

# Zpravodaj Československého sdružení uživatelů TeXu

---

Ladislav Dobiáš

Začínáme s METAPOSTem aneb Udělejte si vlastní logo

*Zpravodaj Československého sdružení uživatelů TeXu*, Vol. 8 (1998), No. 3-4, 167–175

Persistent URL: <http://dml.cz/dmlcz/149828>

## Terms of use:

© Československé sdružení uživatelů TeXu, 1998

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

Martině a panu P. Olšákovi touto cestou poděkovat za jejich pomoc a podporu.

Na mě pak už zbylo pouze všechny připomínky a korektury přenést do elektronické verze dokumentu a vše vystavit ve zdrojové formě, ve formě PDF, HTML, PS a DVI na URL <http://www.cstug.cz/pub/CTAN/info/lshort/czech/>.

Nyní, po zveřejnění, už jen čekám, až se ozve první nespokojený čtenář, který bude mít konstruktivní připomínky, které povedou k dalšímu, lepšímu, dokonalejšímu *Stručnému úvodu do systému  $\text{\LaTeX}$  2<sub>ε</sub>*.

Michal Kočer  
kocer@nime.cz

---

---

## Začínáme s METAPOSTem aneb Udělejte si vlastní logo

---

LADISLAV DOBIÁŠ

V tomto dokumentu si řekneme o tom, co je to METAPOST, co umožňuje, jak se dá nainstalovat, a ukážeme si několik jednoduchých příkladů, podle kterých si snadno vyrobíte i nějaké vlastní logo.

### Možnosti METAPOSTu

METAPOST je program se silným jazykem pro kreslení obrázků a výstupem do PostScriptu – jazyku moderních tiskáren.

METAPOST přebírá část zdrojového kódu z programu METAFONT (samozřejmě se svolením autora – D. E. Knutha) a tím i většinu příkazů METAFONTu. Navíc umožňuje plně využít možnosti PostScriptu.

Protože METAPOST obsahuje část METAFONTu, nabízí i stejné základní grafické možnosti a práci s nimi. Tedy umožňuje řešit soustavy lineárních rovnic, obsahuje objekty pro křivky, obrázky (pictures), transformace, tvary pera. Navíc přibírá PostScriptové věci jako barvy, výřezy, stínování, čárkované čáry.

Z těchto základních věcí jde udělat snad vše, co dokáže papír snést. Ovšem někdy to může být dosti náročné. A protože je v METAPOSTu silný programovací jazyk, jsou na Internetu dostupná různá makra, která ulehčují práci s grafickými objekty.

Standardně jsou u METAPOSTu makra na grafy. Také existují například makra na animaci objektů, které mohou být i 3D (třírozměrné). V literatuře [5]

se udává, že „D. Knuth povedal, že nepoužíva nič iné na kreslenie diagramov, keď píše nejaké texty.“

Výstup z METAPOSTu lze jednoduše použít v sázecích systémech jako  $\text{T}_{\text{E}}\text{X}$ , *troff* apod.

O „opačném toku dat“ – o možnosti vkládat kód  $\text{T}_{\text{E}}\text{X}$ u do METAPOSTu se zmíním později.

## Instalace METAPOSTu

Systém METAPOST je součástí distribuce  $\text{T}_{\text{E}}\text{X}$  pro UNIX od Karla Berryho *web2c*. Pokud ji máte nainstalovanou, máte pravděpodobně nainstalován i METAPOST jako program *mpost*.

Balík *web2c* s podporou pro českou sazbu lze nalézt např. na <ftp://math.feld.cvut.cz/pub/cstex/unix>.

S instalací pod DOSem nemám zkušenosti, vím jen, že existuje již přeložený METAPOST na CTANu v adresáři *graphics/metapost/dos*.

## Rozdíly mezi METAPOSTem a METAFONTEM

METAFONTovým výstupem je bitmapa, zatímco z METAPOSTu vzniká PostScriptový kód. Proto v METAPOSTu odpadají tzv. ostré jednotky (*sharp units*).

METAFONT je primárně určen pro generování fontů pro  $\text{T}_{\text{E}}\text{X}$ , v METAPOSTu je tato funkce potlačena do pozadí. Proto například práce s *.tfm* soubory a podobné věci jsou v METAPOSTu dostupné až po vložení souboru *mfplain.mp*.

Dále odpadají možnosti METAFONTu zobrazit obrázky na terminálu.

Do METAPOSTu jde navíc vkládat text, který se někde umísťuje, proto tu jsou příkazy pro práci s fonty.

O využívání PostScriptových možností jsem se zmínil výše, některé z nich (práce s barvami a vyřezávání) si ukážeme v příkladech.

Přesný popis rozdílů lze nalézt v literatuře [3] s odkazy na základní knihu o METAFONTu [4].

## Spuštění a jednoduchý příklad grafu

Předtím než si ukážeme, jak se METAPOST spouští, si musíme říci, jak vypadá program pro METAPOST, a jednoduchý program si ukážeme.

Program v METAPOSTu obecně obsahuje jednotlivé obrázky začínající *beginfig* (*< číslo >*); a končící *endfig*; Program končí většinou příkazem *end*;

Standardně je u METAPOSTu soubor *maker graph.mp*, který umožňuje jednoduše vytvářet grafy.

Jednoduchý graf vytvoříte programem `graf.mp`:

---

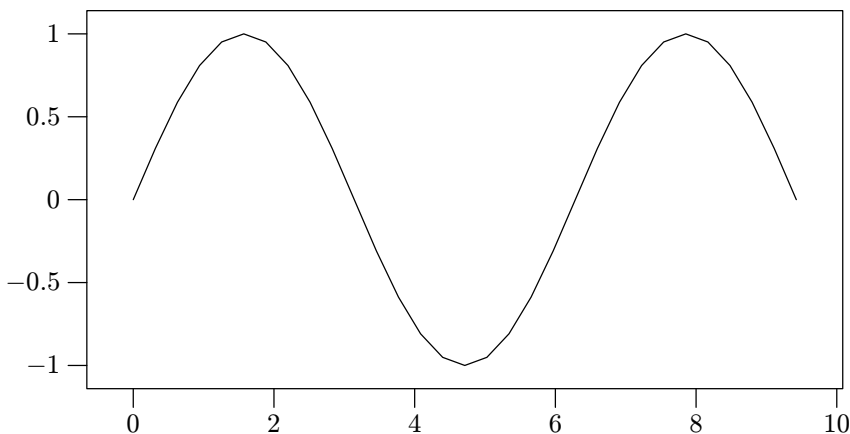
```
beginfig(1);
input graph;
draw begingraph(10cm,5cm);
  gdraw "graf1.d";
endgraph;
endfig;
end;
```

---

METAPOST spustíme pomocí `mpost graf.mp` a vznikne PostScriptový soubor `graf.1`.

Pokud teď chcete obrázek vložit do svého T<sub>E</sub>Xovského dokumentu, napište na začátek dokumentu `\input epsf` a pak někam vložte `$$\epsfbox{graf.1}$$` a máte graf vložen.

Graf vypadá takto:



Poznámky k programu:

- `input` vkládá soubor maker `graph.mp`
- dále definujeme jen celkovou velikost grafu a soubor, z něhož se mají data pro graf číst
- soubor `graf1.d` obsahuje 2 sloupce čísel oddělených mezerami (souřadnice  $x$  a  $y$ )

## Příklad 1 – nota

Bez další teorie si ukážeme jednoduchý příklad na spojování bodů.

---

```

beginfig(1);
width=.5cm;
height=.3cm;
stem=1cm;

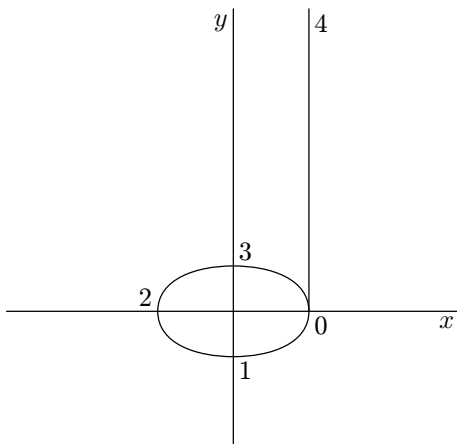
y0=x1=y2=x3=0;
-x2=x0=x4=width/2;
-y1=y3=height/2;
y4=stem;
draw z0..z1..z2..z3..cycle;
draw z0--z4;
endfig;

end;

```

---

Program vykreslí obrázek podobající se nějaké notě. Zde jsem ho trochu zvětšil a popsals body:



Poznámky k programu:

- `width`, `height`, `stem`, `x0`, `x1`, ... `y0`, `y1`, ... jsou proměnné typu souřadnice
- `z0`, `z1` ... je zkratka pro `(x0,y0)`, `(x1,y1)` ...
- `:=` je operace přiřazení
- `=` definuje závislosti (to jsou ty soustavy lineárních rovnic)
- `draw` spojuje body
- `..` spojuje body Bézierovými křivkami
- `cycle` uzavře křivku
- `--` spojuje body úsečkami

Pokud bychom místo prvního řádku s `draw` napsali `draw z0--z1--z2--z3--cycle;`, dostali bychom místo



„hranatou“ verzi noty



Snad jste viděli, že to je docela lehké.

## Příklad 2 – vlajka

Nyní si ukážeme jednoduchou práci s barvami.

---

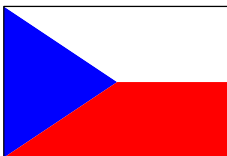
```
vyska:=2cm;
delka=3/2vyska;

beginfig(1);
y1=y2=x1=x4=0;
x2=x3=delka;
y4=y3=vyska;
z5=.5[z1,z3];
z6=.5[z2,z3];
fill z1--z4--z5--cycle withcolor blue;
fill z1--z2--z6--z5--cycle withcolor red;
fill z3--z4--z5--z6--cycle withcolor white;
draw z1--z2--z3--z4--cycle;
endfig;

end;
```

---

Tento program vykreslí naši vlajku



Poznámky k programu:

- $z5=.5[z1,z3]$  znamená, že bod  $z5$  leží „na polovině cesty“ mezi body  $z1$  a  $z3$
- barvy se dají volit pomocí předdefinovaných slov jako **blue**, **red**,... nebo pomocí složek červené, zelené, modré, jako trojice  $(r,g,b)$ , kde jednotlivé složky jsou v rozmezí od 0 do 1.

## Příklad 3 – monáda

Teď si ukážeme práci s cestami, perem a jednoduchou transformaci.

Program

---

```
prumer:=1cm;

beginfig(1);
% Zakladni vlna a kolo
path p;
-x1=-y2=x3=y4=prumer;
x0=y0=y1=x2=y3=x4=0;
p = z1..z2..z3;
fill p{up}..z0{down}..{up}cycle;
draw p..z4..cycle;

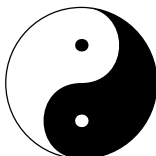
% Tecky
pickup pencircle scaled (prumer/6);
-z5=z6=.5[z0,z3];
draw z5;
undraw z6;

% Prevratime a otocime "na bok"
currentpicture:= currentpicture xscaled -1 rotated 90;
endfig;

end;
```

---

vykreslí znak symbolizující věčné prolínání Jin a Jang.



Poznámky k programu:

- **path p** definuje **p** jako cestu (tu jsme použili již předtím u **draw**, aniž bychom o tom věděli)

- `fill` vyplňuje uzavřenou cestu
- `pickup pencircle scaled (prumer/6)` říká, že použijeme kruhové pero daného průměru
- `undraw` (a podobně `unfill`) maže cestu
- proměnná `currentpicture` obsahuje vše, co jsme nakreslili, a to převrátíme a otočíme o 90 stupňů

## Příklad 4 – logo

Nyní zkusíme udělat nějaké logo, např. pro firmu.

Je zde použito cyklu `for`, typu obrázků a cesta a nakonec tam jsou dvě oříznutí a spojení oříznutých částí ve výsledné logo.

---

```

h:=2cm;          % celkova vyska (i sirka)
n:=22;           % pocet car je (n+1) !!!
wn:=(h/n)/20;    % nejtenci cara (baj voko)
dd:=(h/n)/5;     % dalsi (pridavna) mezera mezi cary
h=d*(n+1);       % d je vzdalenost mezi stredu car
w0=d-wn-dd;      % nejsirsi = sirka - nejtenci - pridavna

beginfig(1);
% zakladni cary
x0=0;
y0=-h/2+d/2; % aby to bylo soumerne okolo 0
for i=0 upto n:
  w:=w0-(w0-wn)/n*i; % zacneme nejtlustsi carou
  draw z0 withpen pensquare yscaled w xscaled h;
  y0:=y0+d;
endfor

% vytvorime obracenou kopii
picture c;
c = currentpicture yscaled -1;

% orizneme na spravny tvar
y1=x2=y3=x4=x5=x6=y7=0;
-x1=-y2=x3=y4;
x3-x1=h;
-y5=y6=x7=(17/32/2)*h; % "baj voko" zmereno pravitkem...

path p; % rozdelujici cara + vybouleni uprosted
p= z2--z5..z7..z6--z4;
clip currentpicture to z1--p--cycle;
clip c to z3--p--cycle;
% spojime obe casti dohromady
addto currentpicture also c;

```

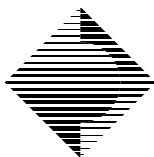


```
endfig;
```

```
end;
```

---

Vyjde z toho takovéhle logo



Poznámky k programu:

- nejdříve nakreslíme základní čáry různé tloušťky, ale se stejnými vzdálenostmi mezi jejich osami (zde je to kresleno jako body různými pery)
- vytvoříme si otočenou kopii
- z originálu necháme (vyřízneme dle cesty) jen levou část loga
- z kopie necháme jen pravou část
- obě části spojíme

(Toto logo má jednu chybu – je to chráněná značka<sup>1</sup>, proto ho nemůžete použít pro vlastní firmu :-)

## Vkládání $\text{\TeX}$ ovského kódu do **METAPOSTu**

Jako součást programu pro METAPOST můžete mezi klíčová slova **btex** a **etex** vkládat příkazy pro  $\text{\TeX}$ . Ty se zpracují preprocesorem, který spouští  $\text{\TeX}$  na pozadí, vytvoří se soubor **.mpx**, kterému již METAPOST rozumí.

Pokud vás přesně zajímá, jak to dělá, stačí udělat mezi **btex** a **etex** nějakou chybu a v aktuálním adresáři zůstane soubor **mpxerr.tex**, ve kterém to uvidíte.

Například, pokud do souboru **chyba.mp** napíšeme **btex \$x etex**, v souboru **mpxerr.tex** bude:

---

```
\shipout\hbox{\smash{\hbox{\hbox{% line 1 chyba.mp
$x}\vrule width1sp}}}}
\end{document}
```

---

Také lze využít příkazů **verbatimtex** a **etex**, mezi které se pouze vloží preprocesorem do dočasněho  $\text{\TeX}$ ovského souboru. To je vhodné např. pro vložení souboru vlastních  $\text{\TeX}$ ovských maker (pomocí **verbatimtex \input mamakra etex**), která se budou později využívat mezi **btex** a **etex**.

---

<sup>1</sup>Logo firmy CZ. TECH s.r.o. je použito se svolením jeho majitele.

## Závěr

Doufám, že se mi podařilo ukázat, že METAPOST je poměrně jednoduše použitelný systém a že ho alespoň někdy vyzkoušíte (třeba pro vytvoření loga vaší firmy).

Pokud by vás METAPOST zaujal, tak přesnější popis a syntaxi příkazů najdete např. v literatuře [3].

## Odkazy

- [1] J. D. Hobby: *The METAPOST System*. Součást distribuce programu METAPOST.
- [2] J. D. Hobby: *Drawing graphs with METAPOST*. Součást distribuce programu METAPOST.
- [3] J. D. Hobby: *A User's Manual for METAPOST*. Součást distribuce programu METAPOST.
- [4] D. E. Knuth: *The METAFONTbook*.
- [5] Často kladené otázky o  $\text{T}_{\text{E}}\text{X}$ u a odpovědi na ně ( *$\zeta\text{S}\text{TUG}\text{FAQ}$* ). Zpravodaj Československého sdružení uživatelů  $\text{T}_{\text{E}}\text{X}$ u, **6** (3), 129–211 (1996).

---

---

# METAPOST

---

ROBERT ŠPALEK

## 1. Co je to METAPOST

METAPOST je jazyk určený pro popis obrázků – technických ilustrací. Výstupem METAPOSTu je program v PostScriptu, který vykreslí na zařízení požadovaný obrázek. Oproti PostScriptu poskytuje mnoho užitečných funkcí, které kreslení zjednodušují a urychlují.

Mezi nejpoužívanější funkce bezesporu patří výpočet souřadnic bodů podle zadaných závislostí, automatické řešení soustav lineárních rovnic, výpočet průsečíků křivek, jejich plynulé navazování, inteligentní prokládání bodů křivkou, vkládání textových popisků (s možností formátování  $\text{T}_{\text{E}}\text{X}$ em a  $\text{TROFF}$ em), kreslení kaligrafickým perem, práce s barvami, vestavěný makrojazyk, ... Při popisu se používá snadno pochopitelný programovací jazyk procedurálního typu.

Vykreslené obrázky je možno použít všude, kde je možno pracovat s PostScriptem, tedy všude. Pomocí maker `epsf` je můžeme vkládat i do  $\text{T}_{\text{E}}\text{X}$ ovských