

# Základy počítačové sazby v systémech $\text{T}_{\text{E}}\text{X}$ a $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

Roman Plch  
katedra matematiky PřF MU

8. prosince 2005

## Vznik a historie TeXu

- 1977, „Mathematics books and journals do not look as beautiful as they used to“, *Donald Ervin Knuth*, profesor Standforské univerzity.

## Vznik a historie TeXu

- 1977, „Mathematics books and journals do not look as beautiful as they used to“, *Donald Ervin Knuth*, profesor Standforské univerzity.
- 1978, první verze

## Vznik a historie TeXu

- 1977, „Mathematics books and journals do not look as beautiful as they used to“, *Donald Ervin Knuth*, profesor Standforské univerzity.
- 1978, první verze
- 1980, založena organizace sdružující uživatele TeXu, TUG (TeX Users Group)

## Vznik a historie TeXu

- 1977, „Mathematics books and journals do not look as beautiful as they used to“, *Donald Ervin Knuth*, profesor Standforské univerzity.
- 1978, první verze
- 1980, založena organizace sdružující uživatele TeXu, TUG (TeX Users Group)
- 1990, vzniká CSTUG

## Vznik a historie TeXu

- 1977, „Mathematics books and journals do not look as beautiful as they used to“, *Donald Ervin Knuth*, profesor Standforské univerzity.
- 1978, první verze
- 1980, založena organizace sdružující uživatele TeXu, TUG (TeX Users Group)
- 1990, vzniká CSTUG
- 1992, zmrazení dalšího vývoje

## Vznik a historie TeXu

- 1977, „Mathematics books and journals do not look as beautiful as they used to“, *Donald Ervin Knuth*, profesor Standforské univerzity.
- 1978, první verze
- 1980, založena organizace sdružující uživatele TeXu, TUG (TeX Users Group)
- 1990, vzniká CSTUG
- 1992, zmrazení dalšího vývoje
- Dnes už Knuth TeX dále nevyvíjí, ale pouze opravuje chyby, o kterých se dozví (a kterých je dnes však už velice málo). Aktuální verze je 3,14159.

## Vznik a historie TeXu

- 1977, „Mathematics books and journals do not look as beautiful as they used to“, *Donald Ervin Knuth*, profesor Standforské univerzity.
- 1978, první verze
- 1980, založena organizace sdružující uživatele TeXu, TUG (TeX Users Group)
- 1990, vzniká CSTUG
- 1992, zmrazení dalšího vývoje
- Dnes už Knuth TeX dále nevyvíjí, ale pouze opravuje chyby, o kterých se dozví (a kterých je dnes však už velice málo). Aktuální verze je 3,14159.



# Co je T<sub>E</sub>X?

- sázecí autorský systém

# Co je T<sub>E</sub>X?

- sázecí autorský systém
- programovatelný, t.j. s vlastním makrojazykem (byl v něm například pro zábavu implementován interpret jazyka BASIC)

# Co je T<sub>E</sub>X?

- sázecí autorský systém
- programovatelný, t.j. s vlastním makrojazykem (byl v něm například pro zábavu implementován interpret jazyka BASIC)
- dávkový: ze vstupu  $\$c=\sqrt{a^2+b^2}\$$  dostaneme  $c = \sqrt{a^2 + b^2}$

# Co je T<sub>E</sub>X?

- sázecí autorský systém
- programovatelný, t.j. s vlastním makrojazykem (byl v něm například pro zábavu implementován interpret jazyka BASIC)
- dávkový: ze vstupu  $\$c=\sqrt{a^2+b^2}\$$  dostaneme  $c = \sqrt{a^2 + b^2}$
- portabilní (od Atari či dvoudisketového PC XT po Cray)

# Co je T<sub>E</sub>X?

- sázecí autorský systém
- programovatelný, t.j. s vlastním makrojazykem (byl v něm například pro zábavu implementován interpret jazyka BASIC)
- dávkový: ze vstupu  $\$c=\sqrt{a^2+b^2}\$$  dostaneme  $c = \sqrt{a^2 + b^2}$
- portabilní (od Atari či dvoudisketového PC XT po Cray)
- stabilní (\$256 za nalezení chyby)

# Co je T<sub>E</sub>X?

- sázecí autorský systém
- programovatelný, t.j. s vlastním makrojazykem (byl v něm například pro zábavu implementován interpret jazyka BASIC)
- dávkový: ze vstupu  $\$c=\sqrt{a^2+b^2}\$$  dostaneme  $c = \sqrt{a^2 + b^2}$
- portabilní (od Atari či dvoudisketového PC XT po Cray)
- stabilní (\$256 za nalezení chyby)
- dobře dokumentovaný (vyšel knižně)

# Co je T<sub>E</sub>X?

- sázecí autorský systém
- programovatelný, t.j. s vlastním makrojazykem (byl v něm například pro zábavu implementován interpret jazyka BASIC)
- dávkový: ze vstupu  $\$c=\sqrt{a^2+b^2}\$$  dostaneme  $c = \sqrt{a^2 + b^2}$
- portabilní (od Atari či dvoudisketového PC XT po Cray)
- stabilní (\$256 za nalezení chyby)
- dobře dokumentovaný (vyšel knižně)
- volně šiřitelný
- s výstupem na zařízení nezávislým

## Co T<sub>E</sub>X není

- editor
- program na grafiku (na to slouží komplementární program METAFONT přibližně stejné velikosti)
- WYSIWYG (nastavby jako LyX existují)
- rychle naučitelný



# Algoritmy

- **Řádkový zlom** –  $\text{T}_{\text{E}}\text{X}$  načte do paměti celý obsah odstavce jako jednu linku a několikrát se ho pokusí nalámat. Bere přitom v potaz mnoho proměnných a typografických pravidel. Matematicky řečeno hledá hledá minimum jakési „cenové“ funkce.

# Algoritmy

- **Řádkový zlom** –  $\text{T}_{\text{E}}\text{X}$  načte do paměti celý obsah odstavce jako jednu linku a několikrát se ho pokusí nalámat. Bere přitom v potaz mnoho proměnných a typografických pravidel. Matematicky řečeno hledá minimum jakési „cenové“ funkce.
- **Dělení slov** –  $\text{T}_{\text{E}}\text{X}$  může současně pracovat s maximálně 256 jazyky. Má uloženy tabulky se vzory dělení slov. Algoritmus pro dělení slov navrhl Frank M. Liang.

# Algoritmy

- **Řádkový zlom** –  $\text{T}_{\text{E}}\text{X}$  načte do paměti celý obsah odstavce jako jednu linku a několikrát se ho pokusí nalámat. Bere přitom v potaz mnoho proměnných a typografických pravidel. Matematicky řečeno hledá minimum jakési „cenové“ funkce.
- **Dělení slov** –  $\text{T}_{\text{E}}\text{X}$  může současně pracovat s maximálně 256 jazyky. Má uloženy tabulky se vzory dělení slov. Algoritmus pro dělení slov navrhl Frank M. Liang.
- **Stránkový zlom** – vzhledem k exponenciální složitosti algoritmu optimalizujícího vzhled celého dokumentu  $\text{T}_{\text{E}}\text{X}$  optimalizuje vzhled jediné stránky. Nastavením parametrů lze určit vhodnost zlomu v tom kterém místě.

# Algoritmy

- **Řádkový zlom** –  $\text{T}_{\text{E}}\text{X}$  načte do paměti celý obsah odstavce jako jednu linku a několikrát se ho pokusí nalámat. Bere přitom v potaz mnoho proměnných a typografických pravidel. Matematicky řečeno hledá minimum jakési „cenové“ funkce.
- **Dělení slov** –  $\text{T}_{\text{E}}\text{X}$  může současně pracovat s maximálně 256 jazyky. Má uloženy tabulky se vzory dělení slov. Algoritmus pro dělení slov navrhl Frank M. Liang.
- **Stránkový zlom** – vzhledem k exponenciální složitosti algoritmu optimalizujícího vzhled celého dokumentu  $\text{T}_{\text{E}}\text{X}$  optimalizuje vzhled jediné stránky. Nastavením parametrů lze určit vhodnost zlomu v tom kterém místě.
- **Matematická sazba** – Knuth věnoval sazbě matematiky mimořádnou pozornost. Propracovanost matematické sazby nemá obdoby v žádném jiném (ani komerčním) systému.

# Algoritmy

- **Řádkový zlom** –  $\text{T}_{\text{E}}\text{X}$  načte do paměti celý obsah odstavce jako jednu linku a několikrát se ho pokusí nalámat. Bere přitom v potaz mnoho proměnných a typografických pravidel. Matematicky řečeno hledá minimum jakési „cenové“ funkce.
- **Dělení slov** –  $\text{T}_{\text{E}}\text{X}$  může současně pracovat s maximálně 256 jazyky. Má uloženy tabulky se vzory dělení slov. Algoritmus pro dělení slov navrhl Frank M. Liang.
- **Stránkový zlom** – vzhledem k exponenciální složitosti algoritmu optimalizujícího vzhled celého dokumentu  $\text{T}_{\text{E}}\text{X}$  optimalizuje vzhled jediné stránky. Nastavením parametrů lze určit vhodnost zlomu v tom kterém místě.
- **Matematická sazba** – Knuth věnoval sazbě matematiky mimořádnou pozornost. Propracovanost matematické sazby nemá obdoby v žádném jiném (ani komerčním) systému.

$\text{\TeX}$  jako takový obsahuje zhruba 300 základních značkovacích příkazů. Pomocí makrojazyka, který je součástí  $\text{\TeX}$ u, můžeme vytvářet nové sázecí příkazy. Takto vznikla také spousta více či méně známých formátů  $\text{\TeX}$ u:

- plain $\text{\TeX}$

Tento formát naprogramoval sám autor  $\text{\TeX}$ u Donald Knuth. Plain $\text{\TeX}$  rozšiřuje počet příkazů, které může uživatel použít, na devět set. Vyžaduje schopnost programátorského myšlení, protože většinu maker si musí uživatel vytvořit a odladit sám.

$\text{T}_{\text{E}}\text{X}$  jako takový obsahuje zhruba 300 základních značkovacích příkazů. Pomocí makrojazyka, který je součástí  $\text{T}_{\text{E}}\text{X}$ u, můžeme vytvářet nové sázecí příkazy. Takto vznikla také spousta více či méně známých formátů  $\text{T}_{\text{E}}\text{X}$ u:

- plain $\text{T}_{\text{E}}\text{X}$

Tento formát naprogramoval sám autor  $\text{T}_{\text{E}}\text{X}$ u Donald Knuth. Plain $\text{T}_{\text{E}}\text{X}$  rozšiřuje počet příkazů, které může uživatel použít, na devět set. Vyžaduje schopnost programátorského myšlení, protože většinu maker si musí uživatel vytvořit a odladit sám.

- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

Je ndstavbou  $\text{T}_{\text{E}}\text{X}$ u vytvořenou panem Leslie Lamportem. Byl vytvořen zejména proto, aby zjednodušil sazbu dokumentů v  $\text{T}_{\text{E}}\text{X}$ u a zpřístupnil tak jinak poněkod složitý jazyk běžnému uživateli.

$\text{T}_{\text{E}}\text{X}$  jako takový obsahuje zhruba 300 základních značkovacích příkazů. Pomocí makrojazyka, který je součástí  $\text{T}_{\text{E}}\text{X}$ u, můžeme vytvářet nové sázecí příkazy. Takto vznikla také spousta více či méně známých formátů  $\text{T}_{\text{E}}\text{X}$ u:

- plain $\text{T}_{\text{E}}\text{X}$

Tento formát naprogramoval sám autor  $\text{T}_{\text{E}}\text{X}$ u Donald Knuth. Plain $\text{T}_{\text{E}}\text{X}$  rozšiřuje počet příkazů, které může uživatel použít, na devět set. Vyžaduje schopnost programátorského myšlení, protože většinu maker si musí uživatel vytvořit a odladit sám.

- $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

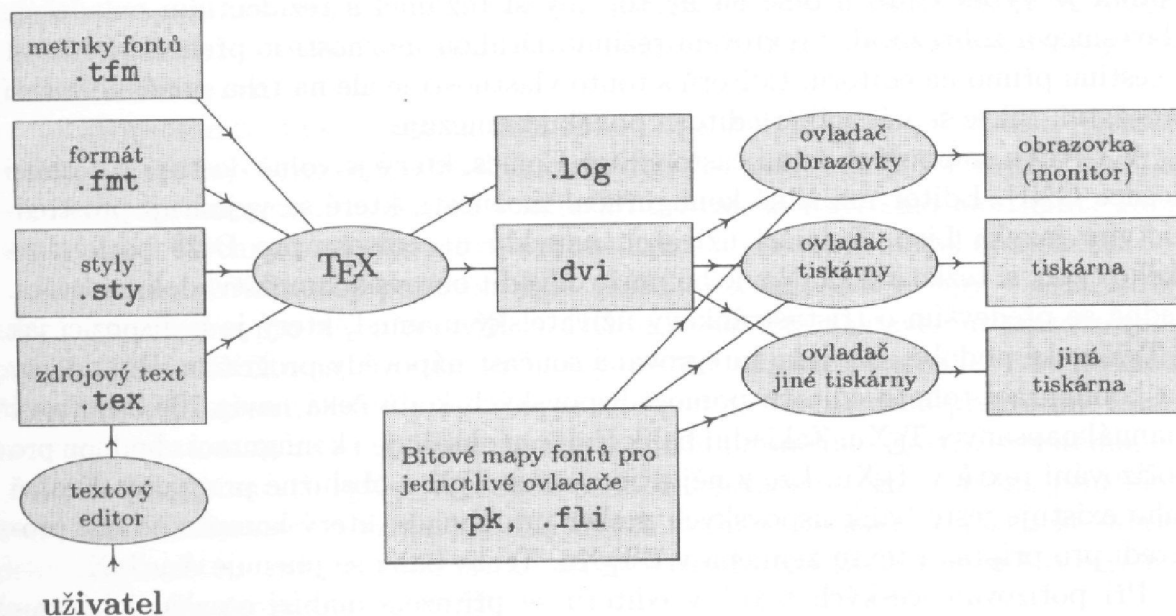
Je ndstavbou  $\text{T}_{\text{E}}\text{X}$ u vytvořenou panem Leslie Lamportem. Byl vytvořen zejména proto, aby zjednodušil sazbu dokumentů v  $\text{T}_{\text{E}}\text{X}$ u a zpřístupnil tak jinak poněkod složitý jazyk běžnému uživateli.

- $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\text{T}_{\text{E}}\text{X}$

Formát napsaný původně pro Americkou Matematickou společnost.



# Tvorba dokumentu



- Editorem vytváříme zdrojový dokument `.tex`.  
(PSPAD: <http://www.pspad.com/>).
- Zdrojový dokument zpracujeme  $\text{T}_{\text{E}}\text{X}$ em – přitom se načítá soubor zvaný formát (`.fmt`) – předzpracovaná sbírka  $\text{T}_{\text{E}}\text{X}$ ových maker k rychlému natažení. Další nezbytností jsou písma. Samotný  $\text{T}_{\text{E}}\text{X}$  se však nestará o to, jak takové písmeno vypadá. Proto  $\text{T}_{\text{E}}\text{X}$  nenačítá samotná písma, ale pouze informaci o šířkách a výškách písmen, ligaturách, sklonu a dalších hodnotách. Tato souhrnná informace o písmu se nazývá metrika. V případě  $\text{T}_{\text{E}}\text{X}$ u má příponu `.tfm`.
- Po skončení práce  $\text{T}_{\text{E}}\text{X}$ u získáme (kromě pomocných souborů) soubor DVI (DeVice Independent). Je to soubor, který popisuje, kam se  $\text{T}_{\text{E}}\text{X}$  rozhodl umístit texty. Kromě toho může obsahovat speciální značky pro výstupní zařízení, jejichž pomocí se do  $\text{T}_{\text{E}}\text{X}$ u implementují barvy, podpora obrázků a další. Tento soubor si již můžeme prohlédnout DVI prohlížečem.
- Často jej však použijeme pro převod do jiného výstupního formátu – PostScriptu (pomocí programu `dvips`).

- Postscript (PS):

Jazyk PostScript byl navržen firmou Adobe v roce 1985 jako jazyk pro popis stránek. Díky svým rozsáhlým možnostem se však brzy stal i formátem používaným k přenosu obrázků mezi aplikacemi.

- Postscript (PS):

Jazyk PostScript byl navržen firmou Adobe v roce 1985 jako jazyk pro popis stránek. Díky svým rozsáhlým možnostem se však brzy stal i formátem používaným k přenosu obrázků mezi aplikacemi.

- zapouzdřený PostScript (Encapsulated PostScript – EPS):

Od normálního strukturovaného PostScriptu se příliš neliší – jediný zásadní rozdíl je v tom, že neobsahuje žádné příkazy pro nastavení zařízení či souřadného systému a nemanipuluje s interpretem tak, aby to ovlivnilo zpracování ostatních částí dokumentu. Některé PostScriptové příkazy tedy formát EPS zcela zakazuje, jiné omezuje. Aplikace pro EPS připraví prostředí (souřadný systém) tak, aby byl obrázek ve správné poloze a velikosti. K tomu jí napomáhá informace o ohraničujícím rámu (Bounding Box), která se nachází ve strukturovaných komentářích každého EPS souboru.

- Postscript (PS):

Jazyk PostScript byl navržen firmou Adobe v roce 1985 jako jazyk pro popis stránek. Díky svým rozsáhlým možnostem se však brzy stal i formátem používaným k přenosu obrázků mezi aplikacemi.

- zapouzdřený PostScript (Encapsulated PostScript – EPS):

Od normálního strukturovaného PostScriptu se příliš neliší – jediný zásadní rozdíl je v tom, že neobsahuje žádné příkazy pro nastavení zařízení či souřadného systému a nemanipuluje s interpretem tak, aby to ovlivnilo zpracování ostatních částí dokumentu. Některé PostScriptové příkazy tedy formát EPS zcela zakazuje, jiné omezuje. Aplikace pro EPS připraví prostředí (souřadný systém) tak, aby byl obrázek ve správné poloze a velikosti. K tomu jí napomáhá informace o ohraničujícím rámu (Bounding Box), která se nachází ve strukturovaných komentářích každého EPS souboru.

- Portable Document Format (PDF):

Interpretace postscriptového kódu je poměrně složitá a relativně pomalá. PostScript je navíc textový formát, což zvyšuje velikost souborů. Formát PDF přebral ideu popisu stránky, ovšem vyřadil všechny programovací konstrukce. Naproti tomu přidal podporu komprese celého kódu, kryptografie, náhledů stran, hypertextových odkazů, appletů, barevných profilů ICC aj.

# Zdroje informací, software

- $\mathcal{C}_g$ TUG: <http://www.cstug.cz/>
- Archívy CTAN a  $\mathcal{C}_g$ TUG:  
<http://www.cstug.cz/ctan/index.html>
- Konference a news skupiny:  
<http://www.cstug.cz/diskuze-index.html>
- Kurs  $\text{\LaTeX}$ u: <http://www.math.muni.cz/~plch/vyuka/tex/tex.html>
- Ghostview, Ghostscript:  
<http://www.cs.wisc.edu/~ghost/index.html>
- Adobe Reader:  
<http://www.adobe.com/products/acrobat/readstep2.html>
- $\text{\LaTeX}$ 2html: <http://www.latex2html.org/>
- PDFCreator:  
<http://www.slunecnice.cz/product/PDFCreator/>



# Literatura

- [1] Beran V.: *Aktualizovaný typografický manuál*, nakladatelství Manuál 2003.
- [2] Čechová L., Plch R.: *Sázíme v  $\LaTeX$ u diplomovou prací z matematiky*, skriptum MU Brno 2003.
- [3] Goossens M., Rahtz S., Mittelbach F.: *The  $\LaTeX$  Companion*, Addison-Wesley 1994.
- [4] Goossens M., Rahtz S., Mittelbach F.: *The  $\LaTeX$  Graphics Companion*, Addison-Wesley 1997.
- [5] Grätzer G.: *Math into  $\LaTeX$* , Third Edition, Birkhäuser Boston 2000.
- [6] Kočer M.: *Ne příliš stručný úvod do systému  $\LaTeX$ 2e*, 1998.  
<http://www.penguin.cz/~kocer/texty/lshort2e/lshort2e-cz.pdf>
- [7] Olšák P.: *Typografický systém  $T_{E}X$* , Konvoj Brno, 2000.
- [8] Olšák P.: *Jak  $T_{E}X$  pracuje s PostScriptem*, Zpravodaj Československého sdružení uživatelů  $T_{E}X$ u, **3** (3), 101–113, 1993.
- [9] Rybička J.:  *$\LaTeX$  pro začátečníky*, 3. vydání, KONVOJ, Brno 2003.