen hemžilo. Sazeč pochopitelně není kompetentní zasahovat do textu v ničem, co by pozměňovalo (být k lepšímu) jeho smysl. A nemůže například u dublet vnucovat autorovi psaní *-ismus*, jestliže ten trvá na podobě *-ismus*. Kde se však jedná o zcela jednoznačný prohrěšek proti kodifikovanému pravopisu, tam se jen pošelilý autor s ješitností zaslepenýma očima pohorší, místo aby za podobné zásahy sazeči poděkoval.

Tady bych se rád zmínil o jednom — spíše grafickém — solitéru, se kterým se pilný čtenář setká takřka denně. Je to *ZOO* umně vyvedené bůhvíproč z verzálek. No, já tuším proč asi: tento zkratkový výraz pro zoologickou zahradu jsme si vytvořili asi tak jako ne zrovna spisovnou *stovku* za stokorunu nebo závod v běhu na 100 m. Jeho zkratkovitost a nezvyklé hláskové složení však v nepřilíš přemýšlivých autorech navozují pocit, jako by šlo o zkratku samu. Navíc zkratku zvanou iniciálovou, vzniklou z počátečních písmen víceslovných názvů — USA, OSN, ZOH ap. Pak by ovšem *ZOO* muselo znamenat třeba *Základní ostravskou organizaci* nebo tak nějak. Našemu případu však sluší skromné minusky a zvířátka v pražské *zoo* to ponесou statečně, věřte.

Ale hrnečku, už nevař, kaše nějak přebývá! Vždyť takhle a podobně bych mohl pokračovat ještě pěknou chvilku — a já nechovám ctižádost sepsat manuál počítačové sazby, nýbrž pouze ve vši skromnosti osvětlit pozadí a některé okolnosti jedné vraždy. Už proto bych měl pomalu škobrtat ke konci. Přece jenom však cítím, že bych si neměl odpustit ještě zmínku o věci, která má v současné době tvář dosti hrůzoplňnou: rozdělování slov. To by však byla jednak celá samostatná — a ne zrovna krátká — kapitola, a jednak jde o záležitost víceméně formální, kde by slušná porce tolerantnosti byla namísto spíš než nadměrná úzkoprsost. Nedá mi nicméně, abych se nezmínil o několika nejčastějších rozdělovacích horůrcích, které — tolerance sem, tolerance tam — volají o pomstu. Způsobilo je zejména mechanicky uplatňované školácké pravidlo, jež ve skutečnosti mělo být jen hrubě orientačním pomocným vodítkem (a které přesto bylo, zdá se, zabudováno do počítačových programů), že

totiž jdou-li ve slově za sebou dvě souhlásky, dělíme slovo právě tam. Takže je tu zcela potlačen daleko podstatnější zřetel ke skladbě slova, jeho předponám, příponám a koncovkám, a vznikají alogické příšery jako *nej sem, naj de, naz dar, pot řeba, krajs ký, bojes chopnost, přet lak, černok něžník, třikrát* atd. do alelujá. Ale i při bezchybném dělení dochází k nežádoucím jevům a do textu se vloudí nechtěná *mys* a *Lenka* (ač Pravidla připouštějí způsob *my šlenka*), mezi *katem* a *skatem* můžeme, ba musíme volit u slovesa *praskat, citu* a *ronu* se však zbavíme povoleným *ci tronem*; zato hned dvě domácí zvířata nám nabídne *tele skop*, a *sex tet* ponecháme bez komentáře. Každý sazeč pak — kromě vyloučených zlomyslníků a lidí bezelstných — se poleká, má-li — jinak zcela regulérně — dělit *se kunda, kni hovna, tlu močení, cá kanec, sere náda, ser vis, ser pentina, ba kule, tu pohlaví, sle pička*.. Bylo na těchto stránkách konstatováno (bohužel s matoucí chybou — Steifelter) *Stiefel tern* a *Spargel der*, kde se nám vnucují *boty* a *chřest* namísto *nevlastních rodičů* a *úspor*; ale i v češtině lze dělením změnit význam slova. Např. můžeme *pod robit* (lid) a *po drobit* (ubrus) ap.

Pravopis samozřejmě není pámbůh, natož jazyk sám, a nelze o někom, kdo zakolísá v použití předložkového či předponového *s* a *z*, zaváhat nad kvantitou *e* ve slově tematika (proti téma) či nad volbou mezi *i* a *y* ve slovese vymítati (zajímavý případ, který tu nemohu rozebírat) hned prohlásit, že neovládá spisovný jazyk. (Tak jako mu nelze vytýkat, použije-li vědomě a zdůvodněně — a nejenom v uměleckém kontextu — nespisovného výrazu či tvaru. V nepochybně odborném textu P. Sojky jsem našel citoslovečné „*Jupííí!*“ a „*ouha*“, které jaksi do odborného stylu nepatří. Chce se ale pro to někdo mračit?)

Mnohé je dáno tím, že jako by zmizely dva důležité mezistupně mezi sazečem a autorem: korektor tiskárenský a korektor redakční. Anebo existují, ale jen nominálně; nemají potřebnou kvalifikaci, nepracují s Pravidly, o náležitostech typografických nemají vůbec potuchy a hlavně je to nezajímá. Opravují a upravují, jak je to zrovna napadne, zcela subjektivisticky, většinou však vůbec ne.

Takřka vše, o čem tu byla řeč, by zmizelo jako mávnutím kouzelného proutku, kdyby neexistoval pozoruhodný fakt, že ač je náš národ po dlouhá desetiletí vybaven knížkou, která se jmenuje Pravidla českého pravopisu a která má ve svém úvodu několik málo (přesněji 65) stran přístupně informujících o těchto a podobných záležitostech, není mu to moc platné. Do těchto pár jakoby zakletých stránek nepocítil zřejmě potřebu nahlédnout aspoň jednou za život skoro nikdo z těch našich milých

spoluobčanů, jimž je psaná, tištěná a profesionálně mluvená čeština denním chlebem. Na to, co je správné a co nesprávné, máme zkrátka raději názor. Nač je nám třeba poučení? Ještě bychom si nakonec třeba museli názor poopravit, a to my neradi. — Nemohl jsem si ostatně nepovšimnout, že nejplamennějšími obhájci správnosti či nesprávnosti v češtině bývají zhusta ti, kterým dělává sestavení kloudné české věty mnohdy značné potíže. Odborníci tu bývají daleko tolerantnější. . .

Vraťme se ale k tématu. Abychom tedy mohli mluvit o skutečně kvalitní počítačové sazbě, musí být splněny dva předpoklady: zřetel čistě typografický, bezpečná znalost historicky vzniklých a ustálených pravidel „správné“ sazby a jejich nekompromisní respektování, a za druhé zřetel ne-li češtinářský (tady jsou věci složitější), tedy rozhodně ortografický, respektování Pravidel českého pravopisu. Nejsou-li obě tyto složky v rovnováze, *vždy* utrpí kvalita celku. Jde tu tedy o odbornost a kvalifikaci interdisciplinární, která v rozvinuté, ba ani zárodečné podobě nenašla dosud místa na slunci. Mám v duchu před očima hypotetický manuál počítačového sazeče, který by stejnou měrou bral v potaz obě složky a propojil je v jeden celek, v němž by se oba zřetele, typografický a ortografický, neustále spájely, překrývaly a posléze splývaly v jedinou vyhraněně specializovanou disciplínu. (Shodou okolností se právě při psaní těchto řádek dovídám, že podobná publikace pod názvem Typografický manuál se už chystá a slibuje být odpovědí na otázky, o nichž jsem před chvílí mluvil. Bylo by to věru pravé slovo v pravý čas!)

Tak, a teď to málem vypadá, že jsem zapomněl, o čem byla původně řeč. Nu ano, já vím, o vraždě pomlčky. Nemáte ale pocit, že tahle zdánlivě osamocená vražda vyvolala celou řetězovou reakci? A že se tu pozvolna vraždí už nejen samotná chudinka pomlčka, ale sám duch, tradice, vyzrállost, dávno už dosažená kvalita a úroveň česky sazeného textu? Začalo to pomlčkou — a konce nedohlédnout.

Díky počítačům, které samy ovšem za nic nemohou, vstoupili do druhdy posvátného hájemství noví lidé, jistě zdatní, jistě schopní, ale taky bez tradičního zázemí, bez náležité odborné (pokud jde o text) průpravy, bez potřebné pokory, která by měla být samozřejmostí před dílem, jež denně ovlivňuje miliony lidí, tříbí (anebo kazí) jejich vkus, předkládá jim královskou krmí — anebo halabala uklohněný blaf.

Jistě, staří sazeči nevyumřeli přes noc. I oni změnivše verpánek usedli k počítačům, možná se zařatými zuby, ale s řemeslem v krvi. A jistě budou mít své následovníky, takže naše typografie — chci pošetile doufat — nemá snad navždy to nejlepší už za sebou. I když dnešní zákazník

spíše přijde (je třeba šetřit!) do tiskárny už s hotovou disketou v podpaží — a sazeč si může své umění a znalosti vstrčit za klobouk...

Nu dobrá, pryč jsou doby, kdy pečlivý akcidenční sazeč prokázal kvality mistra i na obyčejné merkantili, dodacím listu nebo účtence, vyrovnával řádku z verzálek za pomoci achtlíků, šestinek i čtvrtpetitů, ba i nastříhaných papírků tak, aby mezi VAV nebylo víc světla než mezi MHM; bylo to věci stavovské cti a o křehké, lámavé achtlíky, osminky, byla v prestižní sazárně vždycky nouze... Co je tam, to je tam a přicházejí — už dávno přišly — doby jiné. O vyrovnávání se postará třeba program Kern-it, některé fonty mají hotové kerningové páry, ale zajímá to někoho? Počítač nemá cit a duši starého fachmana, teprve virtuózní programátor mu dokáže vdechnout cosi až nebezpečně podobného. Nic není ztraceno, všechno se může vynořit znovu jako Venuše z pěny mořských vln.

Priznám se bez mučení: Ono prvotní okouzlení z nádhery tohoto řemesla, které si tyká s uměním, mě neopustilo dodneska, takže všechno, co mu ubírá na kráse, mě doslova fyzicky bolí. Nemluvím proto o věci z nedostatku lepších témat či proto, aby řeč nestála. Spíš bych tu chtěl najít způsob, jak se postavit té vražedné lavině, jak ji sprovodit ze světa. Věřu nepodléhám iluzi, že je to úkol na týdny či měsíce. Víím, jak strašná je síla setrvačnosti. Jak nesnadné je pohnout s jednou zaběhnutými zvyklostmi, zvláště u velkých, těžkopádných kolosů.

Ale! Ale teď je přece u nás mnoho čiperných a ambiciózních editorů drobných a o to pružnějších, kteří by si — možná — dali spíše říct. Třeba proto, že by jim zkrátka moje snaha po odčinění jedné vraždy byla blízka, nebo jen proto, že by jim zalichotila úloha štiky v rybníce. Oni, ne já, mají v rukou tu potřebnou výkonnou moc. Oni jsou to, kdo by se mohli stát těmi — jak se to říká? — orgány činnými v trestním řízení. K nim proto především upínám své naděje, že by náš boj o zmrtvýchvstání, o znovuvzkříšení pomlčky nemusel zůstat jen a jen donkichotský.

Myslím si opravdu vážně, že každý vydavatel, který si chce na sobě dát jen trochu záležet, se nutně stane spojencem takového tažení. Nebude ho to dohromady nic stát, chce to spíš probudilost a dobrou vůli. A přesto by to byla jeho možná nejdůležitější investice, která by ho takřka naráz vyzvedla do nejvyšších pater nakladatelského podnikání. (Pokud by — pochopitelně — vsadil zároveň na kartu kvality umělecké či odborné.)

Nu — udička je hozena, uvidíme, kdo ůkne první. Já se teď s tímhle textem obrátím na svou milou Pražskou imaginaci — třeba se jí zachce

zahrát prim, vydá mi to jako novoročenku — a tažení bude zahájeno ve vší parádě. (A neříkejte mi, prosím, že tohleto je rána pod pás panu Kadlecovi, protože teď nemůže ucuknout, aniž by ztratil glanc. I může, samozřejmě že může! Jen se podívejte na jeho postavu! Pochybují, že by vám při tom pohledu vytanula na mysli právě štika...)

A potom si to tvrdohlavě vydám nákladem vlastním a budu to rozesílat všem, kteří to mají a mají a mají čist; nechám si to něco stát, jak by řekl Karel Poláček.

Protože pomlčka není žádný starý křáp. Protože — — —

Ale mám takový dojem, že už jsem vám to říkal.

— — —

P. S. Nebo jste nic neslyšeli?

Pozn. autora: Text skutečně vyšel v nakl. Pražská imaginace jako novoročenka 1995, což omlouvá jeho fejetonistický, lehce nahozený ráz, nedotýkající se subtilnějších záležitostí. Pro nynější účel jsem jej poněkud přepracoval a rozšířil o několik pasáží, čímž jsem patrně narušil jeho causeristické ladění, ale zde mi to snad bude vzhledem k potenciálnímu čtenáři prominuto. (I tak jsem se dotkl jen základů sazby a nebral jsem v potaz další fáze: stojíme přece před faktem, že computer je zároveň — a málem bych řekl v jedné osobě — sazečem, metěrem i výtvarníkem.) Že můj text vzbudil zájem právě u editorů tohoto bulletinu, považují za víc než nadějně pro další osudy naší typografie. A že se jej navíc odhodlali přetisknout, pokládám si za čest. Ty slyšící uši jsem totiž — což je pro mne víc než krásné — našel právě tam, kde se o tomto lepším příštím nikoli sní, ale kde se už v tuto chvíli s velkým nasazením a na vysoké profesionální úrovni mění ve skutečnost. (Např. stať J. Veselého v č. 2/93 je pro mne v tomto ohledu ambrózií, nektarem a manou zároveň.) Co si mohu přát víc?

Autor (*1929) vyučen ruční, poté strojní sazeč; po dokončení filosofické fakulty UK (čeština – literární věda, dva semestry slovenštiny v Bratislavě) časopisecký, nakladatelský a rozhlasový redaktor, poté volný publicista a překladatel, po srpnu '68 znovu v tiskárně, nyní editor a externí spolupracovník Rádia Svobodná Evropa.

Vražda na obálce je dílem Josefa Váchala (výřez z dřevorytu z knihy Krvavý román).

Program IMARK: Nástroj usnadňující vytváření rejstříků v T_EXu a L^AT_EXu

JAN KYBIC

1. Úvod

Rejstřík je velmi důležitá a užitečná část každé vědecké či technické knihy. Umožňuje čtenáři velice rychle nalézt žádanou informaci. Bohužel ale jen málo dnes vydávaných knih rejstřík má, nemívají ho ani knihy vytvořené na počítači.

Zaměřím se nyní na dokumenty vysázené v T_EXu (čimž myslím i L^AT_EX a další makra). Uživatelé L^AT_EXu mohou vytvářet rejstřík relativně jednoduše. Stačí označit ve zdrojovém textu všechna slova, která se v něm mají objevit. Po prvním průchodu L^AT_EXem vznikne soubor `idx`, který je nutné programem `makeindex` (nebo `csindex`) převést na soubor typu `ind`. Po druhém průchodu L^AT_EXem se už rejstřík objeví na svém místě. Uživatelé PlainT_EXu použijí v zásadě stejný přístup, pouze budou muset napsat několik jednoduchých maker.

Bez ohledu na další zpracování je však vždy nutné nějakým způsobem označit všechna slova, která přijdou do rejstříku. Někteří autoři doporučují po napsání dokumentu ho celý znovu projít a ručně označit každý významný výskyt klíčových slov a termínů. Je to ovšem zdlouhavá práce a je zde riziko chyb a přehlédnutí. Druhý přístup je proto použit k označování slov nějakým způsobem počítač. A to byl důvod k napsání programu *IMark*.

2. Popis činnosti

IMark umí vytvořit abecedně seřazený seznam slov ze zdrojových textů T_EXu. (Slovo se skládá z jednoho či více písmen). Je rovněž možné vytvořit jeden seznam z mnoha zdrojových souborů, máme-li například každou kapitolu knihy v jednom souboru, ale rejstřík chceme mít společný. Jelikož vzniklý seznam bývá opravdu rozsáhlý, je nutné jej ručně editovat. (Seznam je čistý ASCII textový soubor. Každá řádka obsahuje jedno slovo.) V seznamu ponecháme jen ta slova, která chceme zařadit do

rejstříku. Pokud do řádky v seznamu napíšete dvě slova, pak bude *IMark* ve zdrojovém textu vyhledávat první z nich a druhé slovo (zde je definice slova zobecněna na libovolný řetězec omezený mezerami) se stane argumentem indexovacího příkazu. Tímto způsobem lze ovlivnit formu konkrétní položky v rejstříku (např. změnit font) nebo posílat informace programu `makeindex`.

Máme-li připraven seznam slov patřících do rejstříku, zavoláme *IMark* ještě jednou, ovšem s jiným příkazem. *IMark* projde zdrojové texty, přičemž hledá slova ze seznamu. Pokud nějaké najde, připiše ze něj indexovací příkaz, což je implicitně `\index`. Argumentem tohoto makra je buď nalezené slovo samo, nebo druhé slovo na příslušném řádku v seznamu slov, jak je zmíněno výše.

Někdy bude nutné vzniklý zdrojový text pro \TeX lehce upravit, pokud například *IMark* umístí příkaz `\index` do argumentu nějakého makra, nebo pokud chceme zahrnout do rejstříku něco, co *IMark* nepovažuje za slovo. (*IMark* například odmítne zahrnout do rejstříku slovo \TeX , jelikož \TeX obsahuje `\`, což není písmeno.) Takové věci je nutné udělat ručně. Pokud by to *IMark* měl umět, bylo by nutné použít mnohem složitější, a tedy i pomalejší algoritmus.

A z tohoto bodu pokračujeme jako obvykle. Spustíme např. \LaTeX , potom `makeindex` a jsme hotovi (skoro).

3. Syntaxe

`IMARK` [přepínače] příkaz seznam [zdrojové soubory]

(*seznam* je soubor obsahující seznam slov, které mají být označeny ve zdrojových souborech pro \TeX)

Přepínače (předchází jim `,-'` nebo `,/'`):

- e** použij anglické řazení (implicitní je české)
- c** *cmd* nastav indexovací příkaz na *cmd* (implicitně `index`)
- o** *ext* nastav příponu výstupních souborů na *ext* (implicitně `new`)

Příkazy:

- h** zobraz nápovědu (anglicky)
- b** vytvoř nový seznam ze slov obsažených ve zdrojových souborech
- a** přidej do seznamu slova obsažená ve zdrojových souborech
- m** označ slova ze seznamu ve zdrojových souborech (*pro nové verze souborů použije příponu a indexovací příkaz nastavený pomocí přepínačů*)

4. Příklad

Celý proces ilustrujeme následujícím příkladem. Mějme zdrojový text pro \LaTeX nazvaný `test.tex`:

```
\documentstyle{article}
\begin{document}
Pro animaci je použita spline
interpolace.
\end{document}
```

Zavoláme-li *IMark* na tento soubor (`IMARK b test test`), dostaneme soubor `test.lw` se seznamem slov:

```
% List of words for IMARK, (C) Jan Kybic 1993
animaci
article
begin
document
interpolace
je
použita
spline
Pro
% End of the list
```

Tento soubor ručně upravíme na tvar:

```
% List of words for IMARK, (C) Jan Kybic 1993
animaci animace
interpolace
použita
spline interpolace!spline
% End of the list
```

Ještě jednou zavoláme *IMark* (`IMARK m test test`) a ten vytvoří ze souboru `test.tex` soubor `test.new`:

```
\documentstyle{article}
\begin{document}
Pro animaci\index{animace} je použita\index{použita}
spline\index{interpolation!spline}
interpolace\index{interpolace}.
```

```
\end{document}
```

A je to!

5. Podpora češtiny

IMark používá implicitně české třídění (anglické zapneme pomocí `-e`). Třídění je dvouprůchodové, v současné verzi řadím velká písmena až za malá, pokud by to někomu vadilo, stačí ve zdrojovém textu přepsat jedinou tabulku (`cstab` v modulu `csort.cpp`). Stejnou tabulku je nutné přepsat, chceme-li použít jiné kódování než Kamenických. V této souvislosti je nutné upozornit ještě na jednu vlastnost programu *IMark* (dodávám, že stejně úmyslnou jako ta předchozí) — začíná-li slovo velkým písmenem, např. na začátku věty, nevidí *IMark* žádnou souvislost s tímž slovem začínajícím malým písmenem.

6. Programátorské poznámky

IMark byl napsán v Borland C++ verze 2.0. Tvoří ho asi 600 řádek zdrojového textu (26 kB). Objektový přístup byl použit v maximální míře. Program používá objektovou knihovnu CLASSLIB, která poskytuje mnoho základních algoritmů pro práce s daty. Není ovšem naprogramována příliš efektivně, a proto bylo nutné v konečné verzi některé algoritmy přepsat, což však díky objektovosti nebylo obtížné. Připouštím, že výsledný program není tak malý a rychlý, jako by mohl být vzhledem k režii při práci s objekty, ale — něco za něco.

Rutiny českého třídění jsou obsaženy v jediném modulu a dají se proto použít i v jiných programech.

7. Závěr

Program lze získat pomocí anonymous ftp z `ftp.muni.cz` — je tam uložen v souboru `imark.zip`, což je zdrojový text, dokumentace a spustitelný MSDOS `imark.exe` soubor, vše zabalené PKZIPem. Přenos pod UNIX asi nebude jednoduchý, bylo by nutné buď přepsat, nebo přenést knihovnu CLASSLIB. Pod UNIXem však existují mocné nástroje, ze kterých se dá něco podobného jako *IMark* zbastlit. Pokud by byl přesto zájem, jsem ochoten se o konverzi pod UNIX časem pokusit.

Program dávám k dispozici jako *public domain* a věřím, že bude někomu užitečný. Uvítám samozřejmě jakékoli připomínky, návrhy na zlepšení a opravení chyb. Pokud použijete můj nápad nebo zdrojový text

jako základ vašeho programu, který bude jakýmkoli způsobem lepší než *IMark*, dejte mi, prosím, vědět.

8. Literatura

- [1] *Lamport Leslie*, MakeIndex: An Index Processor for L^AT_EX.
- [2] Dokumentace k programu MakeIndex.
- [3] *Wágner Zdeňek*, CsIndex v. 2.11 – Česká/Slovenská implementace MakeIndexu.
- [4] *Wágner Zdeňek*, Tvorba rejstříku, Zpravodaj C_STUGu 4/92.

Jan Kybic

Prosecká 681

190 00 Praha 9

e-mail: kybic@earn.cvut.cz

kybic@csearn.bitnet

Pár poznámek vděčného uživatele

Dovolím si drobný komentář k právě popisovanému programu, protože informace o něm mě zastihla právě v okamžiku, kdy se mi *velmi* hodil. Občas se totiž najde autor, který ví, co chce v rejstříku mít, ale nestíhá ani další dokončovací práce na knize, natož aby ještě jednou celé dílo pečlivě přečetl (i když už je půlka v tiskárně) a všechna slova odkazovaná v rejstříku v textu vyznačil. V případě jmenného rejstříku to dokonce snad ani zvláštní problémy nenes, zvlášť vyskytuje-li se většina jmen jen jednou či dvakrát. V takovém případě velice pomůže, můžeme-li rychle zjistit, v jakém tvaru se příslušná slova v textu vyskytují, a pak si je nechat automaticky označit rejstříkovou značkou.

Příznám se, že stejně jako *IMark* píšu tuto značku obvykle rovněž na konec slova (dokonce jsem si na to napsal kopírovací makro v qeditu), ale z hlediska správného odkazu by asi bylo vhodnější psát odkazy před slovo, protože jen tak můžeme zabránit tomu, aby někde nezůstalo dole na stránce rozdělené slovo, na které bude rejstřík odkazovat následující stranou! Rejstříková makra v T_EXbooku (*manmac*) v sobě samozřejmě zahrnují zákaz dělení odkazovaných slov.

Co se týče třídící funkce zmíněného programu, nebylo by marné, kdyby bylo možno požadovat vypsání a utřídění jen těch slov, která začínají zvoleným písmenem (popříp. více vybranými písmeny). To by navíc notně zjednodušilo práci s rozsáhlejšími texty, protože počet slov, jež program

dokáže zpracovat v rámci jediného souboru, není z pochopitelných důvodů nijak závratný.

A nakonec pár „drobných nedostatků“ té verze, kterou jsem si stáhl z brněnského archivu v době, kdy jsem článek redigoval: program nevypisuje slova obsahující písmena s přehláskou (v kódu Kamenických, jiné kódování jsem nezkoušel) a plete si velká písmena O a P (to že třídí zvláště velká a zvláště malá počáteční písmena je někdy výhoda, někdy nevýhoda).

Karel Horák

BMV — Jak na PostScript pod Linuxem

JAN KYBIC

Mnoho z nás již pro $\text{T}_{\text{E}}\text{X}$ ování (a nejen pro $\text{T}_{\text{E}}\text{X}$ ování) používá UNIX. Na počítačích třídy PC se často jedná o variantu zvanou **Linux**. K tomu nás vede kromě jeho nepopíratelné kvality i to, že je ho možné získat zadarmo, například pomocí `ftp`. Chcete-li se o **Linuxu** dozvědět více, zkuste si přečíst například *Bajt* číslo 46 nebo 41 z roku 1994.

Mnoho z nás také používá výstup z $\text{T}_{\text{E}}\text{X}$ u do PostScriptu. Ptáte-li se proč, zalistujte několika posledními bulletinů, články o PostScriptu se to tam jen hemží. Převod souboru `dvi` do PostScriptu za nás udělá program `dvips`. Otázkou ovšem zůstává, jak si výsledný PostScriptový soubor prohlédnout. Ano, i pod UNIXem existuje volně šiřitelný PostScriptový interpret zvaný Ghostscript. Sám o sobě ale příliš komfortní není a o interaktivitě si můžeme nechat jen zdát. Zvětšení je možné pouze jediné — většinou zvolíme takové, aby se na obrazovku vešla celá stránka, pak je ale text příliš drobný, než aby se dal číst. Svou základní úlohu Ghostscript splní dobře — PostScriptový soubor převede na bitovou mapu, kterou pak podle naší volby buď zobrazí, pošle na tiskárnu nebo uloží do souboru. Co dělat, abychom se alespoň přiblížili ke komfortu Mattesova `dvicsr`?

1. Použijeme PostScriptový prográmeček `ps_view`. Jeho velkou výhodou je univerzalita — měl by fungovat všude, kde funguje Ghostscript. (Asi jsem neměl dostatek trpělivosti a znalostí PostScriptu, ale nepodařilo se mi všechno rozchodit. To, co fungovalo, bylo (zákonitě) pomalé, neboť Ghostscript musí generovat bitovou mapu vždy znovu při každém našem povelu.) Podle mých informací se teď `ps_view` zabývá několik lidí, takže veškeré problémy by měly být brzy odstraněny. (Před několika dny se na síti objevil balík `gsv4dos` pana Mikulíka, obsahující dávkový soubor pro DOS, který by měl umožnit používání `ps_view` bez dalšího experimentování.)
2. Použijeme GhostView — *front-end* pro Ghostscript běžící v prostředí X Windows. Umí asi všechno, co bychom si přáli. Bohužel jsou však jeho požadavky na výkonnost počítače vysoké, asi ještě větší než u jeho pod MS Windows běžícího bratříčka. Pokud máte 486 DX2 běžící na 66 MHz, 16 MB RAM a kartu s grafickým akcelerátorem, běží všechno radost pohledět. U slabšího stroje je to ovšem utrpení, nejste-li zvlášť trpěliví.
3. Použijeme BMV.

BMV* je program, který jsem napsal s pomocí několika dalších lidí na Internetu. Moje původní verze pracuje pod **Linuxem** a panu Williamu Baderovi se ho podařilo přenést i pod SCO Unix. Hlavní výhoda BMV je, že ke své činnosti nevyžaduje X Windows, a běží tudíž slušnou rychlostí i na slabším počítači. BMV můžete získat z archivu `sunsite.unc.edu`:

`/pub/Linux/apps/graphics/viewers/bmv-1.1.tar.gz,`

nebo z některého z jeho zrcadel, například `ftp.muni.cz`.

Instalace BMV podle přiloženého návodu by měla být bezproblémová. Pokud nemáte nějakou zvláštní konfiguraci adresářů, mělo by stačit napsat `make`, jinak bude nutno upravit `Makefile`. Změny `default` parametrů můžete provést v souboru `bmv.h`.

Spustíte BMV příkazem `bmv <soubor.ps>`, kde `<soubor.ps>` je PostScriptový soubor vzniklý například překladem z `dvi` pomocí `dvips`. BMV zavolá Ghostscript a požádá ho, aby zpracoval PostScriptový kód první stránky vašeho souboru. Jakmile Ghostscript vytvoří bitovou mapu, BMV vám z ní na obrazovce zobrazí výřez. Tímto výřezem jde pohybovat klávesami `h`, `j`, `k`, `l` (tradice z editoru `vi`), je také možné

* Název původně znamenal „**Bit**map **V**iewer“, protože umožňoval prohlížení bitmap ve formátu PBM.

bitmapu zvětšovat a zmenšovat (*zoom in/out*) klávesami +, -. To vše jde už velmi rychle, neboť bitmapa je uložena v souboru na disku.

Pokud se vám na stránce něco nelíbí, přepněte do jiné *virtuální konzoly* (zase jedna z vymožeností některých UNIXů — počítač se nám virtuálně „rozdělí“ na několik nezávislých počítačů, mezi nimiž přepínáme klávesovými kombinacemi) stiskem s. Tam může běžet třeba textový editor, s jehož pomocí opravíme zdrojový text pro T_EX. Nejsme-li si jisti, jak výsledek vypadal, přepneme znovu na konzolu, kde běží BMV, a podíváme se. Je výhodné opravit co nejvíce chyb najednou a pak zdrojový text znovu přeložit. BMV musíme bohužel při každé změně ps souboru spustit znovu.

O stránku vpřed/vzad se posunete pomocí kláves n/p, stiskem q BMV opustíte.

BMV je stále ve fázi vývoje. Budu rád, když ho vyzkoušíte a napíšete mi své připomínky a komentáře.

27.1. 1995

Jan Kybic <kybic@earn.cvut.cz>


Písanie cs textov trochu ináč

LADISLAV BITTÓ

Ja a diakritika na PC

Hoci osembitové fonty sú už dávno hotové, ja som naďalej (až do roku 1995) písal po starom. Väčšinu textov som musel písať po anglicky a to málo slovenského som pohodlne zvládol so svojím Latexmanagerom. Urobil som v editore pár vecí, aby to išlo čo najjednoduchšie a čo s najmenším možným počtom stlačení. Nemusel som prepínať na iné klávesnice, lebo všetky znaky, ktoré bolo treba vysvietiť na obrazovku, boli na štandardnej klávesnici. Ale čím ďalej tým viac ľudí prešlo na osembitové fonty aj v mojom okolí. Je to jasné, má to svoje výhody, o ktorých už bolo popísané dosť na to, aby ich každý využil na písanie cs textov. Na ústave je nás už pomalý stávala pre mňa neúnosná. Niektorí kolegovia písali tak, ja zasa onak a potom sa to muselo dávať dokopy atď.

Moje prvé skúsenosti s diakritikou

Začal som teda aj ja písať diakritické znaky na obrazovku. Už po prvých textoch som si uvedomil, že dajako som z toho dosť psychicky unavený. Je to dosť nelogické. Po anglicky píšem v pohode, hoci neviem po anglicky (pár výrazom už ale rozumiem, hlavne čo sa týka PC), po slovensky som sa naučil celkom obstojne, takže navyše aj rozumiem tomu, čo píšem. Uvedomil som si teda, že to spôsobuje zložitý zápis diakritických znakov. V podstate som mal dve možnosti. Buď budem písať ako ostatní a postupne na ten zložitý zápis diakritických znakov navyknem, alebo niečo vymyslím, čo bude možno pre mňa (a samozrejme aj pre kolegov) lepšie. Pre prvú možnosť hovoril fakt, že tak je to v T602, ĽED a možno aj v iných špeciálnych editoroch pre cs písanie. Takže čo ja už vymyslím, iba budem vymýšľať samé blbosti. Naproti tomu ma stále hneval fakt, že niektoré malé písmená musím zapísať na jedno stlačenie, niektoré na dve a niektoré dokonca až na tri stlačenia rôznych kláves. Navyše ani jedno veľké písmeno sa nedá zapísať na dve klávesy. Som dostatočne lenivý na to, aby som o tom aspoň intenzívne nepremýšľal. Spomenul som si na podobné (ale len podobné) problémy pri ruštine. V škole ma síce tri roky učili po rusky, ale teraz neviem ani tolko po rusky, čo po anglicky. Mozog človeka je po dlhých rokoch písania na PC (predtým iné počítače) podvedome navyknutý na určité rozloženie znakov na klávesnici. Rusi majú takmer celú klávesnicu inú. Začal som písať veľmi jednoduchým systémom — jeden riadok za jednu hodinu. Po práci som rozmýšľal, že čo by sa tu dalo urobiť. Žiadalo sa prehodiť takmer všetky znaky. Lenže ako to urobím, keď všetky ovládače, ktoré nám kolega zohnal, boli tak nastavené ako boli a nedalo sa s tým pohnúť. Napadla ma jednoduchá vec. Napíšem do jedného súboru všetky znaky štandardnej klávesnici pekne po poriadku od najmenšieho kódu po najväčší (32–127). Potom som sa trochu pohral, aby som dostal všetky ruské znaky (malé aj veľké) pod ekvivalentné naše. V editore som pridal pár riadkov, aby som mohol prepnúť na túto klávesnicu. Odrazu písanie išlo ako z gulometu. Možno radosť z toho mali aj kolegyne, ktoré tiež začali odrazu písať rýchlejšie, navyše s ďaleko menšou psychickou záťažou. Dokonca jedna kolegyňa, ktorá vedela písať na ruskej klávesnici (bola jediná, ktorá neprešla okamžite), po dvoch či troch týždňoch začala písať podľa môjho návrhu. Tých ruských textov nemáme až tak veľa a sú to väčšinou len hladké texty (bez rovníc a tabuliek). Čísla mám v editore všetky aj pod , takže nemuseli sme preplínať klávesnice.

Na toto všetko som si spomenul. Svoj editor som už dávno prispôbil, aby fungoval aj s našimi diakritickými znakmi (kvôli kolegom). Môžu byť dve klávesnice s ľubovoľnými znakmi na ľubovoľnom mieste. Nastavil som si teda diakritické znaky po svojom (nad všetky naše malé adekvátne veľké). Nastavil som si tak 15 malých a 15 veľkých písmen. Ostatné písmená sa v slovenčine nevyskytujú, alebo sa použijú raz na každej piatej strane. Takto to išlo hneď o poznanie rýchlejšie a pohodlnejšie. Na hladký text je to povedzme fajn. Už ma upokojilo vedomie, že veľké písmená sú nad malými a všetky malé stlačím na jedno ťuknutie, veľké na dve ťuknutia (presne tak ako Angličania). No ale slovenský text píšeme často aj s rovnicami, tabuľkami. Je jasné, že „ukradnuté“ znaky chýbajú. Ako to riešiť. Buď budem prepínať klávesnice (to veľmi nerád robím), alebo ich dám do formúl. Svoj editor som napísal v prvom rade preto, aby mi podstatne uľahčoval generovanie textov. Takže ja horné znaky klávesnice prakticky nepoužívam (už dobré dva roky). Každý asi vie, že načo sú v $\text{T}_{\text{E}}\text{X}$ u tie horné znaky na štandardnej klávesnici. Vyrobil som si dostatočné množstvo formúl (v mojom editore ich môže byť vyše 2 600). Zo začiatkov si spomínam, že každá rovnica ma priam desila. Teraz ich už píšem v pohode (často sa pristihnem, že dve susedné veľmi podobné rovnice znova generujem, namiesto prekopírovania a následného opravovania rôznych rozdielov) a celú stranu plnú rovníc zapíšem rýchlejšie než hladký text. Teda ja sa už zaobídem bez horných znakov a tie „ukradnuté“ znaky mi až tak nechýbajú. Nájdu sa ľudia, ktorí si vyrobili dostatočné množstvo formúl, ale väčšina väčšinou len píše a píše. Ako by sa dalo pomôcť týmto ľuďom. Na prvý pohľad nijako. Keď chcú jednoducho písať cs znaky, potom musia prepínať a potom zasa naspäť a potom za chvíľu zase prepínať... Alebo nebudú prepínať a cs znaky budú zložito generovať, čo je myslím ešte horšie.

Niečo sa dá vymyslieť

Uvedomil som si, že dosť ľudí píše všetky diakritické znaky na dve ťuknutia, alebo veľké na tri ťuknutia (to som im už dávno zabezpečil). V mojom okolí to odhadujem tak na 30–35%. Prečo tí ľudia píšu tak. Prečo nedajú prednosť aspoň tým najčastejšie používaným malým písmenám, ktoré by mohli zapísať na jedno ťuknutie. Jednému vadí, že každé štvrté písmeno (diakritické) sa ináč zapisuje, druhému sa nepáči, že musí stlačiť *shift* a potom niečo. Na príčine je teda zložitost' a klávesa *shift*. Bolo mi to jasné hneď, keď som si rozanalyzoval písanie. Čo vlastne robí mozog, keď

píšeme? Búchame na znaky na klávesnici jeden za druhým. Pohoda. Za chvíľu je pohody koniec. Musíme buchnúť na jednu klávesu, ale nesmieme ju pustiť. Musíme počkať na nej aspoň jedným prstom (navyše aj mozgom), kým nestlačíme aj druhú. Potom ich môžeme pustiť. Ale keď nám mozog trochu navykol na pohodu, tak vyšle podvedomý signál a prst zdvihne skôr, než druhý stlačíme. Výsledok je pre nás neuspokojivý. Občas sa vám to stane, však? Napriek tomu, že sme stlačili dve klávesy, zapíšeme jedno malé písmeno. A naopak, keď podržíme *shift* o niečo dlhšie (hlavne vtedy keď ďalšie písmeno je tiež na opačnej strane), zapíšeme dve veľké písmená. To asi tiež poznáte. Najlepšie si to môžeme uvedomiť, keď všetky malé písmená dáme do jednej skupiny a označíme ich bodkou. Keď stlačíme *shift* a písmeno označíme písmenami II. Teraz, keď stlačíme iba malé písmená, vznikne rad znakov zo samých bodiek. Keď ale stlačíme už aj *shift*, vznikne nám úplne chaotický rad dvoch znakov, napr. II...II...III...III...III...II...II...II... Je zrejmé, že samé bodky sa ľahšie zapisujú. Teraz si predstavte, že dostanete za úlohu zapísať celú obrazovku s bodkami, alebo zapísať celú obrazovku podľa druhej možnosti. Ten, kto si vyberie druhú možnosť, klobúk dolu.

A potom je tu ďalšia sťažujúca okolnosť pre náš mozog pri písaní diakritických znakov. Angličania majú dve možnosti na zápis všetkých písmen. My ale potrebujeme na zápis všetkých písmen až osem rôznych spôsobov. Navyše niektoré malé písmená musíme zapísať na jedno stlačenie, niektoré na dve stlačenia a niektoré dokonca na tri rôzne stlačenia. Podľa mňa je pre mozog ešte aj to ťažké, keď používame 9–10 malých písmen, ktoré sú obvykle na cs klávesnici. Aspoň môj mozog nad nimi stále čaká veľké, ktoré však tam nie sú. Sú niekde úplne inde. Takže *shift* a spomenuté okolnosti nám komplikujú život. Je možné to obísť aspoň pre cs znaky. To nie je až taký problém. Postavme veci tak, že budeme mať k dispozícii súčasne dve klávesnice. Jedna, s ktorou obvykle pracujeme, a druhá pod jednou mŕtvou klávesou. Aby nám nechýbala tá klávesa, môžeme na ňu naprogramovať pár kombinácií (napr. na ; by stačila medzera a enter). Teda predstavujem to tak, že keď stlačíme ; nepíše sa nič, ale potom po medzere sa vypíše aj ; a aj medzera. Takto všetky diakritické znaky môžeme mať tam. Ľudia, ktorí pracujú tak, že ani za svet nechcú prepínať medzi klávesnicami, to majú vybavené. Používajú štandardnú klávesnicu a všetky naše znaky majú tam (teda pod mŕtvou klávesou). To už je úplne na každom, kde čo dá. Kombinácií je veľa. Za chvíľu som pridal ešte jeden mŕtvy znak, aby som tam mohol dať aj veľké písmená bez diakritiky. Takto *shift* ani nemusíme používať (ale

samozrejme môžeme), lebo všetky naše aj všetky anglické veľké písmená sme uložili na dolnú polohu. Hornú polohu teda nemusíme ani využiť. Keby som toto všetko neurobil a neskúšal tak písať, asi nikdy by som neprišiel na ešte lepšie písanie.

Keď je tam, nemôže byť aj tu?

Zrazu mi prebehlo myslou, že úplne perfektné by bolo, keby som mohol zapisovať veľké písmeno tak, že stlačím príslušné malé dvakrát za sebou. Najprv sa mi to zdalo veľmi sprosté (určite aj vám), ale po uvedení si pár skutočností mi bolo jasné, že to pôjde a možno to bude veľmi dobré. Naprogramovať to nie je problém. Táto myšlienka bola veľmi dráždivá, hlavne v kontexte s diakritickými znakmi. Zrazu som si uvedomil, že je asi veľmi málo slov, ktoré sa začínajú s dvoma rovnakými znakmi. Nevedel som si spomenúť ani na jedno slovo. Prezrel som slovenský slovník (25 000 slov) a na moje veľké prekvapenie som nenašiel ani jedno jediné také slovo. Prezrel som aj anglický (tiež asi 25 000 slov) a tam som našiel iba dve slová. A už som vedel, že je to dobré. Ďalej som ani nehľadal, lebo som si uvedomil, že vlastne môže byť aj milión takých slov, je to jedno. Variácií je len toľko, koľko je malých písmen. Na každú jedna možnosť (aa, bb ...). To sa dá veľmi ľahko obísť. Dám všetky do formúl a mám ich na dve ťuknutia. Ale ja sa napríklad ani nepamätám, či som niekedy vôbec potreboval aspoň jednu dvojicu. Toto je teda úplne dokonalé pravidlo, treba to využiť. Ďalšie pravidlo, ktoré to umožňuje, je tiež celkom obstočné: drvivá väčšina slov (99,9 %) sa zapisuje s jednou z týchto troch možností:

1. celé slovo sa skladá len z malých písmen,
2. prvé písmeno je veľké a ostatné malé,
3. celé slovo sa skladá len z veľkých písmen.

Zvyšok sú všelijaké skratky a ostatné vymoženosti. Jeden človek používa takýchto skratiek určite málo. Buď ich dá do formúl a vytiahne ich na dve ťuknutia, alebo ich zapíše normálne so *shiftom*. Takže ideme písať týmto spôsobom bez *shiftu*. Všetky tri možnosti môžeme ošetriť tou istou podmienkou (ináč by to ani nešlo). Keď stlačíme za sebou to isté malé písmeno (aj diakritické samozrejme), prepíše ho na veľké, keď je na prvom stĺpci, keď je pred ním medzera alebo niektorý zo znakov $\{ \sim \}$. V opačnom prípade ho zapíše ako normálne, čiže dvakrát. Po skúsenostiach môžeme pridať zložitejšie algoritmy. Pre 3. možnosť musíme urobiť funkciu, ktorá nám naspäť po vyššie uvedené

znaky prepíše malé písmena na veľké a opačne. `[Caps]` by nebol vhodný pre diakritické veľké písmená, navyše sa musí aj vypnúť. A môžeme veselo písať text bez *shiftu* veľmi jednoducho. Možno, že pre profesionálne pisárky to je jedno, ale pre takých obyčajných ľudí, ktorí sem tam píšu texty, pre ľudí, ktorí sú radi, že dajaký znak našli na klávesnici (sem patrí aj ja), pre ľudí, ktorí si uvedomia, že *shift* naozaj rozhodí rytmus písania, to môže byť celkom vhodné písanie. Samozrejme *shift* funguje naďalej tak, ako má. Moje prvé skúsenosti sú veľmi príjemné. Ten prechod je takmer okamžitý. Sem tam ešte mozog zaplujie a stlačím *shift*, ale to sa napraví. Píšem v pohode. Všetko, čo potrebujem k slovenskému písaniu, mám k dispozícii. Čísla môžu byť ešte aj na numerickej klávesnici (tá je ale od ruky, ja ich mám aj pod `[Alt]`, aj pod jednou z tých dvoch mŕtvych kláves), všetky horné znaky môžu byť pod *shiftom* (ja ich už používam cez mŕtvu klávesu a nie cez formuly ako doteraz) a nakoniec ešte tam zmestia aj slovenské znaky. Samozrejme tých variácií je viac, každý si to nastaví tak, ako mu bude najviac vyhovovať. Toto nám teda podstatne zmenší psychickú únavu. Zo všetkého najdôležitejšie je, že teraz máme všetky písmená (malé a veľké) rovnakým spôsobom k dispozícii navyše spolu s ostatnými znakmi.

Nemôžeme zabudnúť ani na fyzickú únavu. Týmto štýlom sa celková dĺžka pohybu našich rúk a prstov podstatne zníži. To najviac dokumentuje nasledovný príklad. Máme zapísať dva znaky Ju jedným prstom (tým istým). Pri normálnom písaní musíme stlačiť nasledovné: `[Caps][J][Caps][U]` (inak to nezapíšeme). Keď počítame od prvého stlačenia, tak náš prst musí prejsť vzdialenosť zhruba 40 cm, pričom musíme stlačiť 4 klávesy. Pri tomto novom písaní stlačíme `[J][J][U]` (teda iba tri klávesy), pričom prst musí prejsť iba vzdialenosť 2 cm. Samozrejme keď máme prejsť jeden meter alebo päť metrov, to je nám ešte takmer jedno. Ale keď musíme prejsť 10 km alebo 50 km, to už asi nie je tak jedno.

Ako ďalej

Kolegu Šurdu zo susedného ústavu, vidiac tieto možnosti, chytila nasledujúca myšlienka. So špeciálnym ovládačom (ktorý umožňuje nielen meniť, ale aj zmeniť všetky znaky) vytvoriť špeciálne znaky pre \TeX . Niečo sa nájde medzi kódmi 174–255, ale takých 50 sa ešte dá vytvoriť. Napr. namiesto `\alpha` by sa na obrazovku vypísal znak α . Bolo by to hneď prehľadnejšie. Tie znaky by boli k dispozícii cez mŕtvu klávesu

(alebo aj cez formulu). Treba len donútiť $\text{T}_{\text{E}}\text{X}$, aby to akceptoval. On sa o to pokúsi.

Pri písaní týchto riadkov som mal za sebou len pár dní od nového spôsobu písania. Teraz už viem, že sa oplatí pre jednoduchosť bojovať. Človek často príde na jednoduché riešenie po krkolomných cestách a potom sa takmer hanbí pred sebou, že tak strašne zložito mu to trvalo. Už ma napadajú ďalšie zjednodušenia.

Editor mi už dávno píše automaticky vlnky namiesto medzier, keď je pred medzerou skzv alebo SKZV, ďalej po bodke, keď pred bodkou je jeden znak a pred ním medzera. Toto doplním o zložitejšie algoritmy.

Tie naše krikľavé skratky (v ktorom sa striedajú malé a veľké písmená) by napr. editor mohol aj poznať a potom to bude paráda. Napr. zapíšeme (teraz už viem, že to je pre mozog príjemnejšie, keď nemusím písanie dajako prerušiť, trebárs tým, že aktivujem formuly): profdr (s medzerou alebo s jedným zo znakov \cdot ({{ ~ na konci, ako nám to bude vyhovovať) a editor len tak sám od seba nám to prepíše na: Prof. ~Dr. ~.

Editor mi už dávno automaticky prepína v matematickom móde (jeden dolár alebo dva doláre). Urobím to na všetky $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ové prostredia, ktoré sú pre matematiku. Zase prejdem na iba otváranie prostredí (bez okamžitého zatvorenia). Čo s tým ušetríme, že sa nám hneď prostredia zavrú a kurzor sa nám pekne nastaví dakde. Keď končíme písanie v tom ktorom prostredí, na konci musíme z neho aj vyjsť, aby sme mohli ďalej pokračovať. No a nato potrebujeme obyčajne trikrát ťuknúť do klávesnice. Urobím to tak, že otvorím prostredie, ktoré bude viazané napr. na prepnutie do zvolenej klávesnice (to už s dolárom dávno mám), na zapamätania otvoreného prostredia (odpadne mi pamätať, že čo som otvoril, kvôli čomu som prešiel na okamžité zatváranie). Tým pádom by som mohol vždy na to isté ťuknutie (stačilo by aj jedno) zavrieť posledné otvorené prostredie (ďalšou by sa mohli uzavrieť trebárs naraz posledné dva). Pritom by mohol pekne vrátiť klávesnicu, aká bola predtým, a nastaviť, aká má byť na nižšej úrovni (keď je otvorených viac), atď. To by bolo tiež fajn.

Nemôžem sa zmieriť ani s tým, že som stratil čísla na štandardnom mieste. Niečo by sa možno dalo urobiť. Napr. mám zapísať text ... rok 1994 bol zlý. Namiesto toho som zapísal toto: ... rok dŕiíc bol zlý. Pozriem sa potom na obrazovku a tam budú čísla. Editor tiež príde na to, že to je nezmysel a správne to prepíše len tak od seba na čísla. V slovenčine od troch takých znakov (nie je tam ani jeden nediakritický znak) vyššie je všetko nezmysel, na dve by sa možno našlo pár výnimiek, no a na jedno

písmeno môže byť iba 10 prípadov. Ale napadá ma iba jedno, ktoré sa aj používa (č., čo by mala byť skratka pre číslo).

Teraz už píšem prakticky bez *shiftu* po slovensky a aj po anglicky. Potom pri písaní e-mailových súborov po slovensky musím dávať pozor, lebo už mozog navyká na diakritiku. Musím urobiť funkciu, ktorá odstráni diakritické znaky a nahradí ich adekvátnymi anglickými písmenami, aby sa ľahko prenášal aj v sieti.

Možno moje myšlienky niekoho zaujmú a bude hľadať aj on pohodlnejší spôsob cs písania. Toto všetko teda spôsobila jedna klávesa.

Takže príjemné

Ladislav Bittó

Geofyzikálny ústav SAV

842 28 Bratislava

geoflabi@savba.sk

Zápis z valného shromáždění Československého sdružení uživatelů $\text{T}_{\text{E}}\text{X}$; datum konání: 14.1. 1995

- V první části vystoupil předseda sdružení Karel Horák se zprávou o činnosti. Nejprve informoval o tom, že sdružení koupilo 20 CDROMů (4all $\text{T}_{\text{E}}\text{X}$), jejichž nákup byl dohodnut na předchozím valném shromáždění. Zásilku přinesl přímo na shromáždění.
- Valné shromáždění po delší diskusi odhlasovala většinou hlasů, že tyto CDROMy se budou prodávat individuálním členům za cenu, která je rovna (po zaokrouhlení) ceně, za kterou byly CDROMy ζTUGem pořízeny. Pro tuto zásilku byla stanovena cena 650 Kč za kus. Za stejnou cenu mohou CDROMy koupit kolektivní členové, kteří se stali členy až letos, tj. netýkaly se jich členské příspěvky ve výši 4 000 Kč.
- Dále valná hromada odhlasovala většinou hlasů (4 hlasy proti), že těm kolektivním členům, kteří někdy zaplatili příspěvek ve výši 4 000 Kč a jsou stále našimi členy, bude jeden kus CDROMu k dispozici zdarma.

Z této dodávky budou dány CDROMy jen těm kolektivním členům, kteří o ně vysloveně projeví zájem na minulém shromáždění ζ TUGu a figurují na seznamu žadatelů.

- Valná hromada uložila výboru ζ TUGu objednat dalších 20 CDROMů, aby mohli být uspokojeni další zájemci.
- Karel Horák informoval o objednávce cca 25 knih od Addisona Wesley u holandské pobočky. Fakturu, která došla na tuto objednávku, ζ TUG ještě neproplatil, protože se výbor nebyl schopen dohodnout, zda jsou ceny přijatelné. Ceny odpovídají tzv. růžovému katalogu, jsou místo o \$2 až \$7 levnější a poštovné vychází cca \$3 na knihu. Po diskusi, která vyzněla zhruba v duchu „existují SNAD levnější cesty nákupu těchto knih, než udává růžový katalog“ se nakonec dohodlo, že se knihy koupí. Petr Olšák dostal za úkol napsat email členům. Email má obsahovat jednotlivé tituly a ceny těchto knih. Členové pak mohou posílat objednávky na knihy. Tento postup znevýhodňuje členy, kteří nemají email, ale než by informace prošla bulletinem, to bychom se asi dlouho načekali.¹⁾
- Karel Horák velmi stručně referoval o dalších aktivitách ζ TUGu. Například o konaných kurzech $\text{T}_{\text{E}}\text{X}$ u v Jevíčku. Zmínil se, že má další typ na místo, kde by se kurzy mohly konat (Rychnov nad Kněžnou).²⁾ Protože ale nepadlo jméno osoby, která by takovou akci zařídila, nedá se předpokládat, že se akce v brzké době uskuteční.
- Dále Karel zmínil, že se ζ TUG zúčastnil výstavy neziskových organizací, kde měl pronajatý jeden stánek za 1 000 Kč, a vystavoval tam některé aktivity kolem $\text{T}_{\text{E}}\text{X}$ u. Stručná zpráva o výstavě je podána v bulletinu č. 3/94.
- Karel přiznal, že bulletiny vycházejí se zpožděním. Momentální situace je, že zrovna na valném shromáždění byly předány podklady pro tisk bulletinu č. 3/94. Členové už více než rok marně čekají na překlad Taylorovy přednášky, o které rozhodl výbor ζ TUGu, že by měla být zařazena do bulletinu. To by vyšlo na celé jedno, možná dvě čísla. Stále se hledají přispěvatelé, kteří by přispěli do bulletinu buď svým vlastním článkem, nebo překladem nějakého zahraničního

¹⁾ Tedy skutečně ano (mea culpa!). Leč hned po zaplacení téměř obratem knihy dorazily. Na jiném místě tohoto Zpravodaje vkládáme závazný objednávací listek, jehož vyplněním si můžete vybrané knihy objednat.

²⁾ Díky aktivitě profesora hradeckého gymnázia Přemysla Šedivého proběhnou *Dva dny s $\text{T}_{\text{E}}\text{X}$ em* v dubnu a v říjnu letošního roku — viz kalendář akcí JČMF v časopise PMFA 1/95.

článku. Padla zmínka, že by se v našem bulletinu mohl objevit Knuthův článek o virtuálních fontech.

- Karel informoval, že finanční rozvaha hospodaření $\mathcal{C}\mathcal{T}\mathcal{U}\mathcal{G}$ u je pro rok 1995 deficitní, tj. že to, co se pravděpodobně vybere na členských příspěvcích, nepokryje náklady na provoz $\mathcal{C}\mathcal{T}\mathcal{U}\mathcal{G}$ u (zvláště vydávání bulletinu). Nicméně informoval, že současný výbor s vědomím této skutečnosti snížil už loni členské příspěvky kolektivním členům na 1 000 Kč, protože má $\mathcal{C}\mathcal{T}\mathcal{U}\mathcal{G}$ poměrně zbytečnou finanční rezervu (cca 750 tis. Kč), takže se nyní bude odčerpávat z této rezervy.
- Karel dále přiznal, že současný výbor $\mathcal{C}\mathcal{T}\mathcal{U}\mathcal{G}$ u není příliš operativní a že na podzimním (zimním) shromáždění (zhruba za rok) budou provedeny nové volby do výboru. Pokud by se do výboru dostali noví lidé, asi by to $\mathcal{C}\mathcal{T}\mathcal{U}\mathcal{G}$ u prospělo. Proto již nyní apeluje na členy, aby se mezi sebou porozhlédli po vhodných kandidátech.
- V Následující diskusi vystoupil Petr Olšák s názorem na udělenou výroční cenu $\mathcal{C}\mathcal{T}\mathcal{U}\mathcal{G}$ u. Z celého kolektivu autorů $\mathcal{C}\mathcal{T}\mathcal{E}\mathcal{X}$ u, kterému byla cena za rok 94 udělena, totiž figurovalo ve zprávě o udělení ceny v bulletinu 2/94 jenom jeho jméno. Řekl, že si váží udělení této ceny, ale že ho mrzí, že ve zprávě o udělení ceny nebyl uveden plný seznam jmen tzv. „kolektivu autorů“, kterým je cena udělena. Důsledkem toho je, že řadový člen vlastně neví, kdo byl oceněn. Petr prohlásil, že vlastně ani on sám neví, koho pod pojmem „kolektiv autorů“ vlastně orgán, který uděluje cenu,³⁾ míní. Dále řekl, že uvedení jména oceněného v našem časopise je vlastně součástí ocenění a že finanční částka, která s cenou souvisí, je pro mnohé z nás až na druhém místě. Proto apeluje na orgán, který uděluje cenu (tj. na výbor $\mathcal{C}\mathcal{T}\mathcal{U}\mathcal{G}$ u), aby uveřejnil v nejbližším čísle bulletinu jmenovitý seznam „kolektivu autorů“, kteří jsou ocenění. Do doby, než bude tento seznam v bulletinu uveřejněn, odmítá Petr převzít finanční částku (20 tis. Kč), která má být mezi oceněné rozdělena. Samozřejmě, jak s touto finanční částkou kolektiv autorů naloží, je čistě věc tohoto kolektivu a nemusí o tom být veřejná zmínka. Pokud ale tento kolektiv není definován jmenovitým seznamem, pak to je špatné.⁴⁾

³⁾ a jehož je Petr sám členem

⁴⁾ Kdyby můj archiv nedočené či jinak odložené elektronické pošty nečítal desítky megabajtů, podíval bych se po Petrově výzvě krátce po schválení ceny výborem, ve které navrhl svým spolupracujícím kolegům několik způsobů (tuším aspoň tři), jak s cenou naložit. Tenkrát v tom měl poměrně jasno a rozhodně nechtěl po výboru, aby mu dodal seznam adresátů této výzvy. -kh

- V reakcích na tento příspěvek padl např. názor, že Petr Olšák vlastně prohlásil, že neví, s kým spolupracoval, a že i to je špatné. Nakonec byl akceptován požadavek na jmenovité zveřejnění oceněných tak, jak žádal Petr ve svém příspěvku.
- Po přestávce následoval blok přednášek. Jako první vystoupil Petr Sojka, který referoval o vývoji českých vzorů dělení generovaných patgenem. Výsledky jsou podle měření velmi dobré, dokonce řádově lepší než vzory dělení, které v současné době většina z nás ve volně šířeném TEX u provozuje. Na tomto díle spolupracoval s panem Ševečkem, který má ovšem s výsledným produktem obchodní úmysly. Chce tyto vzory prodávat firmám, které je implementují do „počestěných“ alternativ různých komerčních typografických systémů. Aby byla jejich práce prodejná, uvolní pro volné šíření ζTUG u pouze takovou variantu vzorů dělení, která je záměrně o něco horší než prodávaná varianta.
- V dlouhé diskusi k této přednášce padl zajímavě formulovaný názor pana Lhotky, že vlastně autoři tohoto dělení udělají jakousi práci navíc, aby toto dělení „zprasili“ a mohli dát veřejně k dispozici. Nezdá se mu to jako dobrá filosofie. J. Veselý navrhuje, že by tu skutečně nejlepší variantu vzorů dělení mohl ζTUG koupit. Karel Horák souhlasí a navíc dodává, že by to byly asi podstatně lépe investované peníze než při nákupu např. ζEdu . P. Sojka říká, že není vedoucím autorské dvojice a nemůže na tomto shromáždění měnit stanovisko, které autoři vůči šíření na půdě ζTUG u zaujali. Navíc prý jsou tyto „zprásené“ vzory dělení uživatelsky k nepoznání od originálních a pozná se to až při speciálních testech. Přítomní s tímto názorem nesouhlasili. Proč zrovna TEX musí být (vesměs) provozován s horšími vzory dělení než třeba QuarkXPress, který bude mít vzory dělení koupené od autorské dvojice? Je správné, aby pravděpodobně nejkvalitnější algoritmus řádkového zlomu neměl mít k dispozici nejlepší vzory dělení? Nebyl náhodou TEX a programy kolem něj nástrojem na přípravu vzorů dělení, které chtějí nyní prodávat? P. Sojka obhajoval stanovisko autorské dvojice argumentem, že by firmy od nich vzory dělení pro komerční programy nekupovaly, když by bylo obecně známo, že jsou volně k dispozici. J. Veselý prohlásil, že sice ζTUG nemá nástroje na kontrolu, zda se softwarem, šířeným pod jeho garancí, bude manipulováno podle licenčních pravidel, ale že se jedná o otázky legislativní a že nevidí důvod, proč by ζTUG nemohl s autory uzavřít smlouvu, vyhovující oběma stranám a to skutečně nejlepší dělení kou-

pit. Žádá proto P. Sojku, aby autorská dvojice znovu zvážila svůj postoj k možnosti prodeje vzorů dělení ζ TUGu.

- Dále (už nezbylo mnoho času) vystoupil v krátké přednášce Petr Olšák, který referoval o svém prográmku `a2ac`, který usnadňuje počestit PostScriptový font (tj. přidat virtuální akcentované znaky) na úrovni samotného PostScriptu a nikoli v rámci mechanismů $\text{T}_{\text{E}}\text{X}$ u. Program je tedy možné použít při práci s libovolným typografickým systémem, který umí pracovat s PostScriptovými fonty. $\text{T}_{\text{E}}\text{X}$ mezi takové typografické systémy patří. Program `a2ac` se použije v případě, že jsou k dispozici fonty dodané bez akcentovaných znaků a kresby akcentů jsou uloženy na samostatných pozicích. Program pracuje nezávisle na použitém kódování, protože se při sestavování kompozitních znaků vyjadřujeme pomocí symbolických jmen znaků a nikoli pomocí jejich kódů. Petr uvedl, že program lze snadno použít při přípravě virtuálních popisů k PostScriptovým fontům pro $\text{T}_{\text{E}}\text{X}$. Virtuální popisy mohou být zvoleny třeba v kódování, které je v souladu s kódováním CS-fontů.
- S poslední přednáškou vystoupil pan Libor Škarvada z Brna. Referoval o své práci na úpravě samotného programu $\text{T}_{\text{E}}\text{X}$, který se do spustitelné varianty kompiluje pod UNIXem ze zdrojového textu `.web`. Úprava nekoliduje s testem `trip`, tj. upravený $\text{T}_{\text{E}}\text{X}$ se v tomto testu neprojevuje odlišně. Úprava je tedy v souladu s copyrightovými požadavky autora $\text{T}_{\text{E}}\text{X}$ u. Úprava se samozřejmě realizuje nikoli uvnitř kódu `tex.web` (do tohoto kódu smí bez změny názvu zasahovat pouze jediný člověk na světě, kterým je Donald Knuth), ale v rámci tzv. změnového souboru, ve kterých lze ve webu dodefinovat speciální algoritmy, týkající se konkrétního operačního systému. Pro systémy typu UNIX udržuje tento změnový soubor pan Karl Berry a pan Škarvada navrhuje ve změnovém souboru udělat několik jednoduchých zásahů, které způsobí, že $\text{T}_{\text{E}}\text{X}$ ve svém vstupně výstupním rozhraní pro textové soubory (a terminál) dokáže konvertovat mezi různými kódováními rozšířených ASCII. Podle jakých kódování konvertuje, závisí na obsahu jisté proměnné prostředí systému (environment variable). Není-li tato proměnná nastavena, je chování $\text{T}_{\text{E}}\text{X}$ u shodné s implementací Karla Berryho. Tím se tedy i UNIXovým $\text{T}_{\text{E}}\text{X}$ ům dostává možnost pracovat s různými vstupními i vnitřními kódováními podobně, jako to umí `emT_{\text{E}}\text{X}` se svými konverzními `tcp` tabulkami. Práce byla zatím testována na SUN OS a Linuxu. Pan Škarvada asi požádá pana Karla Berryho, aby zahrnul toto „vylepšení“ do své, tj. standardně distribuované implementace $\text{T}_{\text{E}}\text{X}$ u pro UNIX.

- Posledním bodem programu byla diskuse k $\zeta\text{T}_{\text{E}}\text{X}_{95}$. Petr Olšák uvedl tři důležité důvody, proč by měl být proveden „upgrade“ $\zeta\text{T}_{\text{E}}\text{X}_{95}$. Jsou to

- zařazení lepších vzorů dělení (viz první přednášku);
- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2\epsilon}$;
- zařadit podporu práce s PostScriptovými fonty v č/s jazyce.

Dále uvedl, že v září volal o pomoc při takové práci na $\zeta\text{T}_{\text{E}}\text{X}_{95}$ v mail-listu. Očekával, že se nabídnou dobrovolníci a že do ledna budou nějaké hmatatelné výsledky. Dobrovolníci se nabídli, byl zřízen diskusní list „tvůrců $\zeta\text{T}_{\text{E}}\text{X}_{95}$ “, ale výsledky v podobě nové instalace $\zeta\text{T}_{\text{E}}\text{X}_{95}$ nejsou. Je to z několika důvodů:

- Každý si dělá na svém a práce se nepodařilo zkoordinovat.
- Čekalo se na zveřejnění nové „patch“ $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2\epsilon}$, tj. na prosinec 94.
- Existují lidé, kteří sice umějí, ale pro $\zeta\text{T}_{\text{E}}\text{X}$ se nechtějí angažovat s výmluvou, že nemají čas.

První důvod (nekoordinované aktivity) má například konkrétní podobu v tom, že se „tvůrci $\zeta\text{T}_{\text{E}}\text{X}_{95}$ “ na listě nedokázali dohodnout na společném vnitřním kódování $\text{T}_{\text{E}}\text{X}_{95}$. Petr požádal, aby se k této problematice vyjádřili i lidé z pléna. Padlo několik názorů na to, zda změnit kódování v $\zeta\text{T}_{\text{E}}\text{X}_{95}$ směrem ke světovému standardu (Cork), jehož základní nevýhodou je nekompatibilita s CM kódováním (Knuthovým, nejčastěji ve světě používaným). Tyto názory se velmi různily. Pokud se nenajde někdo, kdo udělá pro $\zeta\text{T}_{\text{E}}\text{X}_{95}$ takový kus práce, aby se stal autoritou, jehož rozhodnutí budou ostatní tvůrci $\zeta\text{T}_{\text{E}}\text{X}_{95}$ respektovat, pak se asi nepohneme z místa. Dalším problémem je zkoordinovat práci na různých platformách (UNIX, DOS, MS-WINDOWS). Tento problém předchozí verze $\zeta\text{T}_{\text{E}}\text{X}_{95}$ neznaly, protože se pracovalo jen pro DOS.

- Pan Wagner informoval o svých zkušenostech s implementací marker pana Zlatušky, které zatím řeší některou problematiku kolem C_{S} v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2\epsilon}$. Pozitivní zkušenosti má se starou verzí $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}_{2\epsilon}$ a s novou verzí bude manipulovat, sotva se vrátí ze shromáždění. Je ochoten výsledný produkt dotvořit do instalační podoby v $\zeta\text{T}_{\text{E}}\text{X}_{95}$ a přizpůsobit jakémukoli dohodnutému kódování.
- Petr Olšák prohlásil, že za situace, kdy jsou v tvůrčí skupině $\zeta\text{T}_{\text{E}}\text{X}_{95}$ roztržštěny názory na zásadní koncepce $\zeta\text{T}_{\text{E}}\text{X}_{95}$, zaujme pozici „mrtvého brouka“. Tj. vyčká, jak situace dopadne. Protože má práci na formátování své knížky o $\text{T}_{\text{E}}\text{X}_{95}$, odkládá své vlastní aktivity na $\zeta\text{T}_{\text{E}}\text{X}_{95}$ minimálně o dva měsíce. Podle toho, že plénum proti tomuto postoji vůbec nic nenamítalo, se dá soudit, že požadované novinky v $\zeta\text{T}_{\text{E}}\text{X}_{95}$

asi zas nejsou tak zcela potřebné. Je ale pravda, že to vede k tomu, že si některé věci bude muset udělat každý sám, což nejen tříští pracovní aktivity na CS v T_EXu, ale také to vede k dalším, hlubším rozdílům mezi jednotlivými lokalitami. Naše organizace by tu měla být v první řadě proto, aby takovým věcem v rozumné míře zabránila a umožnila pokud možno koordinaci prací a kompatibilitu výsledných instalací.

Zápis zachytil a vlastními slovy zaznamenal

Petr Olšák

Redakčně upravil

(zejména vyškrtáním nadměrně se vyskytujícího slova *exekutiva*)

Karel Horák

P.S. V době definitivní (pseudo)uzávěrky tohoto čísla (5. 4. 1995) máme v ruce korektury Petrovy knihy *Typografický systém T_EX* a před sebou jistě bouřlivou schůzku tvůrců ζ T_EXu (6. 4. 1995).

Knihy z nakladatelství Addison Wesley

Vážení přátelé,

přikládáme informaci o knihách z nakladatelství Addison Wesley, jejichž první zásilka objednaná na přelomu ledna a února zdárně dorazila do Prahy a téměř vzápětí se rozeběhla k příslušným zájemcům (velikost objednávky přesně tak, jak jsme informovali na lednové valné hromadě, byla dána naším odhadem skoro zaručeného zájmu v našem bezprostředním okolí. Volné svazky pak byly inzerovány na ζ T_EXovém listu). Po velmi dobré zkušenosti zveřejňujeme následující objednávku pro všechny naše členy s tím, že knihy budeme průběžně dle vašeho zájmu doobjednávat tak, aby počet knih v jedné zásilce neklesl pod pět (při tomto počtu vychází poštovné na jednu knihu zhruba na tři dolary).

Ceny jsou uvedeny podle předběžné faktury, případně podle katalogu, který máme k dispozici (druhá skupina). V druhém případě lze očekávat, že účtovaná cena za knihu bude ve skutečnosti o trochu nižší. Výsledné ceny (včetně poštovného) přepočítáme na koruny podle aktuálního kursu, zaokrouhlíme (nahoru) na desítky korun a prodáme podle pořadí došlých objednávek.

Zároveň se chystáme doobjednat CDROM 4all \TeX , proto je rovněž uveden v následující objednávce.

Z následujícího seznamu závazně objednávám:

Eijkhout: \TeX by Topic	\$ 39,50	ks	<input type="checkbox"/>
Goosens: \LaTeX Companion	\$ 33,95	ks	<input type="checkbox"/>
Graham, Knuth, Patashnik: Concrete Mathematics	\$ 32,95	ks	<input type="checkbox"/>
Knuth: \TeX book, paper	\$ 33,95	ks	<input type="checkbox"/>
Knuth: \TeX book, hard	\$ 39,95	ks	<input type="checkbox"/>
Knuth: METAFONTbook, paper	\$ 26,50	ks	<input type="checkbox"/>
Knuth: METAFONTbook, hard	\$ 39,95	ks	<input type="checkbox"/>
Knuth: 3:16, Bible Text Illuminated	\$ 31,50	ks	<input type="checkbox"/>
Lamport: \LaTeX , 2nd edition	\$ 34,95	ks	<input type="checkbox"/>
Abrahams: \TeX for the Impatient	\$ 33,95	ks	<input type="checkbox"/>
Knuth: The CWEB System of Structured Documentation	\$ 28,95	ks	<input type="checkbox"/>
Knuth: The Stanford GraphBase	\$ 52,50	ks	<input type="checkbox"/>
Knuth: Computer Modern Typefaces	\$ 53,95	ks	<input type="checkbox"/>
Kopka: A Guide to \LaTeX	\$ 41,50	ks	<input type="checkbox"/>
Snow: \TeX for the Beginner	\$ 33,95	ks	<input type="checkbox"/>
Adobe: Portable Document Format	\$ 31,50	ks	<input type="checkbox"/>
Adobe: PS Language Tutorial and Cookbook	\$ 24,95	ks	<input type="checkbox"/>
Adobe: PS Language Program Design	\$ 31,50	ks	<input type="checkbox"/>
Adobe: PS Language Reference Manual	\$ 39,95	ks	<input type="checkbox"/>
Adobe: Programming the Display PS System with NeXT-Step	\$ 33,95	ks	<input type="checkbox"/>
Adobe: Programming the Display PS System with X	\$ 37,50	ks	<input type="checkbox"/>
CDROM 4all \TeX	Kč 650,-	ks	<input type="checkbox"/>

Jméno (firma):

Adresa:

.....

.....

Telefon:

.....

podpis, razítko

Vážení T_EXisté,
leader L^AT_EX 2_ε a L^AT_EX3 teamu

Frank Mittelbach

navštíví ve dnech 15.–18. 6. na pozvání ζ TUGu Českou republiku a 17. 6. v odpoledních hodinách (patrně v Brně) prosloví přednášky (v angličtině) na níže specifikovaná témata. Přesné místo a hodinu konání jednotlivých akcí ještě upřesníme buď v příštím Zpravodaji, anebo písemným oznámením. Součástí patrně bude i valná hromada členů Československého sdružení uživatelů T_EXu.

- **‘Round-table discussion’**

do této diskuse můžete předem klást písemné dotazy na adresu Petra Sojky, ÚVT Masarykovy univerzity, Burešova 20, 601 77 Brno

- **L^AT_EX 2_ε’s view of encoding interfaces**

This talk will describe the models used by L^AT_EX 2_ε to translate input characters in a source document via internal representations to glyphs in a font. This is important for everybody writing in a language other than English and/or using a T_EX system that allows 8-bit input.

The talk will cover

- output encodings (font encodings) introduced with NFSS
- input encodings (keyboard encodings) introduced with L^AT_EX 2_ε 1994/12/01
- standard internal representations

Related issues discussed will be a proposal for 8-bit math encoding developed by the L^AT_EX3 Project and a general mechanism for providing “short references” to ease coding tasks such as “a → ä as they are already available in certain language files. The latter will probably be added to L^AT_EX 2_ε in the June release 1995.

- **Lessons learned from the Companion**

This talk will discuss problems in book production exemplified by the life-cycle of the The L^AT_EX-Companion. The author will discuss all steps of the production cycle and outline the various problems that can occur therein.

- **High quality typesetting issues: Shortcomings in T_EX**

Output produced by T_EX has higher standards than output generated automatically by most other typesetting systems. Therefore, this

talk will focus on the quality standards set by typographers for handtypeset documents and ask to what extent they are achieved by \TeX . Limitations of \TeX 's algorithms are analyzed; and missing features as well as new concepts are outlined.

Z obsahu příštího čísla

Luděk Berec: \LaTeX a hieroglyfické písmo starověkého Egypta

Stanislav Brabec: Lunisolární výpočty v \TeX u

Vydalo: Československé sdružení uživatelů \TeX u
vlastním nákladem jako interní publikaci
Obálka: Bohumil Bednář
Počet výtisků: 650
Uzávěrka: 5. dubna 1995
Tisk a distribuce: KOPP, Máchova 16, 370 01 České Budějovice
Adresa: \zTUG , MÚ UK, Sokolovská 83, 186 00 Praha 8
Podávání novinových zásilek povoleno Oblastní správou pošt
v Českých Budějovicích, j.zn. P 3.202/94 ze dne 19. července 1994

D E S A T E R O

nejhouževnatějších
pravopisných
P O V Ě R

1. **těhle** (*s pikantní mutací těchle*)
2. **ranné** brambory
3. **a nebo, na** levo
a naproti tomu
být **sto**, nashledanou
4. **standartní** (*z výchozího
standardu je standarta*)
5. **Himaláje** (*hattrick v jednom slově*)
6. **Vy, Váš** jinde než v dopisech
7. **napjetí**
a naproti tomu
idei, ideí, *ba i* idee
8. **dvěstěpadesátčtyři**,
5 ti, 5-ti, 5ti denní,
dokonce 5-tého
9. **plzeňsko, Plzeňský**,
středoevropan, **Střední** Evropa
10. **pedikúra, úsilovný, Úprka**
(poslední proniklo až do slovníků)

*A jako tři mušketýři byli čtyři, i našich po-
věr je o jednu víc:*

vyjímka