TeX kontra Word

Na diskusním e-listu `mensa@cs.felk.cvut.cz` se nedávno objevila určitá výměna názorů na téma, zda je lepší Word nebo TeX. Správně se tam došlo k závěru, že otázka je špatně položena a že se těžko dá srovnávat, který softwarový prostředek je lepší. Každý má něco do sebe. Ani já v tomto příspěvku tuto věc nerozsoudím, pouze se pokusím seznámit čtenáře s problematikou tohoto dilematu trochu podrobněji. Jako uživatel TeXu a hlavně jako člověk, který stojí v čele vývojového týmu českého TeXu, znám tento program do posledního detailu. Na druhé straně musím přiznat, že jsem Word ještě nikdy nepoužil. Tato pozice mě sice staví do situace, kdy těžko mohu o problému hovořit nestranně, přesto ale aspoň z jednoho pohledu hodně zasvěceně.

Chtěl bych předeslat, že ačkoli zde mluvím jmenovitě o Wordu, lze na jeho místo dosadit skoro libovolný komerční interaktivní softwarový produkt určený k formátování textu (Ami Pro, Pagemaker, Quark apod.). Slovo Word zde používám jen proto, že o něm začala na zmíněném e-listu diskuse a tento program asi zná nejvíce uživatelů.

Hledání standardu

Diskuse byla vyvolána po zaslání jistého příspěvku do tohoto listu ve Wordu a otevřenou otázkou, zda formát tohoto programu je vhodným standardem pro zasílání informací do Internetu e-poštou. Touto otázkou se má smysl zabývat až tehdy, kdy nám nevystačí k vyjádření myšlenek v e-poště prostý text a chceme naše sdělení zdůraznit některými formátovacími prvky, například volbou písma pro určité úseky textu. Nebo prostě chceme zaslat již formátovaný dokument. Zde se může stát formát Wordu standardem ve společenství lidí, kteří používají jediný operační systém MS Windows jediné konkrétní verze a všichni na něm mají MS Word jediné konkrétní verze. Toto společenství lidí se ovšem diametrálně liší od všech lidí, kteří používají Internet a dívají se například do diskusních skupin (jakými je třeba mensa) prostřednictvím net news. Jen část těchto lidí používá operační systém MS Windows konkrétní verze, který jediný je živnou půdou pro MS Word konkrétní verze. Já osobně pracuji především s UNIXovými grafickými terminály a podle složení Internetové komunity je zcela zřejmé, že podobných uživatelů je poměrně dost.

Zabývejme se chvíli otázkou, zda je formát MS Wordu přenositelný do jiného operačního systému. Představme si, že by mohl existovat prohlížeč tohoto formátu třeba pro UNIXy. Neexistuje. Proč? Protože formát Wordu obsahuje v sobě pouze odkazy na fonty, které jsou instalovány v uživatelově operačním systému, přičemž algoritmy pro vykreslení znaků z těchto fontů chybějí. Takový dokument je zpracovatelný pak jen na takovém systému, kde jsou všechny použité fonty instalované. Bohužel, v prostředí MS Windows se rozšířil formát fontů TrueType, který nemá v UNIXovém světě obdoby. Možná, že návrháři ostatních operačních systémů pochopili, že formát firmy Adobe Type1 skrývá daleko větší možnosti a odmítli se formátem TrueType zabývat. Možná se za tím skrývá i nějaká obchodní politika. Důsledkem toho ale je, že sady fontů, užívaných v systému MS Windows, nemají v jiných operačních systémech obdobu. Navíc záleží i na kódování fontů. České dokumenty Wordu jsou výhradně kódovány v CP1250, což je kódování, které nepřekračuje hranice jednoho operačního sytému. Mimo tento systém je tedy zmíněný dokument obtížně dešifrovatelný.

Někdo z diskutujících uvedl, že nová verze Wordu umí do dokumentu přibalit použité fonty. To je stejně na nic, pokud formát těchto fontů není operačním systémem příjemce podporován. Navíc se tato vlastnost týká až nejnovější verze a může tedy v současné době být zdrojem nekompatibility i mezi uživateli Wordu různých verzí.

Ukázal jsem, že zaslat příspěvek ve Wordu do Internetu neurčité skupině lidí je stejný masochismus, jako poslat binární `dvi` soubor, který je výstupem TeXu. Stejný v tom smyslu, že dvi soubor má rovněž pouze odkazy na použité fonty a předpokládá se, že příjemce má TeXovou instalaci v níž tyto fonty budou. Ovšem on je to masochismus ještě větší. Onu instalaci TeXu totiž může příjemce mít na *libovolném* operačním systému. V případě, že se použije standardní sada fontů CM (která je implicitně součástí každé TeXové instalace), pak nejsou s tím `dvi` na celém světě žádné problémy a český text je zobrazitelný a tisknutelný třeba v Sydney. Žádnou českou lokalizaci příjemce nemusí mít. Tato možnost je postavena na schopnosti TeXové instalace na úrovni postprocesingu v případě nutnosti automaticky sestavovat složené znaky z jednotlivých elementů (například háček a písmeno).

Přesto uživatelé TeXu takové věci, jako zasílání `dvi` do e-listů, nedělají. Proč? Vědí, že TeX není jediný možný prostředek, který mají určitě všichni. Jak to, že se tak často setkáváme s Wordovými dokumenty? Zřejmě uživatelé Wordu nabyli dojmu, že to je prostředek, který musí mít každý. Kladou nesprávné rovnítko: počítač = PC = MS Windows a tím dokazují svoji omezenost. Nebo netoleranci. Zde tedy spatřuji i rozdíl ve způsobu myšlení těchto dvou skupin uživatelů. Proč? Protože technické podrobnosti jsou před uživatelem Wordu dobře skryty a to vytváří iluzi, že všechno je jednoduché. Na druhé straně uživatel TeXu většinou určitou technickou představu o věci má.

Pokud se uživatel TeXu rozhodne poslat něco TeXového do e-listu, je to text, který stojí na vstupní straně TeXu a který je čitelný běžnými editory textových souborů. Pravda, občas se tam vyskytne nějaká značka, která může být pro cizince tajemná. Ovšem v případě hladkého textu (což je většina textů třeba v tomto časopise) autor potřebuje nanejvýš vyjádřit, že je nějaký text v `\uv{českých uvozovkách}` a něco zdůrazní `{\it kurzívou}`. Více značek, než právě zde předvedené, pro hladký text nepotřebuje. Odstavce odděluje prázdným řádkem, na což je už zvyklý z e-pošty. Takový text může číst určitě každý, i ten, kdo si jej nemá zrovna možnost zpracovat TeXem.

Ani takový formát není ovšem žádným standardem, a proto se vraťme k původní otázce, jak vypadají standardní formáty prezentace textů na Internetu. Uživatelé přistupují k nestandardním formátům hlavně proto, že chtějí zachovat „háčky a čárky“. Pokud komunikují se skupinou lidí, kde všichni používají program na poštu se schopností zpracovat dokumenty podle standardu MIME, pak by akcenty neměly být problémem. Věc pak není závislá na operačním systému. Pro kódování se používá ISO 8859-2 rozepsané pomocí hexa kódů do „ASCII 7 bit“ (nutné pro bezpečný transport starými poštovními servery). Jde tedy jen o správnou konfiguraci programu, který manipuluje s poštou na straně uživatele.

Pro zaslání informace do diskusního listu větší skupině lidí je bohužel třeba smířit se s tím, že nepouzivame hacky a carky. Ačkoli se může zdát, že na nás dýchá počítačový pravěk, podstatně důležitější je, aby zprávu mohli číst *všichni*. Třeba i ti uživatelé, kteří momentálně jsou na cestách v zahraničí a pro připojení do Internetu jsou vděčni i za sedmibitový textový terminál.

Chceme-li prezentovat už formátovaný text, pak doporučuji použít PostScript nebo PDF (viz níže). Chceme-li prezentovat text s formátovacími informacemi, ale k možnému doplnění příjemcem (například různé dotazníky), pak doporučuji použít formát html. Ony dotazníky (různé grantové přihlášky apod.) by se daly udělat na úrovni WWW serveru a CGI scriptů, pokud pořádající organizace má šikovného web mastera. Přitom Netscape či jiné prohlížeče html nám běhají na všech dostupných operačních systémech.

PostScript je formát pro grafický popis strany, který je nezávislý na použitém operačním systému a výstupním zařízení. Dražší výstupní zařízení mají PostScriptový RIP (raster image procesor) zabudovaný jako svou součást. Pokud na to nemáme, můžeme použít volně dostupný (public domain) program Ghostsript, který implementuje RIP do počítače a je instalovatelný na libovolném operačním systému. Použitím Ghostscriptu si můžeme též PostScriptový soubor prohlížet v grafickém okénku na obrazovce počítače. PostScript je de facto standard pro profesionální zpracování sazby a popisuje úplný vzhled strany. Byl vyvinut firmou Adobe v roce 1985. Použité fonty jsou v PostScriptovém souboru zahrnuty. Příjemce si může PostScriptový dokument prohlédnout například Ghostscriptem na libovolném operačním systému. Kódování dokumentu je shodné s kódováním přibalených fontů a nemusí mít nic společného s kódováním fontů systému.

PostScriptový soubor obsahuje instrukce zapsané jen sedmibitovými znaky, takže jej můžeme přidat k dopisu bez dalšího komplikovaného „balení“. Poštovní servery nám dokument nezničí. Protože je ale takový soubor poměrně rozsáhlý a původní text je v něm velmi špatně čitelný, není vhodné zasílat jej do diskusních skupin. Vyspělé chování je poznat podle toho, že přispěvatel napíše, že informace zhruba pojednává o tom či onom a zájemce ji může v podrobnější podobě získat na URL `ftp://muj.server.cz/pub/tam/info.ps`.

PDF (portable document format) je rovněž formát vyvinutý firmou Adobe. Doplňuje (zhruba řečeno) PostScript o hypertextové možnosti. Firma Adobe nabízí zdarma prohlížeč tohoto formátu Acroreader, který je instalovatelný v libovolném operačním sytému.

Uvedené formáty jsou skutečně definovány jako standard, tj. s cílem udržet tento formát beze změn i v budoucnosti. Jsou to formáty, ve kterých lze archivovat informace pro příští generace. Velmi výjimečně se provádí změny formátu, ovšem vždy zachovávající kompatibilitu původních dokumentů. Například poslední a zatím jediné rozšíření PostScriptu (směrem k barvám) bylo provedeno v roce 1991 (tzv. level 2) a tato verze PostScriptu samozřejmě umí zpracovat původní level 1 (dokumenty z roku 1985). Na druhé straně ze zkušenosti víme, že formáty navržené pro provoz konkrétní verze konkrétního editoru jsou časově velmi pomíjivá záležitost.

Když se podíváte například na moji www stránku, zjistíte, že vystavuji své články a knihy vesměs ve třech formátech: TEX, PostScript a PDF. Ani v jednom případě tím nezvýhodňuji uživatele jednoho operačního systému proti jiným. Ani v jednom případě tím nenutím uživatele kupovat k prohlédnutí dokumentu komerční softwarový produkt.

Domnívám se, že ačkoli se nestal vstupní formát TEXu rozšířeným standardem, je to asi jediný možný formát, ve kterém lze archivovat dokumenty pro použití za deset a více let. Pod pojmem „použití" nyní nemyslím jen jejich prohlédnutí a vytištění (na to by stačil i PostScript), ale i případnou změnu formátování a úpravy textu, pokud budeme někdy v budoucnu chtít. Stačí se podívat do minulosti. TEX pracuje již od roku 1978 a dokumenty vytvořené v té době lze dnes zpracovat zcela stejně. Takovou devatenáctiletou konstantu bychom ve světě počítačů asi těžko hledali. Navíc autor TEXu tento projekt v roce 1989 zcela zmrazil a dal tím záruky, že se navěky bude TEX chovat stejně, jako dnes. Nemám tedy obavu o své knihy a články, pokud se někdy na stará kolena rozhodnu provést jejich reedici.

Dostupnost softwaru

TEX je program vytvořený v rámci státního projektu na Stanfordově univerzitě a je k dispozici zdarma pro obecné použití. Autor Donald Knuth jej předal veřejnosti ve formě více než megového souboru `tex.web`, který obsahuje všechny algoritmy TEXu současně s jejich podrobnou dokumentací. Také dal k dispozici nástroje na zpracování tohoto souboru v libovolném operačním systému a některé další podpůrné programy. Výsledkem zpracování souboru `tex.web` je spustitelný program TEX pro konkrétní operační systém. Mimoto lze informace z `tex.web` zpracovat TEXem a získat tak pětisetstránkovou knihu obsahující všechny algoritmy TEXu vyjádřené v metajazyku WEB a navíc velmi důkladně dokumentované.

K dispozici jsou i manuály k TEXu z Knuthovy pětidílné monografie Computers & Typesetting. Nejpodstatnější je zřejmě první díl, který Knuth nazval The TEXbook a který je zcela dostačující i pro uživatele, který nechce číst zdrojový text TEXu. Mnohým uživatelům i tato kniha přetéká přes hlavu, a proto se vyrojilo plno dalších publikací různých autorů typu „začínáme s…".

Implementaci `tex.web` pro daný operační systém dělají buď komerční firmy (které z hlediska poměrně nízké poptávky a vysoké konkurence z univerzit z toho moc neprofitují), nebo nadšenci na univerzitách. Ti druzí po vzoru svého vzoru (Knutha) dávají své výsledky lidem zdarma. Mezi nejvíce rozšířené implementace patří Berryho web2c pro všechny rozmanité UNIXy i jiné operační systémy, které mají rozumný překladač jazyka C. Dále uveďme Mattesův emTEX pro OS/2 a DOS. Mattes dodává TEX už ve spustitelné podobě `*.exe` společně se sadou dalších nástrojů pro TEXování (prohlížeče `dvi`, programy pro tisk apod.). Všechny tyto věci jsou součástí tzv. CTAN (Comprehensive TEX Archive Network), což je systém ftp zrcadel po celém světě, který nabízí veřejný TEXový software. Ten v dnešní době dosahuje sumárně tři gigabajty informací.

Po celém světě existuje plno diskusních e-listů zaměřených k TEXu. Tam mohou uživatelé hledat radu. Na těchto e-listech jsou připojeni odborníci, kteří TEX implementují do jednotlivých výpočetních prostředí, takže se uživatelům dostává rady skutečně fundované.

Dostupnost komerčního software a důkladných manuálů k těmto produktům, ze kterých by se člověk dozvěděl *jak* to funguje a ne jen *co* se má dělat, nebudu raději komentovat a srovnání si udělá každý sám. Jistě se též mnozí čtenáři setkali se servisem, kdy zprostředkující firma vyšle k uživateli rádoby „odborníka", za kterého účtuje nekřesťanské peníze a který mnohdy ví míň, než jen trochu poučenější uživatel.

Dávkový kontra interaktivní způsob manipulace se sazbou

TEX samotný je program, na jehož vstupní straně je soubor připravený v určité notaci nejčastěji textovým editorem a na výstupní straně je úplná informace o vzhledu sazby nezávislá na výstupním zařízení (tzv. `dvi`). Provoz TEXu je tedy postaven na dávkovém způsobu zpracování. Připravíme vstupní soubor, zpracujeme TEXem a prohlédneme výslednou sazbu. Pak upravíme vstupní soubor a vše se opakuje.

Způsob, s jakým komfortem tuto činnost provozujeme, záleží jednak na výkonnosti hardware, dále na schopnostech použitého operačního systému a v neposlední řadě na vlastní implementaci TEXu. Není vyloučeno, že uživatel bude pracovat s aplikací, ve které může interaktivně realizovat svá přání o vzhledu sazby pomocí myši, přičemž tato aplikace bude mít tlačítko s označením řekněme „věrný náhled". Po použití tohoto tlačítka uloží aplikace dokument ve tvaru vstupního souboru pro TEX, spustí TEX a nakonec aktualizuje prohlížeč `dvi`. Uživatel uvidí ve vedlejším okénku totéž, ovšem rozmístění sazby je nyní naprosto přesně takové, jak to bude vypadat na tiskárně. Uživatel pak může pokračovat dál v práci ve své interaktivní aplikaci. Ve veřejně dostupných implementacích TEXu se taková aplikace nedodává, nicméně například LyX pro UNIXy je volně dostupným experimentem tohoto druhu. Zatím bohužel ne zcela dotažen. Uživatelé TEXu takové interaktivní nástroje většinou nepoužívají a ani po nich moc nevolají. Raději pracují s „holým" vstupním textem, protože si nenechají ujít možnost sahat přímo na páky, které ovládají sazbu. Na první pohled zajímavý uživatelský polštář jim jenom brání ve vyjadřovacích možnostech.

Otázka, zda zvolit interaktivní nebo dávkový způsob přípravy sazby, vyplývá často z celkového přístupu uživatele k počítači. Tomu uživateli, který se neštítí slova algoritmus a který někdy v životě programoval, může dávkový přístup ovládání sazby připadat přirozený. Ostatním uživatelům, k jejichž rukám se počítače po heslem „počítač není nic složitějšího než rádio nebo automatická pračka" dostaly do rukou v posledních letech, je samozřejmě dávkový způsob uvažování zcela nepřirozený. Připadá jim, že počítač musí na jejich úkony reagovat okamžitě jako to rádio. I na chyby očekávají okamžitou reakci, například houknutí. Jiná je situace, když zapisujeme nějaké informace pro program do souboru. Ono zpoždění, než se tento program spustí a vyjdou najevo naše chyby, může být pro mnohé psychicky neúnosné. To je celkem pochopitelné a nechci zde proto zpochybňovat výhody interaktivního přístupu.

Nikam nevedly diskuse, zda je účelnější dávkový nebo interaktivní přístup k sazbě, který se na e-listu mensy rozváděl do otázek, při kterém režimu použijeme méně úhozů, při kterém režimu je rychleji k dispozici množina informací, která nás vede k dalším krokům, jak snadno je tato množina informací zapamatovatelná, při kterém režimu může pracovat laik apod. Obě strany zjistily, že počet úhozů lze zkracovat klávesovými zkratkami (v dobrém textovém editoru vstupního souboru pro TEX velmi pružně konfigurovatelnými) a že nabídky dalších možností v on-line „menu" je jedna věc, knihy roztahané po stole vedle počítače druhá a tytéž knihy převedené do elektronické a hypertextové podoby věc třetí. Tak, jako někdo má dobrou paměť „na slovíčka" a lépe se mu pracuje s knihou a programovacím jazykem, jiný může mít paměť „na obrázky" a lépe se mu orientuje v obrázkových nabídkách interaktivních programů. Tyto dvě skupiny lidí by k vůli tomu neměli mezi sebou vyhlašovat válku.

Pokud laika naučíme používat výše zmíněné dvě značky (`\uv` a `\it`) a necháme ho připravit text pro hladkou sazbu, pak text dokáže odevzdat v pořádku. K dalšímu formátování ovšem musí přistoupit odborník. Za výhodu bych považoval, že je možno při vhodné dělbě práce oprostit autora textu od starostí nad *formou* a může se plně věnovat jen *obsahu* svého díla. Problémy s formou komplikují mnoha autorům práci a jsou přitom často zhola zbytečné, protože autorův text nakonec bývá stejně přeformátován v jiném prostředku, než je editor, ve kterém autor své dílo připravoval.

What You See Is What You Get

V souvislosti s interaktivní úpravou sazby se často používá zkratka WYSIWYG. Naopak o programech, které zpracovávají sazbu dávkovým způsobem se tvrdí, že nejsou WYSIWYG. Vzhledem k tomu, že tato zkratka znamená „co vidíš to dostaneš", je tato terminologie velmi jemně řečeno nepřesná. V TEXové instalaci si mohu prohlédnout výsledné `dvi` na obrazovce a vidím přesně to, co dostanu na tiskárně. Náhled je v případě sazby naprosto věrný a k dalším „překvapením" nemůže z technologických důvodů dojít. Na druhé straně pracovní plocha interaktivních programů sice nabízí pružně opracovatelný náhled, ale mám reference od mnoha uživatelů, že se často po vytištění nestačí divit, co vlastně před tím viděli na obrazovce.

Zřejmě se tedy tou zkratkou míní, že nelze přímo v náhledu dělat opravy a změny. Ano, v TEXové instalaci pracujeme ve vedlejším okénku, kde třeba programujeme sazbu. Pokud máme dosti rychlý počítač, pak po zpracování našeho kódu na pozadí TEXem vidíme v náhledu provedené změny prakticky okamžitě. Takže vlastně děláme v náhledu změny, jen ne přímo zásahem do okénka tohoto náhledu.

Pojem WYSIWYG se tedy asi používá v souvislosti s tím, jak připravuje text pro sazbu autor. Ovšem před chvíli jsme zmínili, že autora může starost o formu jenom zdržovat a že nakonec bude jeho dílo zpracováno v jiném prostředku, než sám používá. Takže to, co vidí, zdaleka není to, co dostane po typografickém a knižním zpracování.

Nedoporučuji tedy používat pojem WYSIWYG, protože v něm mnohé interaktivní programy na celé čáře prohrávají. Raději se držme označení interaktivní a dávkový způsob zpracování sazby.

Uživatelská přítulnost

Program TeX samotný neřeší problém uživatelské přítulnosti vůbec. V určitém smyslu se s problémem potýkají konkrétní TeXové implementace, které mohou vycházet z výhod použitého operačního systému (grafické rozhraní, členění nápověd, multitasking apod.). Proto je třeba v této věci důsledně rozlišovat mezi pojmy TeX a konkrétní TeXová implementace.

Na druhé straně programy s interaktivním způsobem zpracování mají uživatelskou přítulnost ve svém popisu práce. S jakým úspěchem se s tím vyrovnají záleží na tom kterém z nich.

Možnosti a flexibilita

Program TeX se věnuje jedinému úkolu, který by se dal z pohledu interaktivních prostředků charakterizovat jako „nalévání textu do sazby". Tuto činnost vykonává dobře a nemá v ní daleko široko konkurenci. Byl navržen tak, aby výsledná kvalita sazby byla co nejlepší. Například algoritmus řádkového zlomu, který hledáním minima jisté cenové funkce řeší co nejlepší rozvržení odstavce jako celku, nemá v mnoha konkurenčních programech obdobu. Mezi další přednosti bych uvedl automatickou tvorbu ligatur a kerning podle informací ve fontu a zcela nepřekonané algoritmy na matematickou sazbu.

Interaktivní systémy většinou mají svůj cíl stanoven šířeji, takže problém „nalévání textu do sazby" je jen dílčí problém. Podle toho taky často vypadá kvalita výsledku. Tyto systémy totiž nemohou do uvedeného dílčího problému investovat příliš mnoho sofistikovaných algoritmů, protože by se ztratila výhoda interaktivního ovládání. Uživatel by musel na každou změnu v textu čekat neúměrně dlouho.

Flexibilita TeXu vyplývá mimo jiné z důkladné dokumentace jeho datových struktur a z přítomnosti na systému nezávislých pomocných nástrojů, kterými lze do těchto struktur zasahovat. Například změna metrických informací fontů nebo možnost použití tzv. virtuálních fontů na úrovni `dvi` otevírá netušené možnosti. Kdo tyto věci zná, je naprostým pánem nad výsledkem sazby.

Naproti tomu mnohé interaktivní programy se chovají jako „černá skříňka". Datové struktury nejsou z komerčních důvodů dokumentovány a zásah do nich a možná změna jiným způsobem, než nabízí uživatelský manuál, je vyloučena.

TeX má také velmi mocný programovací jazyk, kterým lze řídit a automatizovat ono „nalévání textu" v podstatě podle jakýchkoli požadavků. Tímto jazykem se definuje například vzhled a automatická tvorba obsahů, rejstříků a odkazů všeho druhu. Rovněž se pomocí tohoto jazyka definuje dialekt vstupních souborů TeXu. Jedná se o způsob značení a členění vstupního textu. Jedním z takových dialektů je například LaTeX. Flexibilita programovacího jazyka TeXu je značná.

Zde bohužel nemohu srovnávat a posoudit zcela věrohodně, jak vypadají programovací jazyky interaktivních nástrojů. Poznamenejme, že i Word má svůj programovací jazyk a že se v něm dají dělat i viry.

Petr Olšák `http://math.feld.cvut.cz/olsak`