

# ***C2110 Operační systém UNIX a základy programování***

**10. lekce**

**gnuplot, bash**

**Petr Kulhánek, Jakub Štěpán**

**[kulhanek@chemi.muni.cz](mailto:kulhanek@chemi.muni.cz)**

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta  
Masarykova univerzita, Kamenice 5, CZ-62500 Brno

# Průběžný test II

---

# Průběžný test II

## ➤ Test prostřednictvím odpovědníku v IS

**Student – Odpovědníky – C2110 – Test IIa/b(-EN)**

**Délka 20 minut.**

**Je možné sestavit pouze jednu sadu otázek.**

**Používejte průběžné uložení.**

**Vyhodnocení je možné pouze jednou.**

**Je povoleno a doporučeno:**

- Testovat příkazy v terminálu.
- Prohledávat manuálové stránky, svoje zápisky a prezentace předmětu.
- Při nejasnostech se přihlaste.

**Není povoleno**

- Komunikovat s další osobou mimo vyučujícího.

## ➤ Gnuplot

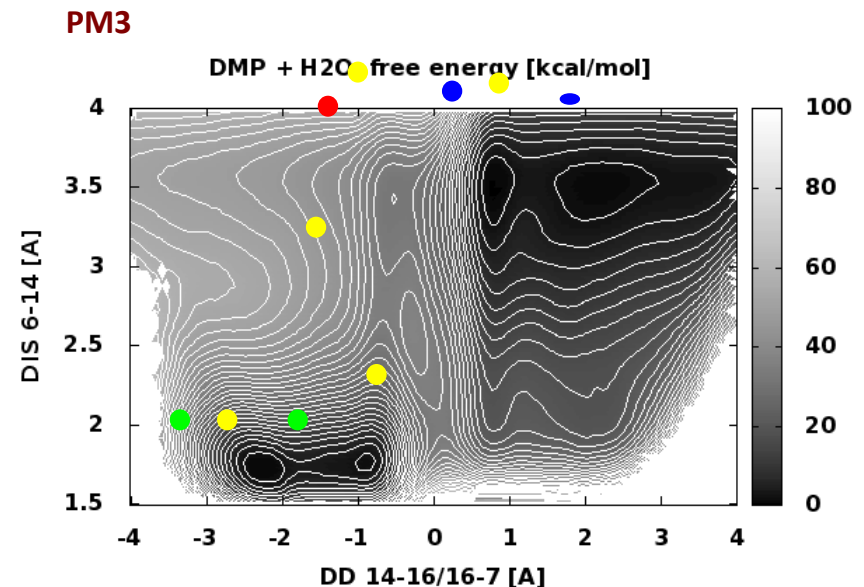
- **přehled jazyka, příkaz plot, terminály, příkaz splot**

# Gnuplot

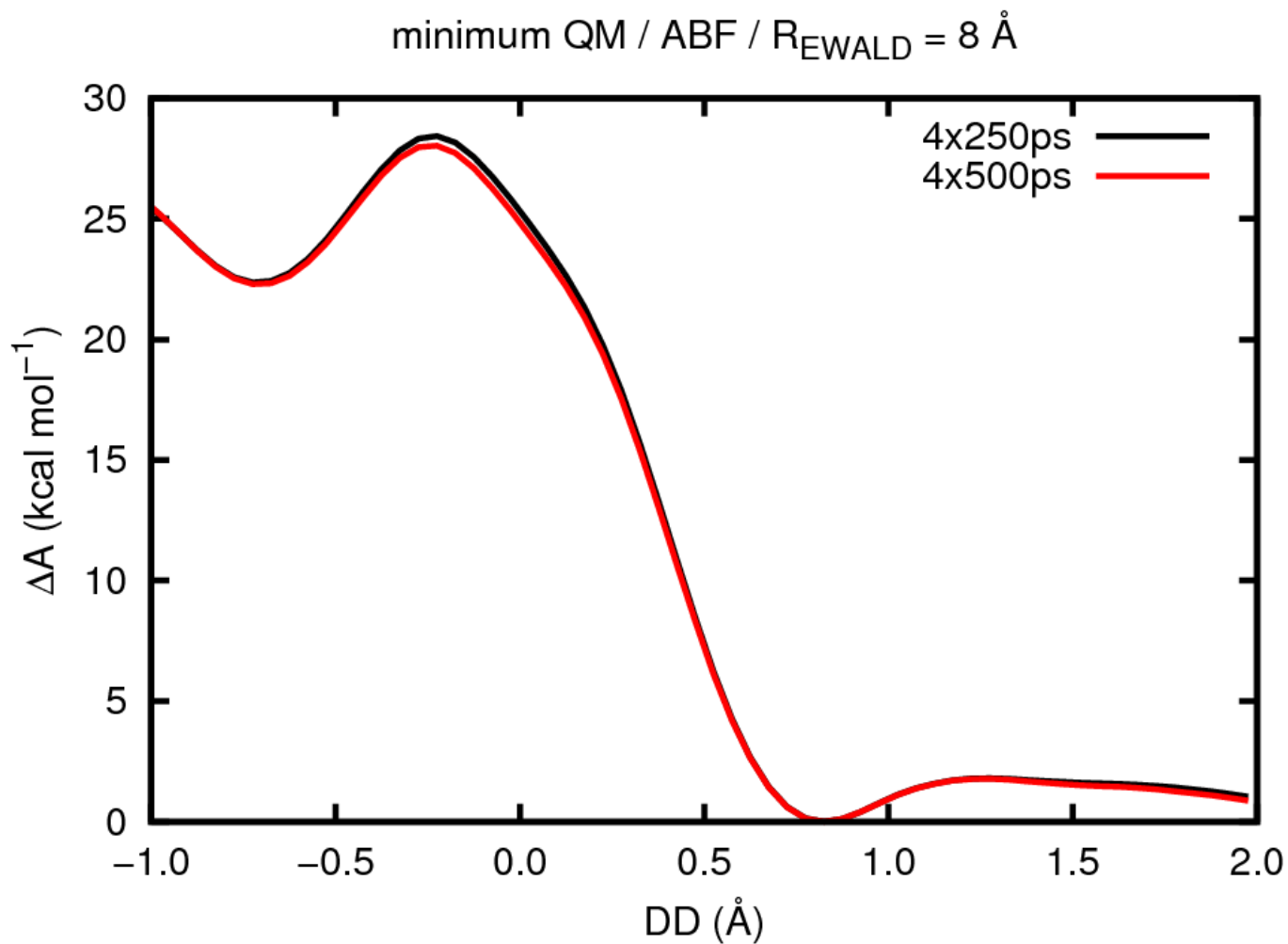
- (Ne)Interaktivní spouštění
- Příkaz plot
- Terminály
- Ukázky

<http://www.gnuplot.info/>

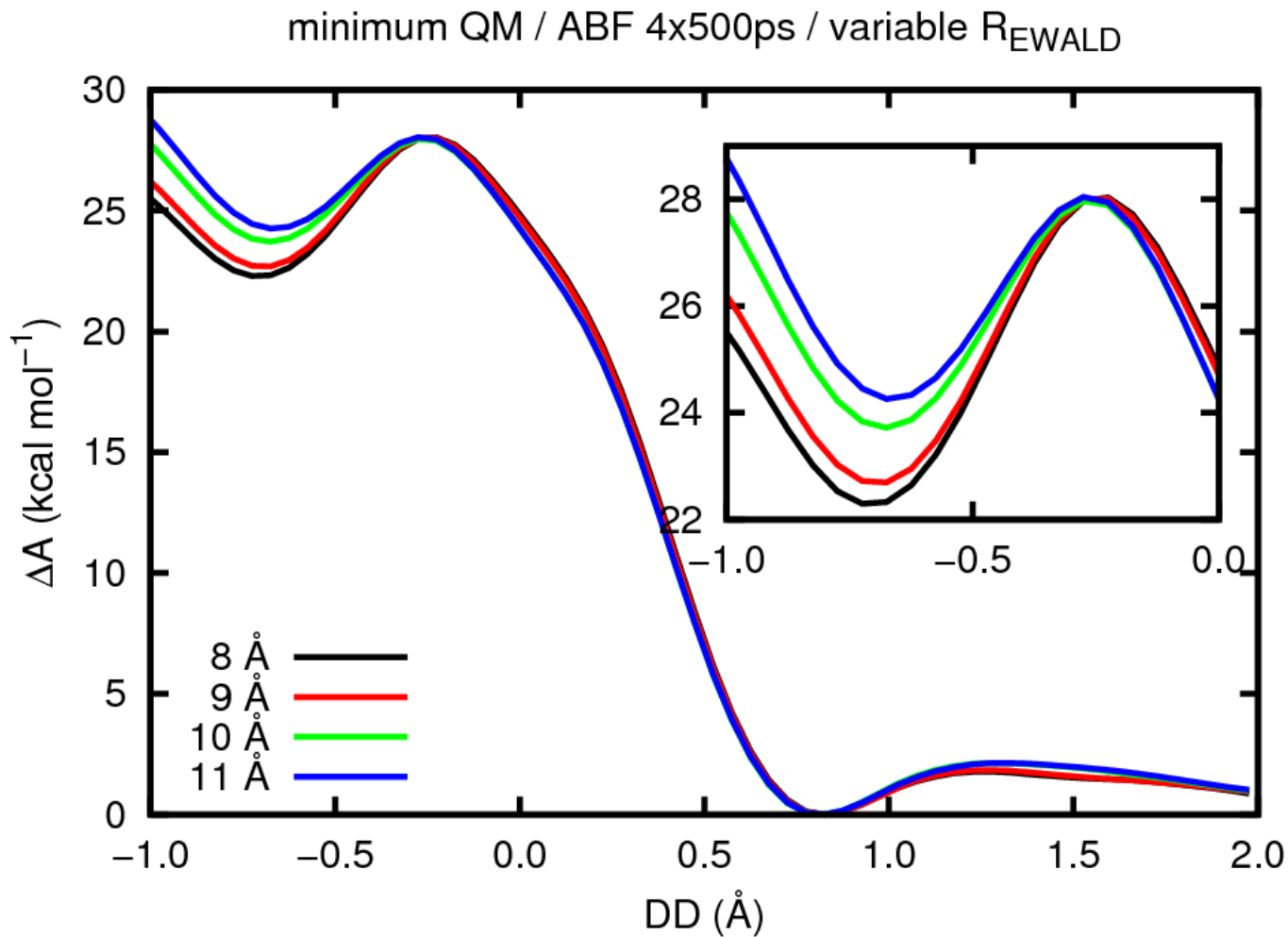
(dokumentace, tutoriály, zdrojové kódy)



# Ukázky



# Ukázky



# Interaktivní spouštění

**Gnuplot** slouží k vykreslování 2D a 3D grafů umožňující práci v interaktivním tak i skriptovacím režimu.

## Interaktivní mód

příkazová řádka shellu Bash

```
[kulhanek@wolf ~]$ gnuplot
```

```
G N U P L O T
Version 4.4 patchlevel 3
last modified March 2011
System: Linux 3.2.0-31-generic
```

```
Copyright (C) 1986-1993, 1998, 2004, 2007-2010
Thomas Williams, Colin Kelley and many others
```

```
gnuplot home:      http://www.gnuplot.info
faq, bugs, etc:    type "help seeking-assistance"
immediate help:    type "help"
plot window:       hit 'h'
```

```
Terminal type set to 'wxt'
gnuplot>
```

příkazová řádka gnuplotu



# Neinteraktivní spouštění

## 1) Nepřímé spouštění

Spouštíme interpreter jazyka a jako argument uvádíme jméno skriptu.

```
$ gnuplot muj_skript_v_gnuplotu
```

Skripty **nemusí** mít nastaven příznak x (executable).

## 2) Přímé spouštění

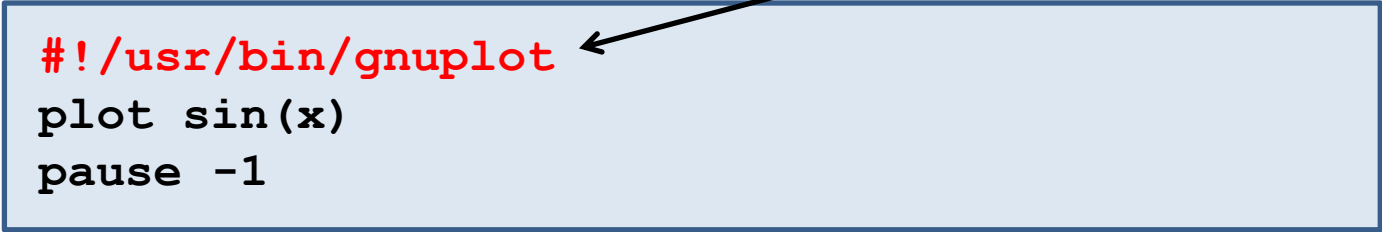
Spouštíme přímo skript (shell automaticky spustí interpreter).

```
$ chmod u+x muj_skript_v_gnuplotu
```

```
$ ./muj_skript_v_gnuplotu
```

Skripty **musí** mít nastaven příznak **x** (executable) a **interpreter** (součást skriptu).

```
#!/usr/bin/gnuplot  
plot sin(x)  
pause -1
```



# Příkaz - plot

> `plot funkce/soubor [nastaveni_zobrazeni] [, fce/soubor ...]`

Zobrazí XY graf funkce nebo datové řady obsažené v souboru.

## Příklady:

lines, points, linespoints, dots

barva čáry

> `plot sin(x)`

> `plot cos(5.7*x+3.4) with points linecolor rgb "red" \  
linewidth 2 title "cos"`

název souboru s daty

tloušťka čáry

legenda

> `plot "input.txt" using 1:2 with lines`

druhý sloupec tvoří y-ové hodnoty

první sloupec tvoří x-ové hodnoty

> `plot sin(x), cos(x)`

zobrazí funkci sin a cos do jednoho grafu

# Cvičení

1. Znázorněte průběh funkce  $y=x^2$
2. Průběh funkce z prvního cvičení zobrazte modrou barvou
3. Zobrazte průběh teploty v čase obsažený v souboru **/home/kulhanek/Data/temp.txt**  
Čas je uveden v prvním sloupci, teplota je uvedena v druhém sloupci.
4. Zobrazte do jednoho grafu funkci  $\sin(x)$  pomocí červené čáry a funkci  $\cos(x)$  pomocí oranžové čáry a bodů.

**Úlohy řešte v interaktivním režimu.**

# Další příkazy

- > `set title "popis"` # záhloví grafu
- > `set xrange [min_value:max_value]` # nastaví rozsah x-ové osy
- > `set xlabel "popis"` # nastaví popis x-ové osy
- > `set yrange [min_value:max_value]` # nastaví rozsah y-ové osy
- > `set ylabel "popis"` # nastaví popis y-ové osy
- > `set nokey` # nezobrazí legendu k datovým řadám
- > `pause -1` # čeká na zmáčknutí klávesy

# Cvičení

1. Napište skript, který znázorní průběh funkce  $y=x^2$  v rozsahu 0-10 pro x-ovou hodnotu. Skript spusťte nepřímo pomocí interpretru gnuplot.
2. Napište skript, který zobrazí průběh teploty v čase obsažený v souboru **/home/kulhanek/Data/temp.txt** . V grafu popište osy včetně určení jednotek. Čas je uveden v picosekundách, teplota v kelvinech.

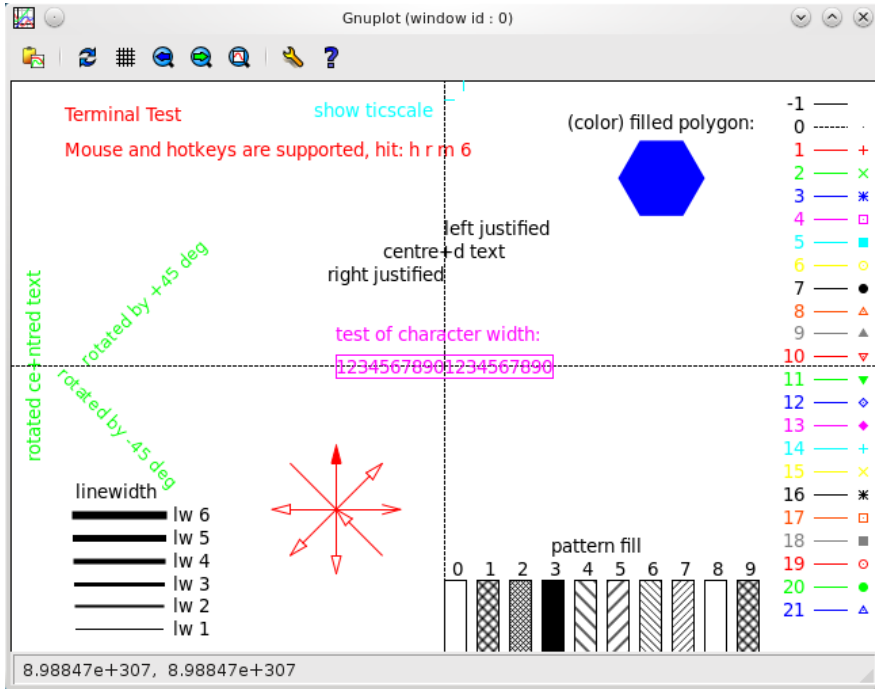
# Terminály

**Terminál** určuje kam bude graf vykreslen.

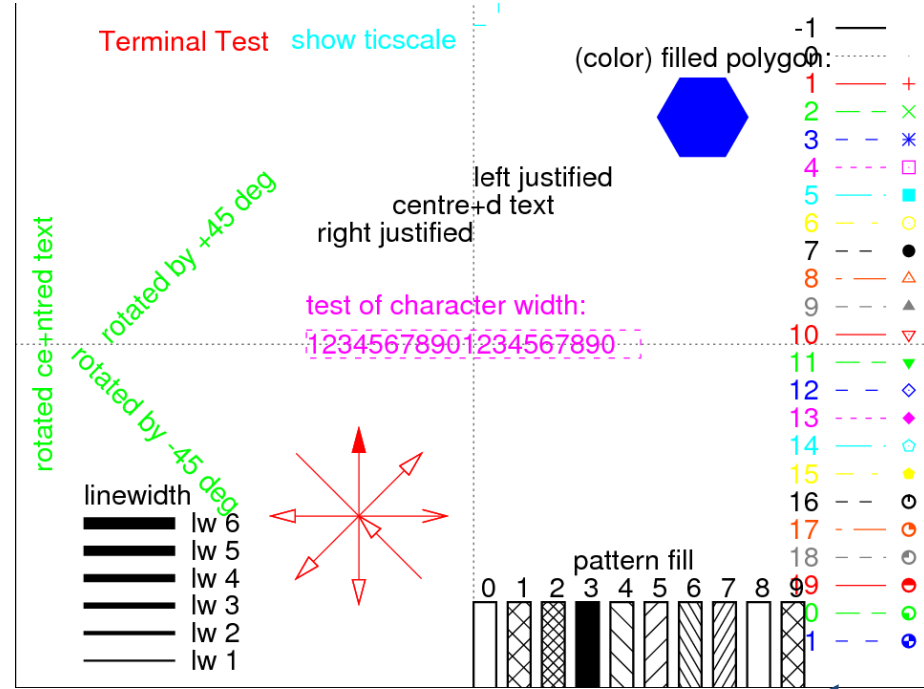
- > `set term x11` # výstup je vykreslen do okna
- > `set term wxt` # výstup je vykreslen do okna (lepší vlastnosti)
- > `set term png size 800,600`  
# výstup je vykreslen jako obrázek ve formátu png
- > `set output "output.png"` # výstup bude uložen do souboru output.png
- > `test` # vytiskne stránku demonstrující vlastnosti terminálu (ne všechny terminály mají stejné možnosti výstupu)

# Ukázky výstupu z různých terminálů

wxt



postscript/eps



podporuje přerušované čáry

# Cvičení

1. Jaké vlastnosti poskytují terminály `x11` a `wxt`. Pracujte v interaktivním režimu a použijte příkaz **test**.
2. Napište skript, který znázorní průběh funkce  $y=5x^3 + 6x^2 - 7$  v rozsahu -10 až 5 pro `x`-ovou hodnotu. Skript spusťte přímo s uvedením interpretru v záhlaví skriptu.
3. Upravte předchozí skript tak, že se graf vykreslí do obrázku ve formátu `png`. Obrázek bude mít rozměry 640x480. Obrázek zobrazte pomocí příkazu **display**.
4. Zobrazte výsledek příkazu **test** pro terminál `png` a `postscript`.
5. Jaké terminály podporuje `gnuplot` (set terminal bez argumentu)?



# Příkaz - splot

K zobrazování funkcí dvou proměnných lze použít příkaz `splot`.

```
> splot funkce/soubor [nastaveni_zobrazeni] [, fce/soubor ...]
```

Zobrazí **XYZ** graf funkce nebo datové řady obsažené v souboru.

Směr pohledu se nastavuje příkazem `set view a,b`, kde **a** a **b** jsou směrové úhly. Pohled shora lze nastavit pomocí `set view map`

Při zobrazování funkcí lze hustotu vzorkování pro x-ový a y-ový směr zadat příkazem `set isosamples a,b`, kde **a** a **b** udává počet vzorků v daném směru.

Pro zvýraznění plochy pomocí funkční hodnoty lze použít `pm3d` zobrazení, např.

```
> splot x*x+y*y with pm3d
```

# Cvičení

1. Zobrazte funkci  $x^2+y^2$
2. Nastavte pohled shora (**set view**)
3. Zrušte pohled shora (**unset view**)
4. Zvyšte hustotu bodů pro zobrazení funkce (**set isosamples**). Použijte hodnoty 10,20 ; 20,10 a 20,20
5. Použijte zobrazení **pm3d**
6. Nastavte pohled shora (**set view**)

Úlohy řešte v interaktivním režimu.



# Bash

---

- Přesměrování vstupu ze skriptu

# Přesměrování vstupu ze skriptu

Přesměrování standardního vstupu programu `my_command` ze souboru skriptu.

```
.....  
./my_command << EOF  
první radka textu  
druha radka textu  
treti radka textu  
EOF  
.....
```

značka určující konec vstupu  
(volí uživatel)

text, který tvoří načítaný vstup

konec vstupu, značku *nesmí*  
*obklopovat mezery*

Tento způsob přesměrování je obzvláště výhodné používat ve skriptech, nicméně funguje i v příkazové řádce. Výhodou je expanze proměnných v načítaném textu.

# Ukázky

```
#!/bin/bash

for ((I=1; $I<=10; I++)); do
    NAME=`printf "%02d.txt" $I`
    cat << EOF > $NAME
    Toto je soubor cislo: $I
EOF
done
```

Výsledek příkazů uvozených zpětnými uvozovkami `` je uložen do proměnné NAME.

Vyznačený text je poslán do **standardního vstupu** příkazu cat, proměnné jsou expandovány před odesláním vstupu, příkaz cat jej pak uloží do souboru \$NAME.

```
#!/bin/bash

gnuplot << EOF
plot sin(x)
EOF
```

Uvedeným způsobem lze programově vytvářet skripty pro gnuplot.

# Cvičení

1. Vytvořte skript, který vytvoří 360 obrázků o rozměrech 800x600 zobrazující průběh funkce  $\sin(x+\text{offset})$ , pro  $x$  v intervalu  $0 - 2\pi$ , kde konstanta  $\text{offset}$  se bude měnit mezi obrázky postupně od 1 do  $360^\circ$ .
2. Vytvořte skript, který vytvoří deset souborů. Jméno souboru bude ve formátu `XX.txt`, kde `XX` je číslo souboru. Pokud je číslo souboru menší než deset, tak jako první cifru v názvu použijte znak 0. Každý soubor bude obsahovat následující text (`X` je číslo souboru):

```
Automaticky vytvoreny textovy soubor  
Cislo souboru je:  X
```