

Be in shell

Big exercise, GNUplot

Pavel Fibich

pavel.fibich@prf.jcu.cz
dep. Botany - Na Zlaté stoce 1

04-2016



Přírodovědecká
fakulta
Faculty
of Science

big exercise 1

Exercise

Write command or script that

- print the content of variable `PATH` on more lines, change `:` by new line
- counts characters of the current user name and number of user's processes
- for variable `i` having content: some alphas, than some digits and than some alphas, print just last alphas, e.g. `i=abc76hell` print `hell`, `i=hh3five` print `five`
- count sum of all lines of all files in the current directory
- print file names and their owners in the current folder, nothing else
- print only the shortest and the longest file names in the current directory

Exercise

- print all files having 'a' as the second character
- write command that will create directories, sub-directories, sub-sub-dir... according variable MFO, e.g. for MFO=first_second_third, will create three directories first/second/third
- write command that creates MFO variable from the current directory
- create function `mkcd` that creates and goes into directory (1. arg.)
- write script having 3 arguments: in the file (name as the 1. argument), change/substitute string (2. argument) for the string (3. argument)
- write script having 2 arguments: 1. is file name (`f`) and 2. is number (`n`), it creates two files `f_begin` and `f_tail`, that contains first `n` lines and the rest lines of the file `f`, respectively

GNUplot

GNUplot

GNUplot is command-line interactive plotting program available for many OSs, see <http://www.gnuplot.info>. To run and quit

```
$ gnuplot
G N U P L O T
Version 4.6 patchlevel 0      last modified 2012-03-04
...
gnuplot> quit
```

- for help with commands, use `help` command
- commands can be shortened (eg. `p` for `plot`)
- commands are separated by `;`
- comments are done by `#`
- shell commands start with `!` (eg. `! ls`)
- filenames have to be enclosed by quotes
- we can prepare scripts, sequence of gnuplot commands

```
$ gnuplot gpscript # run gnuplot script
...
gnuplot> load 'gpscript' # run script from gnuplot
```

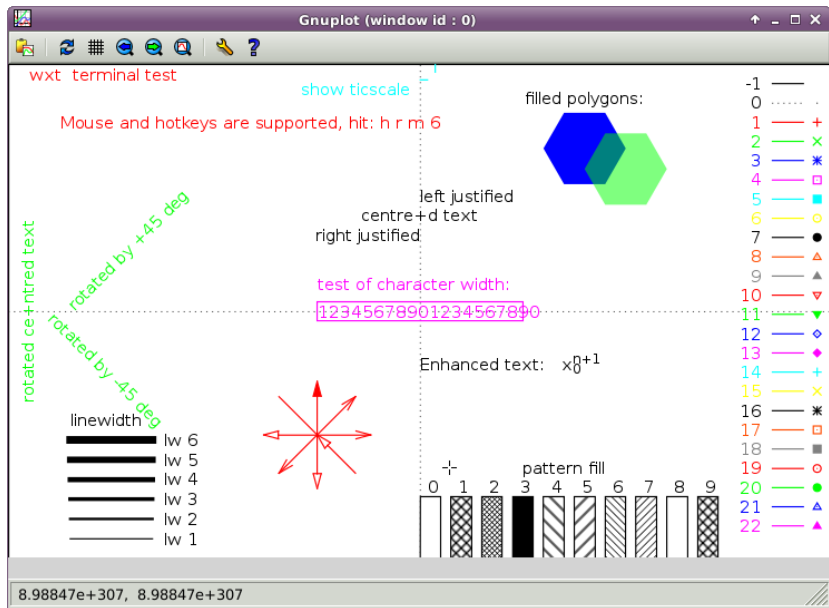
GNUplot output device

```
gnuplot> set terminal # list of output devices
Available terminal types:
    cairolatex  LaTeX picture environment using
                graphicx package and Cairo backend
    canvas      HTML Canvas object
    cgm          Computer Graphics Metafile
...
$ cat gpscript # list of gnuplot script
set terminal png
p sin(x)/x
$ gnuplot gpscript > obr.png # run script and store figure
```

After start, if you see Terminal type set to 'unknown', than install gnuplot-x11 package to see figures directly (not exported into files).

```
$ gnuplot
...
gnuplot> p sin(x) # print graph of sin(x)
gnuplot> test # print test figure for terminal
```

GNUplot test device



GNUplot formatting options

- place or hide key

```
set key top center  
set no key
```

- set a title

```
set title "the title"
```

- axis labels

```
set xlabel "pH"  
set ylabel "a"
```

- plot an arrow `set arrow from 0.5,0 to 0.5,0`
- define a label `set label "b point" at 0.5,0`
- border `set border lw 3`
- ticks `set ytics 0.5, set xtics 0.1`
- linewidth (`lw`), pointsize (`ps`), line or point type `lt`, `pt`

After changing of settings, it is often necessary to run `replot` `rep.`

GNUplot input files

gnuplot can read data from files

```
# which changes are done by following command?
$ sed -e 'ld' -e 's/,/ /g' ideff.csv >ideff2.csv
gnuplot> plot 'ideff2.csv' using 4:8 # 4. and 8. col scatter plot
# short commands with point type
gnuplot> p 'ideff2.csv' u 4:8 pt 8
# multiple data series
gnuplot> p 'ideff2.csv' u 4:7 pt 8, 'ideff2.csv' u 4:6
gnuplot> p 'ideff2.csv' u 4:($7/5) # do math on column
```

Columns are indexed from \$1 and \$0 is for running index.

To change scale, zoom or style

- [un]set logscale [xy]
- set xrange [0:10]; set yrange [*:*] y is set automatically
- multiplot options by set multiplot

You can specify column separator set datafile separator ";"
and also presence of headline set key autotitle columnhead.

Many built-in function for

- math - abs, sin, log, exp, round, floor, pi, ..
- strings - gprintf, sprintf, strlen, substr
- other - system (calling shell command), column (get column), value (value of variable), rand (random value), fitting of functions, reset settings.

and also operators. User can easily prepare own variable or functions

```
gnuplot> a = floor(tan(pi/2 - 0.1))
gnuplot> print a
gnuplot> sinc(x) = sin(pi*x)/(pi*x)
gnuplot> p sinc(x)/x
gnuplot> min(a,b) = (a < b) ? a : b
```

Nice demo files at <http://gnuplot.sourceforge.net/screenshots/index.html#demos>

GNUplot example 1

```
$ wget http://botanika.prf.jcu.cz/fibich/bash/using.dat
$ less using.dat
$ gnuplot
```

We will plot 4-6. column (y-axis) based on 3. column (x-axis) by impulses, points and lines.

```
gnuplot> plot 'using.dat' using 3:4 title "Logged in"
      with impulses,\
'using.dat' using 3:5 t "Load average" with points,\
'using.dat' using 3:6 t "%CPU used" with lines
```

To set title, move the legend and xlab, you must call before the plot

```
gnuplot> set title "Computer load"
gnuplot> set key below
gnuplot> set xlab "Days"
```

To change the color and point size and type use

```
'using.dat' using 3:5 t "Load average" with points pt 3
      ps 4 lc rgb "cyan",
```

Exercise

- write function that will get data for specified experiment from the file `ideff.csv`, argument of the function will be the name of the experiment (1. column)
- plot dependence of 6. col on 5. col (`u 5:6`) for GE1 and GE2 in one figure
- write script (or function) that will have 1 argument (filename), script will run gnuplot, that will print scatter plot of the first and second column into some graphical file (eg. add png suffix for the input filename)
- add title to the figure according to the given filename
- use boxplot to differentiate values of the fifth column of GE1 and GE2

```
gnuplot> set style data boxplot # pre-set the data style
```

- add xlab according the experiment and ylab