

Poznámky k předmětu: Práce s počítačem a programování

11. ledna 2011

Gnuplot, Linux, Pascal

Pavel Srb
Katedra fyziky nízkých teplot, Troja
budova C (kryopavilon)
1.patro, dveře 132
telefon: 22191 2887
email: pavel.srb@matfyz.cz

Úvod

Tento text slouží jako informativní přehled dovedností probíraných v rámci předmětu NAFY008 v zimním semestru ak. roku 2010/2011. Z těchto dovedností budou zároveň vycházet úlohy zadávané za účelem získání zkoušky.

1 Gnuplot

1.1 Základní informace

Instalační balíky pro windows i linux, stejně jako manuál a příklady použití jsou k nalezení na adrese gnuplot.info.

1.2 Použití

Doporučeným způsobem použití gnuplotu je tzv. dávkový mód, kdy v nějakém oblíbeném textovém editoru vytvoříme skript s příkazy, které chceme provést, a tento pak zavoláme buď z gnuplotu příkazem `load "název_souboru"`, nebo z příkazové řádky pomocí `gnuplot nazev_souboru`. V interaktivním režimu naopak můžeme spustět příkazy rovnou.

1.3 Nejdůležitější příkazy:

- `plot`:

```
plot sin(x)
plot "data.dat" using 1:3 with lines lw 2 title "brand new curve"
```

- `set` se používá k nastavení všemožných vlastností objektů např.:

```
set out "vystup.ps"
set xlabel "delka [m]"
```

- `help` se hodí velmi často, například `help set` prozradí jaké všechny parametry lze nastavit.
- `fit` provede proložení dat zadanou funkcí kterou je třeba předtím definovat, stejně jako počáteční hodnoty fitovaných parametrů.

```
fit f(x) "soubor.dat" u 1:3 via par1,par2
```

1.4 příklady skriptů

- vykreslení dat ze souboru

```
set terminal postscript eps color enhanced
set output 'vystupni_obrazek.ps'
set xrange [zleva:doprava]
set yrange [zdola:nahoru]
set xtics 5 out
set mxtics
set grid
set ylabel "popisek osy y"
plot "soubor.sdaty" u 1:2 w l lw 2 lc rgb "red" title "nazev krivky"
```

- fit funkcí zadaného tvaru

```
set terminal postscript eps color enhanced
set output 'vystupni_obrazek.ps'
f(x)= a*exp((x-b)**2/c)
a=100; b=5; c=115;
fit f(x) "data.dat" using 1:3 via a,b,c
plot "data.dat" u 1:3 w points ps 5 ,\
    "a*exp((x-b)**2)/c" title "fit funkci f"
```

2 Jak přežít s příkazovou řádkou operačního systému Linux

2.1 Vzdálené přihlášení

Zásadní výhodou systémů založených na Linuxu je možnost ovládat je ze vzdáleného počítače. K přihlášení většinou používáme protokol *ssh*. Pod Windows je například zdarma k dispozici program *putty*, který umožňuje přihlašovat se ke vzdáleným počítačům. Dále potřebujeme už jen *uživatelské jméno* (*login*) a *heslo*. Po přihlášení se pravděpodobně ocitneme v adresáři, který má *uživatel* nastaven jako domovský adresář (*home*). Takzvaná absolutní cesta k tomuto místu v souborovém systému je typicky */home/uzivatelske_jmeno*. V domovském adresáři může uživatel vytvářet libovolnou adresářovou strukturu včetně souborů. V jiných místech souborového systému k tomuto většinou nemá oprávnění (viz. příkaz *ls -la* níže).

2.2 Několik opravdu základních příkazů

- *ls* vypíše obsah aktuálního adresáře, *ls /cesta/k/nejakemu/adresari*¹ vypíše obsah tohoto adresáře, velmi užitečná varianta *ls -la* (často funguje i *ll*) poskytuje výpis všech souborů včetně skrytých a navíc k nim ještě soupis uživatelských oprávnění k danému souboru.

Základní uživatelská oprávnění jsou *r* (read), *w* (write), *x* (execute). Jejich existence se značí odpovídajícím písmenem, zatímco neexistence mezerou na odpovídajícím místě. Pro každý soubor rozeznáváme tři množiny uživatelů: *user*, *group* a *others*. Zápis práv pro soubor, ke kterému má uživatel všechna práva, uživatelé ze stejné skupiny ho mohou pouze číst a spustit a ostatní pouze číst by vypadal následovně:

```
rwxr xr fajlik.txt
```

- *pwd* vypíše úplnou cestu k aktuálnímu adresáři
- *cd* slouží ke změně adresáře, samotné nastaví uživatelův domovský adresář, doplněno cestou nastaví tuto cestu, např. *cd archiv/leden2010/tajne*. Pozor! poslední použitá cesta je tzv. relativní, to jest odkazuje na adresářovou strukturu relativně vůči aktuální pozici. Absolutní cesta pro stejné umístění by mohla být například */home/strejda/archiv/leden2010/tajne*².
- *chmod* slouží ke změně uživatelských práv. Jedna z možností použití je: *chmod* volba množiny uživatelů *u*, *g*, nebo *o*, pak + nebo - a pak dané právo *r*, *w*, nebo *x*. Například

```
chmod g+w fajlik.txt
```

Umožní všem uživatelům ze skupiny do souboru zapisovat.

- *cat soubor* vypíše soubor na standartní výstup (obrazovka)
- *cat soubor > vystupni.txt* umožňuje přesměrovat výstup z obrazovky do souboru

¹Všimněte si, že zde (na rozdíl od Windows) figuruje obyčejné, nikoliv zpětné lomítko.

²Pro tajný adresář je vhodné, aby žádný jiný uživatel než my sami neměl ani právo číst soubor

- *less* podobně jako *cat* slouží k výpisu souboru ale je interaktivnější - například umožňuje se po textu pohybovat pomocí kurzorů, vypne se pomocí *q*
- *mv starejmeno novejmeno* slouží k přejmenování
- *cp original kopie* slouží ke kopírování
- *mkdir jméno* vytvoří nový podadresář v aktuálním adresáři.
- *rm soubor* smaže soubor
- *man příkaz* poskytne manuálovou stránku daného příkazu, kde jsou nalezeny jeho všechny možné parametry. Často velmi užitečné!!!

2.3 Textový editor vi

Je velice rozšířený textový editor. Hlavním důvody jeho oblíbenosti i přes poměrně vysokou vstupní bariéru pro nového uživatele je jednak možnost editovat soubory na vzdáleném počítači bez grafického rozhraní (například právě při připojení přes *putty*) a druhak jeho nástroje umožňující provádět relativně snadno sofistikované úpravy klidně i mnoha souborů najednou. Z příkazové řádky spustíme *vi*. Klávesou *i* se dostaneme do vkládacího módu (*insert*), který umožňuje psát text. Klávesou *Esc* se vrátíme do původního tzv. normálního módu, zde lze například smazat celou řádku pomocí *dd*, smazat jeden znak pomocí *d*, nahradit znak pomocí *r*. Z normálního módu se přes dvojtečku dostaneme do příkazového módu, kde pomocí *:w jméno_souboru* uložíme soubor, *:wq* uložíme soubor a opustíme *vi*.

2.4 K procvičení

- vytvořte adresář s několika podadresáři a do jednoho z nich zkopírujte soubor *.bashrc* z vašeho domovského adresáře, přejmenujte jej a přidejte do něj na konec libovolný text.
- spojte dohromady soubory *boring.gvr* a *escape.gvr* v adresáři *Gvr* a vytvořte soubor *spojeno.gvr* ve svém domovském adresáři.
- soubor */etc/passwd* obsahuje údaje o uživateli daného systému, zkopírujte soubor do svého domovského adresáře a zjistěte jaké *uid* má uživatel *student*.
- zjistěte jak se kopírují adresáře i s obsahem.
- zjistěte jak lze smazat neprázdné adresáře.
- vytvořte vlastní skript s názvem *lb*, který vypíše obsah souboru *.bashrc* na obrazovku tak, že odpovídající příkaz zapíšete do textového souboru na druhou řádku (na první řádce souboru musí být *#!/bin/sh*). Tento soubor učíte spustitelný pro všechny uživatele.

3 Prvních několik kroků při programování v jazyce Pascal

Jazyk Pascal byl vytvořen počátkem sedmdesátých let 20. století profesorem Niklausem Wirthem za účelem výuky programování. Současná verze je k dispozici zdarma na adrese www.freepascal.org.

3.1 Program hallo

```
program helloworld;
Begin
  write("Hello, world!");
end.
```

3.2 Deklarace proměnných

Úvodní povinná sekce programu uvedená klíčovým slovem `var`.

```
var a:byte; {přirozené číslo 0..255}

    i:integer; {celé číslo zhruba -30000 až 30000}
    r:real;
    ch: char; {znak}
    s:string; {řetězec}
```

3.3 Načtení hodnoty do proměnné a výpis

```
var a:integer;
.
.
write("zadej hodnotu promenne a: ");
read(a);
.
write("Zadana byla hodnota", a);
.
```

3.4 Operace přiřazení

```
var a,b:real;
.
.
a:=12.5;
b:=10*exp(a/98);
...
```

3.5 Cyklus for, while, repeat

```
...
for i:=1 to Horni_mez do
  begin
    ....
  end;
.
.
while a>2 do
  begin
    ...
  end;
.a
..
repeat
  ...
  ...
until Dokud_podminka_neni_splnena;
```

3.6 Větvení programu

```
if podmínka then příkaz;
```

Případně lze použít složitější konstrukci

```
if podmínka then příkaz
.
else if jinýpříkaz
.
else úplně_jiný_příkaz;
```

Samotná podmínka může být složená, k její konstrukci využijeme například **not**, **and** nebo **or**.

```
if (a>0) and (b<100) then ...
```

Pozor na správné uzávorkování!

3.7 Práce s textovými soubory

```
uses dos,crt;
var t:text;
.
.
assign(f,'C:/cesta/jmeno.txt'); {přiřazení jména souboru}
```

```

rewrite(f); {otevře soubor pro zápis}
write(f, "ted pisu do souboru");
close(f); {na konci je třeba soubor zavřít}
.
.
reset(f); {otevře soubor pro zápis}
  while not eof(f) do {funkce eof (End Of File) kontroluje zda nejsme na konci souboru}
    begin
      readln(f,s);
      writeln(s);
    end;
close(f);
.
.

```

3.8 Příklady k procvičení

- napište program který zjistí, zda zadané číslo je dělitelné některým z čísel 2 až 20.
- do textového souboru vypište argumenty a funkční hodnoty polynomu třetího stupně v intervalu 1 až 100 s krokem 0.2, přičemž koeficienty polynomu budou vstupem programu.
- napište program na převod čísel z desítkové do trojkové soustavy. Jak udělat převod do soustavy šestnáctkové?
- napište program na ciferný součet zadaného čísla
- napište program, který vypíše na obrazovku čtverec pomocí znaku hvězdička

Nalezené chyby a nedostatky reportuje prosím na mou emailovou adresu. Děkuji P.S.