

C2110 *Operační systém UNIX a základy programování*

9. lekce

Petr Kulhánek, Jakub Štěpán

kulhanek@chemi.muni.cz

Národní centrum pro výzkum biomolekul, Přírodovědecká fakulta
Masarykova univerzita, Kotlářská 2, CZ-61137 Brno



INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Obsah

➤ Bash

- cyklus pomoci while

➤ Gnuplot

- přehled jazyka, příkaz plot, terminály, příkaz splot

Bash

➤ **cyklus pomoci while**

Cyklus pomocí while

Cyklus (smyčka) je řídicí struktura, která opakovaně provádí posloupnost příkazů. Opakování i ukončení cyklu je řízeno podmínkou.

cyklus probíhá **dokud** prikaz1 vrací v návratové hodnotě 0

```
while prikaz1
do
    prikaz2
    ...
done
```

Kompaktní zápis:

```
while prikaz1; do
    prikaz2
    ...
done
```

Cyklus pomocí for versus while

```
for((I=1;$I <= 10;I++)); do  
    echo $I  
done
```

provede se před spuštěním cyklu
(inicializace počítadla)

pokud je podmínka splněna, vykonají se
příkazy v bloku do/done

```
I=1  
while test $I -le 10; do  
    echo $I  
    I=`expr $I + 1`  
done
```

aktualizace počítadla po
vykonání příkazů

Přesměrování a roury

Čtení souboru po řádcích:

```
cat soubor.txt | while read A; do
    prikaz2
    ...
done
```

roura

```
while read A; do
    prikaz2
    ...
done < soubor.txt
```

přesměrování

Přesměrování do souboru:


```
for ((I=1;I <= 10;I++)); do
    echo $I
done > soubor.txt
```

Výstup všech příkazů v cyklu je přesměrován do **soubor.txt**.

Domácí úkol I


Vysvětlete rozdílné chování následujících skriptů. Soubor data.txt obsahuje pět řádků.

```
#!/bin/bash
I=0
cat data.txt | while read A; do
    I=$((I+1))
done
echo $I
```



vypíše číslo 0

```
#!/bin/bash
I=0
while read A; do
    I=$((I+1))
done < data.txt
echo $I
```




vypíše číslo 5

Domácí úkol II

Soubor rst.out (wolf.ncbr.muni.cz:/home/kulhanek/Data/rst.out) obsahuje výsledky z molekulové dynamiky. Úkolem je ze souboru vyextrahovat závislost teploty simulovaného systému na čase.

```
.....
NSTEP =      500    TIME (PS) =      0.500    TEMP (K) =      288.02    PRESS =      0.0
Etot   =      942.6248    EKtot   =      151.0990    EPtrot   =      791.5258
BOND   =      51.3204    ANGLE  =      292.3619    DIHED    =      176.5980
1-4 NB =      17.7099    1-4 EEL =      981.4071    VDWAALS  =      -68.3301
EELEC  =     -494.7423    EGB    =     -164.7991    RESTRAINT =      0.1822
EAMBER (non-restraint) =      791.3436
.....
```

čas teplota



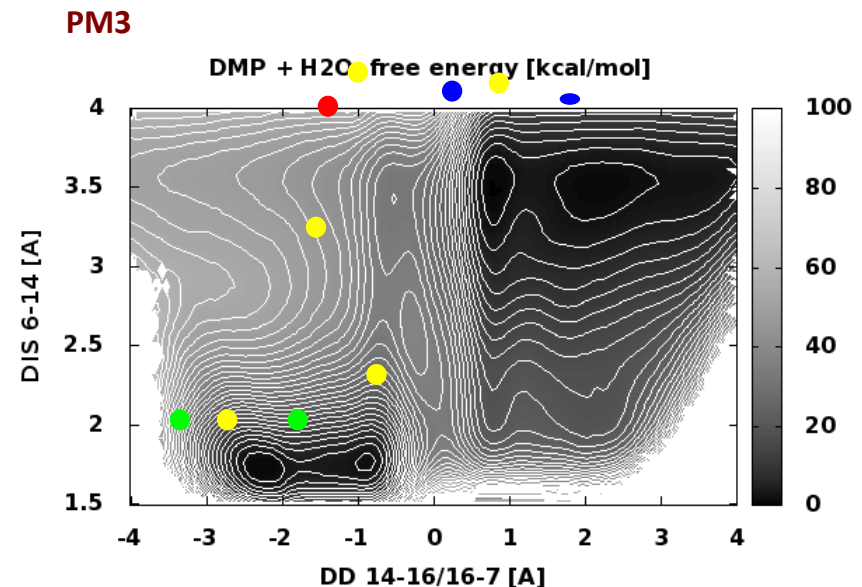
POZOR: Skript nesmí obsahovat příkazy grep, awk a ani jejich varianty. Při řešení použijte příkaz read a while.

Gnuplot

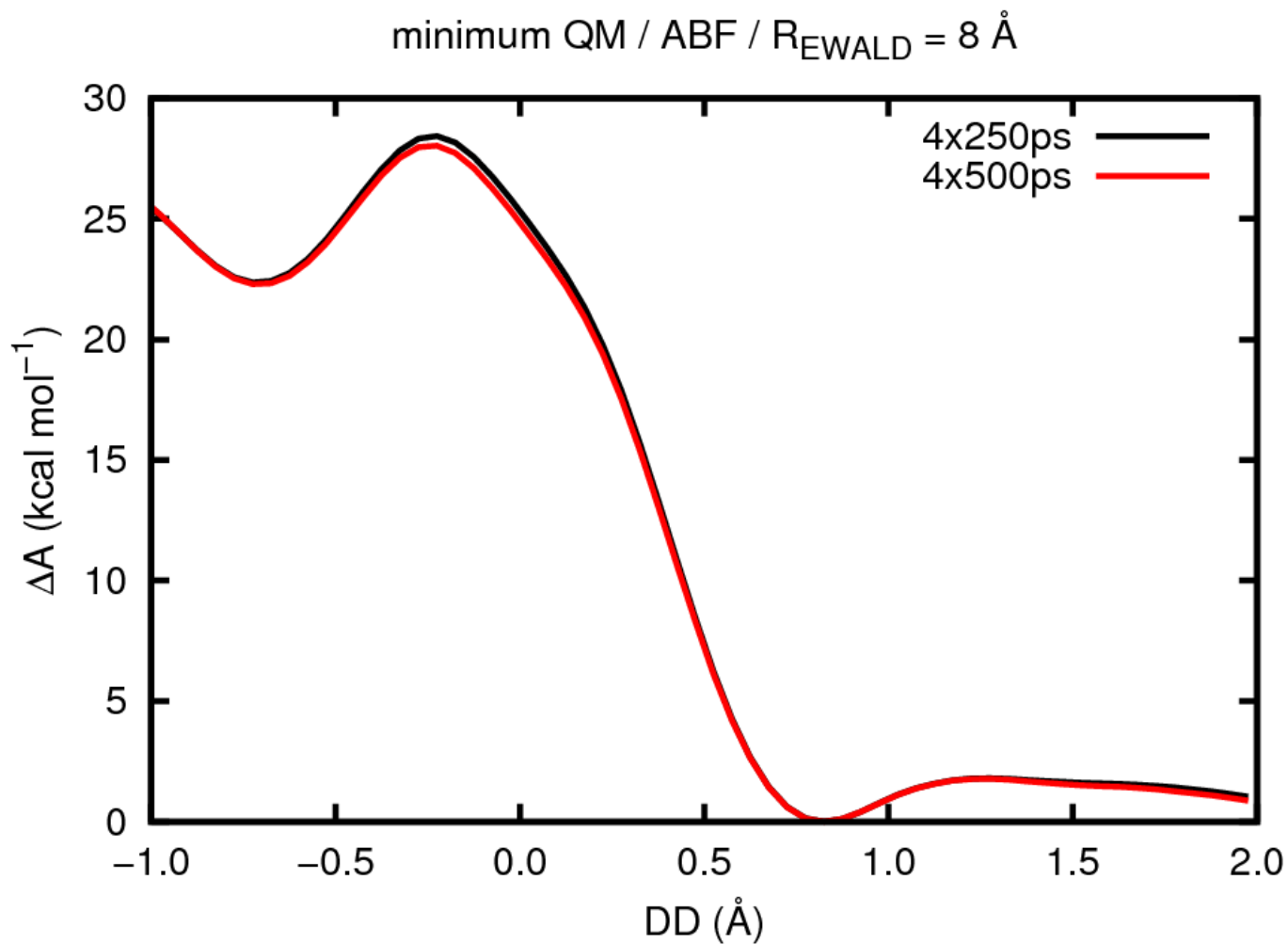
- (Ne)Interaktivní spouštění
- Příkaz plot
- Terminály
- Ukázky

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

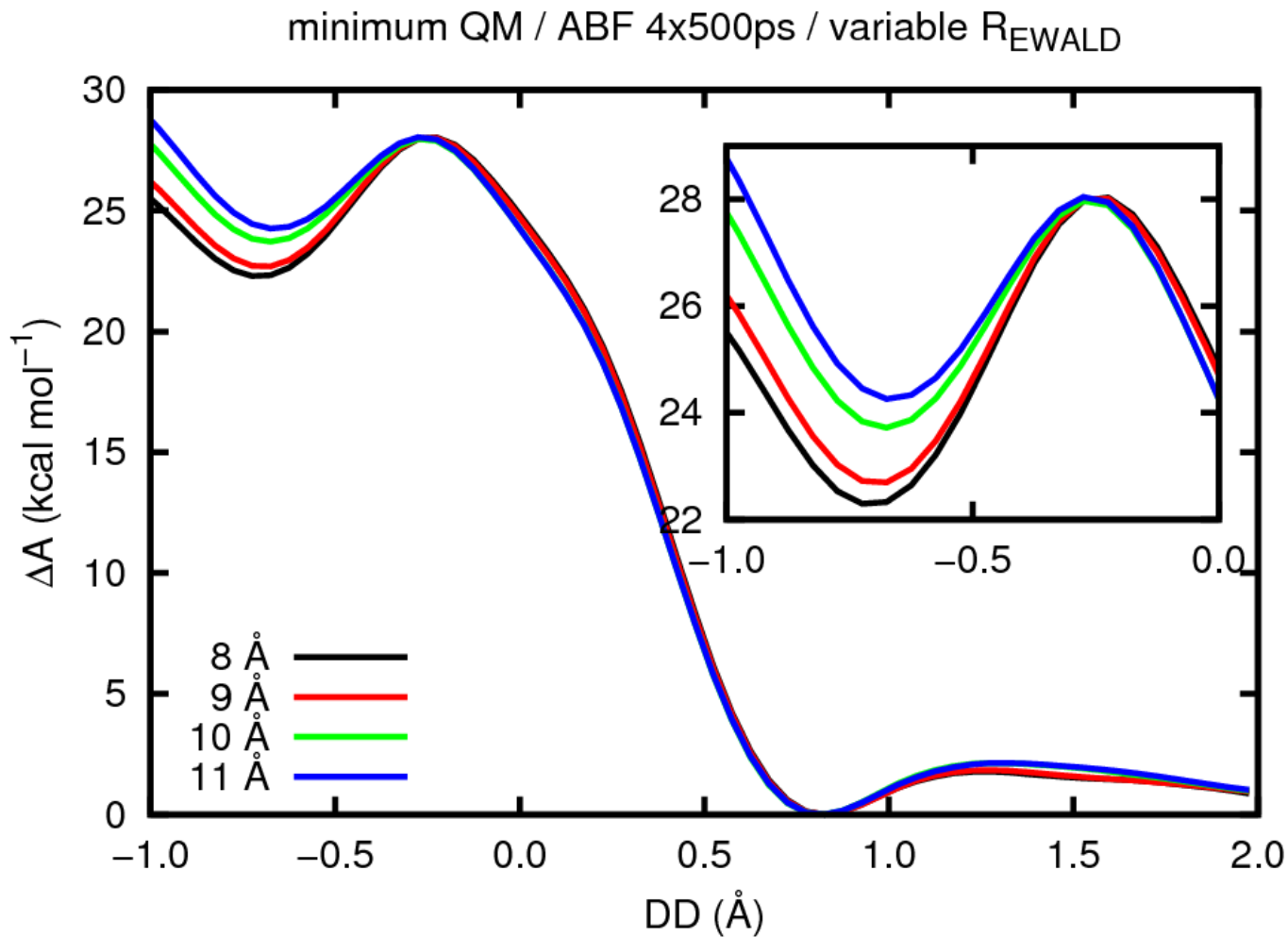
(dokumentace, tutoriály, zdrojové kódy)



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

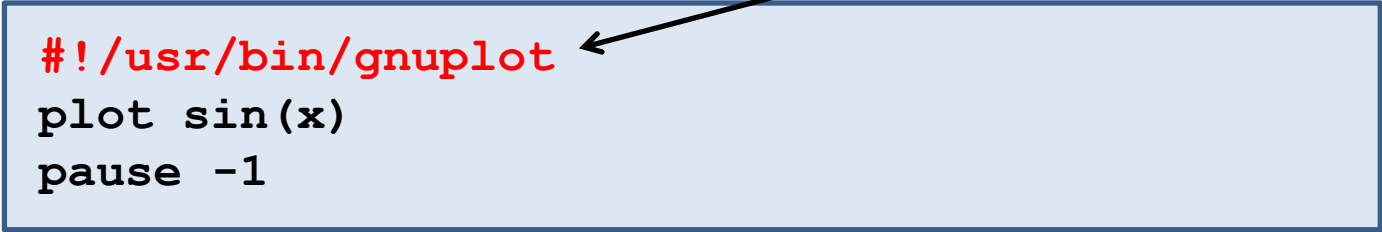
Spouštíme přímo skript (shell automaticky spustí interpreter).

```
$ 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.

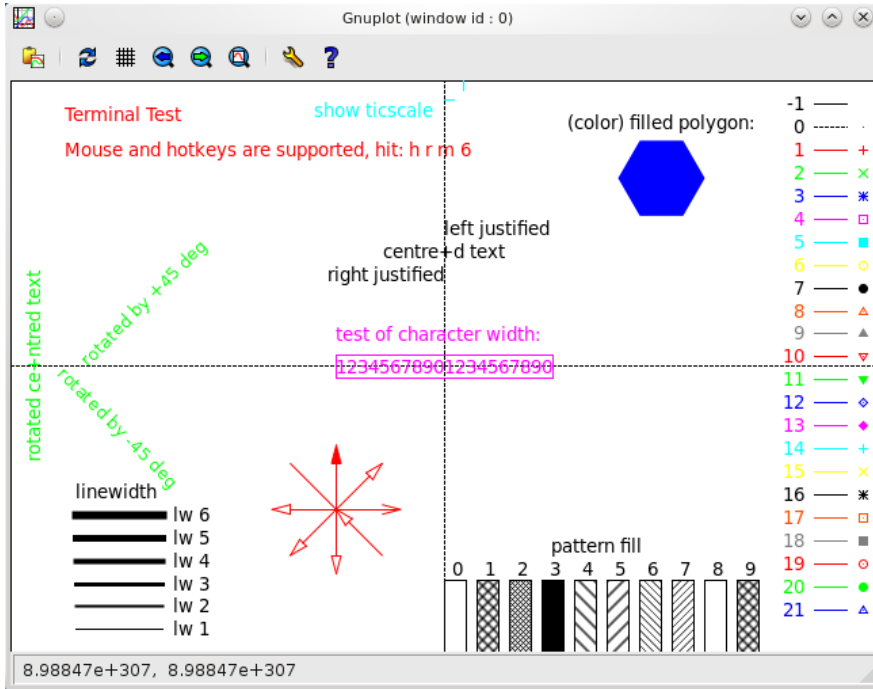
Terminály

Terminál určuje kam bude graf vykreslen.

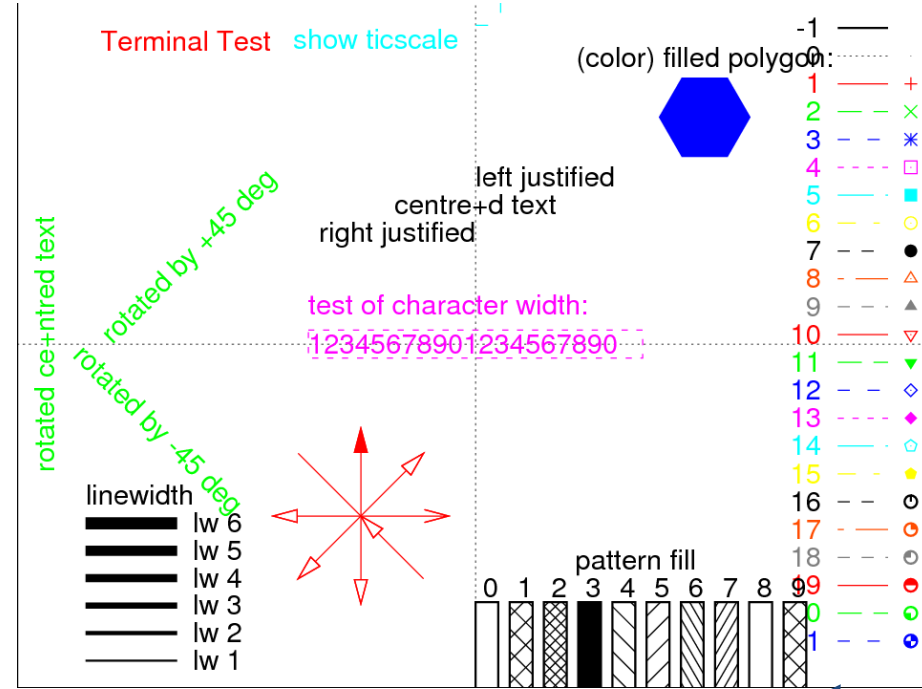
- > `set term x11` # výstup je vykreslen do okna
- > `set term wxt` # výstup je vykreslen do okna (lepší vlastnosti)
- > `set term png size 800,600`
výstup je vykreslen jako obrázek ve formátu png
- > `set output "output.png"` # výstup bude uložen do souboru output.png
- > `test` # vytiskne stránku demonstrující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