By preparing a manuscript in TEX format, you will be telling a computer exactly how the manuscript is to be transformed into pages whose typographic quality is comparable to that of the worlds finest printers; yet you won't need to do much more work than would be involved if you were simply typing the manuscript on a ordinary typewriter.

Donald Ervin Knuth: *The TEXbook.*

Co je TEX ?

Kapitola první: Seznamujeme se s TEXem.

Petr Olšák

Program, o kterém si troufám tvrdit, že zabere nejvíce času jak CPU tak i každému uživateli počítače, je ve většině případech editor. Editorem připravujeme texty nejen pro kompilátory a jiné programy (textové soubory pro počítač), ale také pro lidi. Ve druhém případě chceme, aby výsledný text byl bezvadný nejen po obsahové stránce, ale aby měl též co nejlepší vzhled. S rostoucí kvalitou výstupních zařízení pro tisk rostou i naše požadavky na kvalitu tištěného dokumentu. Velice záhy každý zjistí, že zde nevystačí s obyčejným editorem (určeným pro pořizování textových vstupů pro počítač) a většinou se poohlédne po některém textovém procesoru. Náročnější uživatelé sáhnou hlouběji do kapsy a pořídí si DTP program. Ti nejnáročnější uživatelé nemusí vůbec sahat do kapsy a pořídí si TEX, program, který v této věci splní v podstatě všechny požadavky a přitom je zadarmo *).

Možná někoho odradí právě ta skutečnost, že je to program šířený jako „public domain software“. Může se zdát, že je to proto, že produkt není příliš dobrý nebo není zcela domyšlen. Opak je pravdou! Autorem TEXu je profesor Donald Knuth ze Stanfordské university. Ti, kteří znají např. některý z dílů jeho učebnice *The Art of Computer Programming*, určitě mi dají za pravdu, že je to autor v oblasti počítačů a programování nesmírně erudovaný. Pokud jste si přečetli článek o TEXu od Oldřicha Ulricha a Petra Nováka, který vyšel v Softwarových novinách dříve, znáte navíc důvod, proč je program zadarmo. Stručně řečeno jde o to, že se program narodil na akademické půdě a ne v nějakých softwarových továrnách, které na světě existují proto, aby si vydělaly prodejem svých produktů. Budu-li zde dělat reklamu TEXu, nikdo mi to proto nemůže mít za zlé. Dělám tím reklamu pouze jeho autorovi, kterého velmi obdivuji.

Donald Knuth shrnul svou práci o matematické typografii do pětidílné monografie *Computers & Typesetting*. O jednotlivých dílech se podrobně zmíníme později; nyní citujme jen první díl:

Volume A: The TEXbook. Addison Wesley, Reading, Massachutssets, 1986.

Tento díl, je pro uživatele, kteří chtějí co nejdříve začít řešit jednoduché úlohy v TEXu, pravděpodobně příliš obsáhlý a složitý. Přiznám se, že jsem zpočátku strávil asi tři měsíce intenzivního studia nad touto 480 stránkovou knihou a ještě dnes v ní stále nalézám něco nového. Pro uživatele TEXu, kteří se nechtějí stát odborníky, stačí něco menšího. Velice pěkná je například knížka:

M. Doob. *Jemný úvod do TEXu.* Skripta MFF UK, Vydavatelství Karolinum, Praha 1990.

Je zajímavé, že program TEX není mezi uživateli počítačů PC příliš rozšířen. Používá se většinou v akademických kruzích, zvláště tam, kde je potřeba vysázet poměrně složité matematické formule. Přitom si myslím, že by našel uplatnění i jinde. Proto jsem se rozhodl přispět do Softwarových novin tímto článkem. Mohl by pomoci mnohým v rozhodování o tom, jak a jakým programem zlepšit úroveň psaných textů na počítači. Text tohoto článku jsem zaslal redakci Softwarových novin nikoli na disketě, ale v podobě hotových stránek sázených TEXem a vytištěných na laserové tiskárně s rozlišením 300 dpi.

V tomto „seriálu“ se pokusíme srovnat možnosti TEXu s komerčně šířenými DTP programy, zběžně projdeme vlastnosti nejčastěji používaných formátů TEXu (tj. PlainTEX, LATEX, AMS-TEX a pod.), popíšeme „public domain“ instalaci tohoto programu na počítačích PC běžící pod DOSem nebo OS/2 (emTEX) a zmíníme se o mnoha dalších zajímavostech souvisejících s tímto programem. Především se zaměříme na ty informace, které se při počáteční práci s programem v literatuře těžko hledají.

* Slovo TEX se čte řecky „tech“ a nikoli „teks“

Na své si přijdou nejen čtenáři, kteří se chtějí něco dozvědět o základech počítačové typografie, ale i programátoři. TEX je totiž vybaven jistým „programovacím jazykem“ (jazykem maker). Tento programovací jazyk je navržen hlavně pro řešení úloh související s návrhem formátu dokumentu (například automatické číslování kapitol, generování obsahů a rejstříků, podpora křížových referencí v dokumentu apod.). Skutečnost, že někdo naprogramoval v TEXu interpret BASICu pouze dokumentuje možnosti tohoto programovacího jazyka, nicméně pro typografii nám to není nic platné.

V této kapitole se pokusíme přiblížit program TEX takovému čtenáři, který o něm zatím nic neslyšel anebo má o programu jen vzdálenou představu.

Představme si, že máme přítele (jmenuje se například Ferda), který už má s TEXem jisté zkušenosti a pozve nás k počítači aby nám ukázal, jak se s tímto programem pracuje. Stojíme (nebo sedíme) u jeho PC a díváme se mu přes rameno, co kutí. Ferda nám předvede, jak připravit dopis v PlainTEXu.

V počítači je připravena dávka, která startuje systém TEX a která se například jmenuje `tex.bat` . O další podrobnosti se zatím nebudeme zajímat. Poté, co Ferda napíše v DOSu příkaz `tex dopis.tex` , na obrazovce se objeví „menu“, které nabízí několik možností, co budeme dělat se souborem `dopis.tex` .

První položka, která nás v „menu“ bude zajímat, má název `Edit`. Ferda zvolí právě tuto položku, což způsobí vyvolání nějakého ASCII editoru pro soubor `dopis.tex` . Zde záleží na Ferdovi, jaký editor si pořídí nebo jaký se mu líbí. Editor není přímou součástí TEXu a každý uživatel si nakonfiguruje takový editor, který mu nejvíce vyhovuje.

V editoru napíše Ferda následující text:

```
Vážení telefonní ústředníci!

Dovolte mi, abych Vás upozornil, že pod číslem jednacím 17063/91 mám
u Vás "uloženu" žádost o telefon. Přitom více než rok se nic neděje.
Věřím, že zdárně pokračujete v budování nových linek a že přijde den,
kdy se my čekatelé dočkáme.

Chtěl bych v této souvislosti upozornit na to, že v normálně fungujících
státech je běžné, že je žádosti vyhověno do druhého dne a to nejen
v hlavním městě státu.

Příjemnou zábavu při čtení urgencí jednotlivých žádostí o telefon přeje

                                                 Ferdinand Mravenec
\bye
```

Poté Ferda ukončí práci v editoru a soubor `dopis.tex` uloží na disk. „Menu“ nyní nabízí další položku s názvem `TeX`. Vybráním této položky se rozeběhne vlastní program TEX, jehož vstupním souborem je `dopis.tex` . Program vypíše na obrazovku hlášení o své činnosti, kterému zpočátku nemusíme rozumět. Zajímavá je pro nás jen informace v hranatých závorkách: `[1]`. Ta říká, že TEX vytvořil jednu stránku dokumentu s pořadovým číslem 1. Máme-li rozsáhlejší dokument, většinou při zpracování TEXem pozorujeme, jak přibývají „zalomené“ stránky právě podle hranatých závorek s čísly.

TEX „vysází“ text do odstavců a stránek podle údajů, které přečte ze vstupního souboru, a vytvoří výstupní soubor `dopis.dvi`, v němž jsou uloženy všechny metrické informace o umístění jednotlivých písmen na stránkách. Tento výsledek si nyní můžeme prohlédnout dalším programem, který se vyvolá z „menu“ položkou `View`. Tento program nám na obrazovce zobrazí výsledek práce TEXu v čitelné podobě tak, jak to bude vypadat na tiskárně:

Vážení telefonní ústředníci!

Dovolte mi, abych Vás upozornil, že pod číslem jednacím 17063/91 mám u Vás “uloženu” žádost o telefon. Přitom více než rok se nic neděje. Věřím, že zdárně pokračujete v budování nových linek a že přijde den, kdy se my čekatelé dočkáme.

Chtěl bych v této souvislosti upozornit na to, že v normálně fungujících státech je běžné, že je žádosti vyhověno do druhého dne a to nejen v hlavním městě státu.

Příjemnou zábavu při čtení urgencí jednotlivých žádostí o telefon přeje

Ferdinand Mravenec

Vidíme, že TEX rozdělil vstupní text na odstavce, přičemž oddělovačem mezi odstavci je ve vstupním souboru prázdný řádek. Každý odstavec má zcela nově „zalomené“ řádky – konce řádků ve vstupním souboru vůbec neodpovídají konečnému výsledku. Kromě toho si všimneme, že ačkoli Ferda před svým jménem udělal větší množství mezer, TEX tuto skutečnost vůbec nerespektoval.

Při prohlížení výstupu na obrazovce Ferda prohlásil, že se mu to tak nelíbí. Chtěl by tomu dát nějakou lepší formu. Ve vstupním textu zatím uvedl pouze dva druhy speciálních formátovacích příkazů TEXu. Jedním z nich jsou prázdné řádky (příkaz pro ukončení odstavce) a druhým je příkaz `\bye`. Tímto příkazem se Ferda neloučí s telefonními ústředníky, ale s TeXem. Přesněji – TeX „zalomí“ poslední (zde jedinou) stránku dokumentu a ukončí svou činnost. Jak dále uvidíme, většinou se formátovací příkazy od běžného textu liší právě znakem ‘\’ před názvem příkazu.

Aby dopis vypadal lépe, přidá Ferda do vstupního textu souboru některé další příkazy. Ukončí proto prohlížení a vrátí se do „menu“. Tam znovu zavolá ASCII editor na doplnění vstupního souboru o další údaje. Postupně začne Ferda řešit před našimi zraky tyto formátovací požadavky:

1. Posunout podpis doprava. Řádek se svým jménem obohatí Ferda o příkaz `\rightline`, který zajistí, že text napsaný za ním ve složených závorkách bude vypsán na pravou zarážku. Dále před svým jménem použije přepínač písma `\it`, který způsobí, že bude podpis vysázen kursivou (`it` je zkratka pro *italic*). Řádek s podpisem je tedy ve vstupním souboru opraven takto:

```
\rightline{\it Ferdinand Mravenec}
```

2. Přidat datum dopisu. To provede Ferda například přidáním řádku

```
\rightline{V Praze dne \today}
```

Příkaz `\rightline` už známe (text bude vysázen na pravou zarážku), nový je příkaz `\today`, který zajistí vysázení textu s aktuálním datem. Zde ale pozor! Toto není standardní příkaz TeXu. Takové příkazy připravují pro uživatele lidi, kteří umí v TeXu „programovat“ a rozumí mu na úrovni knihy *The TeXbook*. Připravují pro ně různé tzv. „styly“, které nastavují TeX do jistých režimů (například jiný bude styl pro psaní dopisu a jiný při sestavování nějaké technické dokumentace). Další „styly“ definují užitečné příkazy. Tak zrovna příkaz `\today` je definován v souboru `czech.sty`, jehož autorem je Oldřich Ulrych. V okamžiku použití příkazu `\today` už musí TeX tento příkaz znát. Proto je třeba před jeho použitím načíst soubor `czech.sty` příkazem:

```
\input czech.sty
```

Programátoři takový příkaz určitě znají (většinou ho nazývají `include`). Příkaz způsobí, že TeX přesměruje svůj vstup na specifikovaný soubor a po jeho načtení pokračuje ve čtení dalších řádků hlavního vstupního souboru. Poznamenejme ještě, že tento soubor je užitečné volat na začátku každého českého dokumentu, protože mimo jiné inicializuje české dělení slov.

3. Oslovení napsat výrazně. Celé oslovení uzavře Ferda do složených závorek (ty TeX interpretuje jako začátek a konec jisté skupiny) a vloží na začátek skupiny přepínač písma `\bf`. Tento přepínač způsobí, že text ve skupině bude sázen polotučně (`bf` je zkratka pro **bold face**). Oslovení tedy po Ferdových opravách dopadlo takto:

```
{\bf Vážení telefonní ústředníci!}
```

4. Mezi oslovením a prvním odstavcem je vhodné vynechat trochu místa. Za oslovení vloží Ferda příkaz `\vskip 0.9cm`, který vytvoří vertikální mezeru požadovaných rozměrů.

5. Uvozovky. Slovo „uloženu“ má Ferda uzavřené do uvozovek. Problém je v tom, že TeX vysázel uvozovky anglické (“ ”) a my chceme uvozovky české („ “). Ovšem k tomu nám poslouží styl `czech.sty`. Zde je definován příkaz `\csprime`, který zařídí, že budou uvozovky sázeny (v celém dokumentu) česky. Stačí tedy použít tento příkaz někde na začátku dokumentu.

6. List nemá být dole opatřen číslem strany. Někde na začátek dokumentu se vloží příkaz `\nopagenumbers`, který potlačí číslování stránek.

7. Vložit vertikální mezeru před rozloučením. Analogicky, jako v případě mezery za oslovením vloží Ferda na příslušné místo příkaz `\vskip 0.6`. Všimněme si, že Ferda udělal chybu – zapomněl napsat jednotku, v níž je zapsán délkový parametr. Zatím mu to neprozrazujme a čekejme, co se stane dál.

Po opravách uloží Ferda vstupní soubor `dopis.tex` na disk a ukončí editaci. Z „menu“ vyvolá TeX. Ten začne číst soubor `dopis.tex` a objeví v něm syntaktickou chybu. Na obrazovce se vypíše následující chybové hlášení:

```
!Illegal unit of measure (pt inserted)
<to be read again>
                   P
<to be read again>
                   ř
l.19 Př
       íjemnou zábavu při čtení urgencí jednotlivých žádostí o tel...
?
```

TeX zde praví, že je zde zvolená nesprávná metrická jednotka s názvem `Př`. Jsou to totiž první dvě písmena za chybně zadaným příkazem `\vskip`. Navíc je patrné, že k nehodě došlo na řádku 19. Tento řádek je na obrazovce vypsán a v místě chyby „roztržen“ na dva. Otazník na obrazovce vyzývá Ferdu, aby něco učinil. Ten na to chvíli nechápavě kouká a záhy zvolá: „Aha! Zapomněl jsem napsat jednotku u příkazu `\vskip`.“ Tato situace Vám určitě silně připomíná ladění programu pod nějakým kompilátorem.

Pokud Ferda na otazník odpoví klávesou `Enter`, TEX si chybu opraví po svém (zde dosadí za neznámou jednotku tiskařský bod „`pt`“, který je asi třicetkrát menší, než centimetr. Tuto jednotku asi Ferda neměl na mysli a proto se výstup TEXu bude lišit od původního záměru autora. Na otazník může Ferda odpovědět také znakem `x`, kterým ukončí předčasně práci TEXu. Potom přejde Ferda do ASCII editoru a chybu opraví. V tuto chvíli je dobré upozornit na to, že existují editory, které zobrazí na obrazovce nejen soubor `dopis.tex`, ale v jiném okně též hlášení TEXu o chybách, které je uloženo v souboru `dopis.log` . Navíc tyto editory dokážou posunovat kurzor postupně na všechny výskyty chyb v dokumentu (je-li jich více), takže to vypadá podobně, jako ladění programu v prostředí Borland C.

Opravený soubor vypadá následovně:

```
\input czech.sty
\csprime
\nopagenumbers

\rightline{V Praze dne \today}

{\bf Vážení telefonní ústředníci!}

\vskip 0.9cm
Dovolte mi, abych Vás upozornil, že pod číslem jednacím 17063/91 mám
u Vás "uloženu" žádost o telefon. Přitom více než rok se nic neděje.
Věřím, že zdárně pokračujete v budování nových linek a že přijde den,
kdy se my čekatelé dočkáme.

Chtěl bych v této souvislosti upozornit na to, že v normálně fungujících
státech je běžné, že je žádosti vyhověno do druhého dne a to nejen
v hlavním městě státu.

\vskip 0.6cm
Příjemnou zábavu při čtení urgencí jednotlivých žádostí o telefon přeje

\rightline{\it Ferdinand Mravenec}
\bye
```

---

a výsledek po zpracování TEXem:

V Praze dne 23. září 1992

Vážení telefonní ústředníci!

Dovolte mi, abych Vás upozornil, že pod číslem jednacím 17063/91 mám u Vás „uloženu“ žádost o telefon. Přitom více než rok se nic neděje. Věřím, že zdárně pokračujete v budování nových linek a že přijde den, kdy se my čekatelé dočkáme.

Chtěl bych v této souvislosti upozornit na to, že v normálně fungujících státech je běžné, že je žádosti vyhověno do druhého dne a to nejen v hlavním městě státu.

Příjemnou zábavu při čtení urgencí jednotlivých žádostí o telefon přeje

Ferdinand Mravenec

Poslední věc, která se nám na výstupu nelíbí, je skutečnost, že předložka „o“ je umístěna na konci řádku. To ve slušných tiskovinách nebývá zvykem. Mezi předložku a další slovo je třeba vložit speciální znak, který zakazuje TEXu v příslušném místě zlomit řádek. Tímto znakem je vlnka: ‘`~`’. V našem případě tedy píšeme: `o~telefon`. Ferda ovšem nebude opravovat všechny výskyty takových předložek ručně v editoru, ale z „menu“ spustí program `vlnka`, který do souboru `dopis.tex` doplní vlnky za všechny předložky „k, o, u, v, s, z“ automaticky. Potom naposledy spustí TEX a konečně vybere z „menu“ položku `Print`, která spustí program, jehož jediným úkolem je vytisknout výsledek TEXu na tiskárnu.

My se s Ferdou loučíme a plni dojmů odcházíme.