


operativni sistem

— part 2 —

© Predrag Pejović, 

help na komandnoj liniji ...

- ▶ `man ls` (manual)
- ▶ `ls --help` (help)
- ▶ `info ls` (GNU info system)
- ▶ `apropos`, `man apropos`
- ▶ `whatis`, `whatis apropos` (vrlo kratak opis)
- ▶ `whereis`, `whereis apropos` (vrlo kratak opis)

file system, nastavak ...

- ▶ bitno različit file system u odnosu na MS-DOS/win
- ▶ ne postoje volumes poput C:\, D:\, ...
- ▶ Ext 3, Ext 4, journal file system
- ▶ case sensitive, kada to može, ne kod FAT; ovo pravi probleme kod kopiranja na USB flash, File i file primer, preklapanje
- ▶ jedinstven file system, stablo počinje u root, /
- ▶ volumes se mount-uju: ubacite USB flash
- ▶ USB flash, CD, u /media/, pogledajte sa `ctrl/1`
- ▶ a sada kroz file system pomoću komandne linije ...
- ▶ terminal na pola, File Manager na pola, ...

redirekcija, cat, tac, head, tail, more, less ...

- ▶ `ls -lht /usr/bin > usrbin1.lst`
- ▶ `ls /usr/bin > usrbin2.lst`
- ▶ `ls -lht > ls.lst`
- ▶ `ls`
- ▶ `cat usrbin1.lst`
- ▶ `tac usrbin1.lst`
- ▶ `head usrbin1.lst`
- ▶ `tail usrbin2.lst`
- ▶ `cat usrbin1.lst usrbin2.lst`
- ▶ `cat usrbin1.lst usrbin2.lst > usrbin3.lst`
- ▶ `more usrbin3.lst`
- ▶ `less usrbin3.lst`

▶ update

- ▶ komandna linija, nastavak ... mnoštvo programa (jezik se zaboravlja ako se ne govori)
- ▶ scripting, bash, sh
- ▶ malo uvoda u Python, IDLE, idle
- ▶ malo C
- ▶ (ni)malo IDE, Integrated Development Environment, Code::Blocks, Eclipse (drugi to rade ...)

synaptic

- ▶ još jedan način za instaliranje programa
- ▶ jako dobro, omogućava precizna podešavanja
- ▶ nije instaliran by default ...
- ▶ zahteva neko razumevanje ...
- ▶ pokrenete Synaptic Package Manager
- ▶ dopunite octave, modul octave-control
- ▶ pokrenete `octave`
- ▶ `h = tf([1], [1, 1])`
- ▶ `bode(h)`
- ▶ `h = tf([10], [1, 0, 100])`
- ▶ `bode(h)`
- ▶ `h = tf([10], [1, 0.1, 100])`
- ▶ `bode(h)`

directory system commands:

pwd, mkdir, rmdir, cd ...

- ▶ `pwd`
- ▶ `mkdir <ime>`
- ▶ `ls`
- ▶ `cd <ime>`
- ▶ `pwd`
- ▶ `ls`
- ▶ `cd ..`
- ▶ `cd -`
- ▶ `cd ~`
- ▶ `cd /`
- ▶ `cd ~/<ime>`
- ▶ `cd`
- ▶ `cd <ime>`
- ▶ `pwd`

pipe, grep ...

- ▶ `ls -lht /usr/bin | more`
- ▶ `ls -lht /usr/bin | less`
- ▶ `ls /usr/bin | more`
- ▶ `ls /usr/bin | less`
- ▶ `grep maxima usrbin1.lst`
- ▶ `grep maxima usrbin2.lst`
- ▶ `ls -lht /usr/bin | grep maxima`

grep i regular expressions

- ▶ grep, globally search a regular expression and print
- ▶ regular expression
- ▶ knjiga, 197 strana
- ▶ Google Analytics, Regular Expressions

scripting ...

- ▶ pokrenuti editor, `gedit radi`
- ▶ View, Highlight Mode, sh
↑ podešava context highlighting
- ▶ scripts su obično sa ekstenzijom sh
- ▶ mada ne mora ...
- ▶ videćete šta radi file

script file 1: radi

```
ls -lh /usr/bin > usrbin1.lst
ls /usr/bin > usrbin2.lst
ls -lh > ls.lst
mkdir lst
mv usrbin1.lst lst
mv usrbin2.lst lst
mv ls.lst lst
cd lst
ls -lh
cd ..
```

script file 2: briši

```
gedit brisi.sh
rm -i lst/*.lst
```

startovanje shell file kao data file

- ▶ `file radi`, koji je tip fajla?
- ▶ `ls -lh radi`, biće još reči o ovome
- ▶ `bash radi`
- ▶ pogledajte šta je urađeno!
- ▶ `bash brisi.sh`
- ▶ pogledajte šta je urađeno!
- ▶ `sh radi`
- ▶ pogledajte šta je urađeno!
- ▶ `sh brisi.sh`
- ▶ pogledajte šta je urađeno!

privileges ...

- ▶ `ls -lhtr`
- ▶ i postoji nešto nalik na `-rw-r--r--` pa podaci o fajlu
- ▶ i nešto nalik na `drwxr-xr-x` pa podaci o direktorijumu
- ▶ prvi znak: `-` za fajl ili `d` za direktorijum
- ▶ `r` je read, `w` je write, `x` je execute (ls)
- ▶ tripleti $\underbrace{rwx}_{\text{user}} \underbrace{rwx}_{\text{group}} \underbrace{rwx}_{\text{other}}$
- ▶ svaka privilegija (`r`, `w` ili `x` je jedan bit)
- ▶ triplet daje oktalnu cifru, `rwX` je 7, `r--` je 4, `--x` je 1, ...
- ▶ privilegija se kodira kao trocifreni oktalni broj
- ▶ `rwX-----` je 700
- ▶ `rwXr--r--` je 744
- ▶ `rw-r--r--` je 644

chmod ...

- ▶ `ls -lh radi`
- ▶ `chmod <code> <file_name>`
- ▶ `chmod 700 radi`
- ▶ `ls -lh radi`
- ▶ pozeleneo?
- ▶ drugi način
- ▶ `chmod {u|g|o}{+|-}{r|w|x} <file_name>`
- ▶ `chmod u-w radi`
- ▶ `ls -lh radi`
- ▶ `chmod u+w radi`
- ▶ `ls -lh radi`
- ▶ `chmod go+x radi`
- ▶ `ls -lh radi`
- ▶ `chmod go-r radi`
- ▶ `radi`
- ▶ `./radi`

script file 3

```
gedit argumenti0.sh

#!/bin/bash
echo $0
echo $1
echo $2
echo $3

file argumenti0.sh
sh argumenti0.sh
sh argumenti0.sh a b c
sh argumenti0.sh d e
bash argumenti0.sh
./argumenti0.sh
```

#!

shell? malo istorije (za zainteresovane) ...

- ▶ #!
- ▶ promenite u `argumenti0.sh` prvi red u `#!/bin/sh`
i save kao `argumenti0a.sh`
- ▶ promenite u `argumenti0.sh` prvi red u `#!/usr/bin/python`
i save kao `argumenti0b.sh`
- ▶ obrišete prvi red i save kao `argumenti0c.sh`
- ▶ `file argumenti0.sh`
- ▶ `file argumenti0a.sh`
- ▶ `file argumenti0b.sh`
- ▶ `file argumenti0c.sh`

- ▶ https://en.wikipedia.org/wiki/Unix_shell
- ▶ https://en.wikipedia.org/wiki/Thompson_shell
- ▶ https://en.wikipedia.org/wiki/Bourne_shell
- ▶ https://en.wikipedia.org/wiki/Stephen_R._Bourne
- ▶ `https://en.wikipedia.org/wiki/Bash_%28Unix_shell%29`
- ▶ i jedna knjiga:
<http://sourceforge.net/projects/linuxcommand/>

script file 4

```
gedit argumenti1.sh
```

```
#!/bin/bash
echo 0
echo $0
echo 1
echo $1
echo 2
echo $2
echo 3
echo $3
```

```
file argumenti1.sh
sh argumenti1.sh
./argumenti1.sh
```

script file 5

```
gedit argumenti2.sh
```

```
#!/bin/sh
echo 0: $0
echo 1: $1
echo 2: $2
echo 3: $3
```

```
file argumenti2.sh
sh argumenti2.sh
```

environment variables, PATH

- ▶ `printenv`
- ▶ `printenv | less`
- ▶ `printenv PATH` (prikaže vrednost promenljive PATH)
- ▶ `printenv PATH | grep home`
- ▶ `echo $PATH` (prikaže vrednost promenljive PATH)
- ▶ `$PATH` (shell pokušava da izvrši)

environment variables, PWD

- ▶ `pwd` (print working directory)
- ▶ `printenv PWD` (isto!)
- ▶ `echo $PWD` (isto!)
- ▶ `echo PWD` (ništa korisno)
- ▶ `$PWD` (pokuša da izvrši)
- ▶ `echo $pwd` (ništa)
- ▶ `echo $(pwd)` (ništa)
- ▶ `$(pwd)` (pokuša da izvrši)

proširimo PATH!

- ▶ `echo $PATH`
- ▶ `echo $PATH:$PWD`
- ▶ `PATH=$PATH:$PWD`
- ▶ `printenv PATH`
- ▶ `printenv PATH | grep home`
- ▶ `radi`

čišćenje ... uvek!!!

- ▶ `cd lst`
- ▶ `ls`
- ▶ `ls -1`
- ▶ `rm -i *`
- ▶ `ls`
- ▶ `cd ..`
- ▶ `rmdir lst`

vaš prvi Python program, proba, bez .py ...

- ▶ python, imate li instaliran python koji nije python3?
- ▶ za ovaj primer razlika je samo u ()

```
#!/usr/bin/python3

print()
print('Hello World!')
print()
print('MPFC')
print()
```

vaš drugi Python program, f.py, in gedit ...

```
#!/usr/bin/python3

print()
n = int(input('argument = '))
print()

f = 1
for i in range(n):
    f = f * (i + 1)

print(n, "! =", f)
print()
```

Zadatak: Fibonačijevi brojevi

- ▶ $f_n = f_{n-1} + f_{n-2}$, $f_1 = 1$, $f_2 = 1$
- ▶ lista od prvih n Fibonačijevih brojeva

zadatak, C program, faktorijel

- ▶ Dennis MacAlistair Ritchie (Username: dmr, September 8, 1941 – October 8, 2011)
- ▶ "UNIX is very simple, it just needs a genius to understand its simplicity."
- ▶ "C is quirky, flawed, and an enormous success."
- ▶ zatraži argument sa `argument =`
- ▶ izračuna faktorijel i javi rezultat u formi `m! = n`
- ▶ `gedit faktorijel.c`
- ▶ pišete program ...

pokretanje, ...

- ▶ `python3 proba`
- ▶ `chmod +x proba`
- ▶ `./proba`
- ▶ `chmod -x proba`
- ▶ `./proba`
- ▶ `python3 proba`
- ▶ `cp proba proba.py`
- ▶ `ls -lhtr`
- ▶ `idle` (Not IDE, Eric Idle?)
- ▶ File, Open, proba.py
- ▶ context highlighting
- ▶ Run, Check Module (Alt+X)
- ▶ Run, F5

pokretanje ...

- ▶ da probamo sva tri nacina, sa i bez .py?
- ▶ `ipython3`
- ▶ `run f.py`
- ▶ `ctrl/d` ili `^d`
- ▶ vrednosti promenljivih ostaju u interpreteru
- ▶ nema problema sa velikim brojevima, dynamic typing
- ▶ a kako je počelo? ovako:
- ▶ `python3`
- ▶ `execfile('f.py')`
- ▶ `^d`

Fibonačijevi brojevi, Python

```
#!/usr/bin/python3

n = int(input('duzina liste (int, >2): '))

print()

a = b = 1

print(16 * '-')

i = 1
print(repr(i).rjust(3) + ' ' + repr(a).rjust(10))
i = 2
print(repr(i).rjust(3) + ' ' + repr(b).rjust(10))

for i in range(2,n):
    a, b = b, a + b
    print(repr(i+1).rjust(3) + ' ' + repr(b).rjust(10))

print(16 * '-')
```

jedan C program ...

```
#include <stdio.h>

int main()
{
    int i, n, f;

    printf("\nargument = ");
    scanf("%d", &n);

    f = 1;
    for (i=1; i<=n; i++) {f=f*i;}

    printf("\n%d! = %d\n", n, f);

    return 0;
}
```

- ▶ `gcc -o faktorijel faktorijel.c`
- ▶ `ls -lh`
- ▶ zeleni `faktorijel`?
- ▶ `./faktorijel`
- ▶ problemi sa velikim brojevima?

```
#include <stdio.h>

int main(void)
{
    printf("\nn = ");
    int n;
    scanf("%d", &n);

    int i = 1, j = 1, k, c;
    printf("\n\n1\n1\n");

    for (c = 2; c <= n; c++)
    {
        k = i + j;
        i = j;
        j = k;
        printf("%d\n", k);
    }

    return 0;
}
```

korak napred, IDE, Code::Blocks

- ▶ IDE, Integrated Development Environment
- ▶ Eclipse, Code::Blocks i Geany popularni ... ovde C::B
- ▶ gdb, command line, sami
- ▶ kreirate projekat, Empty
- ▶ Debug/Release version
- ▶ Build
- ▶ watches
- ▶ breakpoints
- ▶ start, step into, shift-F7
- ▶ izvršavanje korak po korak, F7
- ▶ na kraju build Release version
- ▶ F7 ili button panel
- ▶ watch Local variables
- ▶ just to get acquainted ...