
Operativni sistemi

- Rad u komandnoj liniji -

Veljko Stanković

- Standardni unos komandi u računarski sistem se vrši uz pomoć komandnog interpretera.
 - Koristi se kada grafički interfejs ne funkcioniše ili pri administraciji sistema.
- Komnadni interpereter pod Linuxom se naziva shell.
 - Postoji više implementacija sa sličnim komandama i funkcionalnošću.
 - Najčešće korišćeni je bash shell.

Pokretanje terminala i virtuelne konzole

1010011
1110100
1100001
1010110

- Najjednostavniji način da se pristupi komandnom interpreteru jeste otvaranjem terminala uz pomoć GUI.
- Drugi način da se pristupi komandnom interpreteru jeste otvaranjem virtuelne konzole.
 - Ukoliko nije moguće pristupiti uz pomoć GUI.
 - Fizička konzola predstavlja kombinaciju monitor-tastatura.
 - Virtuelne konzole daju mogućnost da koristimo više konzola za unos komandi iako imamo na raspolaganju samo jednu fizičku konzolu.
 - Virtuelnoj konzoli se pristupa kombinacijom Ctrl+Alt+Fx
 - ☑ Možemo otvoriti do 6 virtuelnih konzola.
 - ☑ Povratak na GUI se vrši kombinacijom Ctrl+Alt+F7.

— Sintaksa shell komandi

- Sve shell komande imaju opšti oblik koji se sastoji iz komande koju prate opcije:

`command option1 option2 option3 ... optionN`

- Komandan linija predstavlja zapis komande na ekranu.
- Na komandnoj liniji se zadaje komanda sa nula ili više opcija (argumenata).
 - ☑ Niz opcija (argumenata) na komandnoj liniji modifikuje način na koji se komanda izvršava.
 - ☑ Komanda i opcije se razlikuju tako što su odvojene (space, tab)
 - ☑ Opcija može sadržati prazan prostor pod uslovom da se opcija nalazi u okviru znakova navoda “”.
 - ☑ Broj i format opcija zavisi od konkretne komande.

— Sintaksa shell komandi

- Ukoliko je dužina komande veća od jedne linije, tada se unos može nastaviti tako što bi na kraju linije stavili simbol \.
 - ☑ Primer: Predstavljanje sadržaja direktorijuma /etc/passwd
 - cat \
 - /etc/passwd
- Više komandi se može nadovezati tako što različite komande razdvajamo uz pomoć ;.
 - ☑ Primer: cd; ls -l; pwd
- Jednostavne shell komande se mogu kombinovati kako bi dobili složenije komande.

— Sintaksa shell komandi

- ↪ Jednставne shell komande se mogu kombinovati kako bi dobili složenije komande.

☑ Primer: Želimo da utvrdimo da li se u direktorijumu /dev nalazi mount tačka uređaja sbpcd. To možemo postići kombinacijom naredbi ls i grep.

```
ls /dev | grep sbpcd
```

- ↪ Vertikalna linija | se zove “cev” (pipe) jer igra ulogu veze između dva programa. Izlaz prve komande predstavlja ulaz druge komande.

— Kontrola ulaza i izlaza komande

- Većina komandi Linux-a ima zajedničku osobinu da podatke uzima sa standardnog ulaza – tastature, i rezultate predstavljana standardnom izlazu – ekranu. Poruke o greškama se šalju najčešće na ekran.
- Ova tri standardna uređaja se obično označavaju sa stdin, stdout i stderr.
- I/O redirekcija
 - ☑ Moguće je da komanda preuzme ulazne podatke iz fajla i da rezultate takođe pošalje u fajl.

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Kontrola ulaza i izlaza komande

<i>Task</i>	<i>Command Syntax</i>
Send stdout to a file	<code>command > file</code>
Send stderr to file	<code>command 2> file</code>
Send stdout and stderr to file	<code>command > file 2>&1</code>
Read stdin from a file	<code>command < file</code>
Read stdin from file.in and send stdout to file.out	<code>command < file.in > file.out</code>
Append stdout to the end of a file	<code>command >> file</code>
Append stderr to the end of a file	<code>command 2>> file</code>
Append stdout and stderr to the end of a file	<code>command >> file 2>&1</code>
Read stdin from the keyboard until the character <code>c</code>	<code>command <<c</code>
Pipe stdout to command2	<code>command command2</code>
Pipe stdout and stderr to command2	<code>command 2>&1 command2</code>

— Kontrola ulaza i izlaza komande

✚ Unos ulaznih podataka komande iz fajla

- ☑ Primer: Komanda sort uzima ulaz iz fajla /etc/passwd

```
sort < /etc/passwd
```

- ☑ Kao rezultat dobija se sortirana lista linija fajl /etc/passwd.

✚ Skladištenje izlaza komande u fajl

- ☑ Primer: Pretražujemo sadržaj fajlova u direktorijumu /usr/include/ koji sadrže typedef i rezultat smešta u fajl typedef.out:

```
grep typedef /usr/include/* > typedef.out
```

— Kontrola ulaza i izlaza komande

- Ako želimo da dodamo izlaz komande na kraj postojećeg fajla, koristimo simbol >> na sledeći način:

```
command >> filename
```

- Primer: Kako kreirati tekst fajl i uneti tekst takoda se unos teksta završi kada otkucamo kraj i pritisnemo Enter.

```
cat << kraj > input.txt
```

— Kontrola ulaza i izlaza komande

↪ Skladištenje poruke o grešci u fajl

- ☑ U slučajevima kada je poruka o grešci koja se javila pri izvršavanju komande izuzetno dugačka, korisno je da izlaz snimimo u fajl kako bi mogli kasnije da pretražimo i utvrdimo razlog pojave greške.
- ☑ Primer: Pretražujemo sistem za fajl pod imenom COPYING ali sve poruke o grešci se ispisuju u fajl finderr:

```
find / -name COPYING -print 2> finderr
```

- ☑ Ako želimo da odbacimo poruke o grešci, tada se sve poruke salju u specijalni fajl /dev/null (bit bucket, korpa za otpatke):

```
find / -name COPYING -print 2> /dev/null
```

— Kontrola ulaza i izlaza komande

- Linux shell omogućava automatsko dopunjavanje imena fajlova i direktorijuma pritiskom na Tab nakon određenog broja unetih karaktera.
- Ispisivanje dugih imena se može izbeći na drugi način upotrebom simbola * i ?.
 - ☑ Uz pomoć * možemo zameniti bilo koju grupu simbola proizvoljne duzine.
 - Primer: `rm *` - brise sve fajlove u radnom direktorijumu.
 - ☑ Uz pomoć ? možemo zameniti jedan simbol.
 - Primer: `ls test?` - lista sve fajlove/direktorijume čija su imena dužine 5 karaktera od kojih su prva četiri test.
 - ☑ Grupa karaktera unutar velikih zagrada [] omogućava da zamenimo neki od tih kratkera u imenu fajla ili direktorijuma.
 - Primer: `ls [aB]*` - lista sve fajlove koji počinju sa a ili B.

— Kontrola ulaza i izlaza komande

↪ Kombinacijom (wildcard) simobla se može znatno pojednostaviti pretraga fajlova i direktorijuma.

☑ Primer: Želimoda nadjemo string typedef struct u svim fajlovima direktorijuma /usr/include koji počinju na s i završavaju se sa .h. To možemo postići sa:

```
grep "typedef struct" /usr/include/s*.h
```

☑ Primer: U radnom direktorijumu imamo niz slika sa imenima image1.jpg, image2.jpg, itd. Želimo da ih iskopiramo u drugi direktorijum:

```
cp image?.jpg /media/disk
```

☑ Primer: U direktorijumu /etc/X11/xdm želimo da nadjemo sve fajlove čija iman počinju sa x ili X:

```
ls /etc/X11/xdm/[xX]*
```

— Kontrola ulaza i izlaza komande

- Kako bi olakšao upotrebu dugačkih komandi bash skladišti do 500 komandi unetih preko terminala.
- Komandom history ispisuje se spiak prethodno unetih komandi.
 - ☑ Komanda history N ispisuje samo N prethodno unetih komandi.
- Komandu možemo ponoviti tako što unesemo !Redni_broj_komande, gde je Redni_broj_komande broj pod kojim je komanda zavedena u listi dobijeneoj komandom history.
- Sa !command izvršava se zadnji poziv te komande.
 - ☑ Primer: ls -l ponavljamo sa !ls.
- Zadnju komadnu ponavljamo sa !!.
- Najjednostavniji način pregleda i unosa porethodno unetih komandi je sekvencijalnim pretraživanjem uz pomoć strelica.

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Pregled Linux komandi

Getting Online Help

<code>apropos</code>	Finds online manual pages for a specified keyword
<code>info</code>	Displays online help information about a specified command
<code>man</code>	Displays online help information
<code>whatis</code>	Similar to <code>apropos</code> but searches for complete words only

Making Commands Easier

<code>alias</code>	Defines an abbreviation for a long command
<code>type</code>	Shows the type and location of a command
<code>unalias</code>	Deletes an abbreviation defined using <code>alias</code>

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Pregled Linux komandi

Managing Files and Directories	
<code>cd</code>	Changes the current directory
<code>chmod</code>	Changes file permissions
<code>chown</code>	Changes file owner and group
<code>cp</code>	Copies files
<code>ln</code>	Creates symbolic links to files and directories
<code>ls</code>	Displays the contents of a directory
<code>mkdir</code>	Creates a directory
<code>mv</code>	Renames a file as well as moves a file from one directory to another
<code>rm</code>	Deletes files
<code>rmdir</code>	Deletes directories
<code>pwd</code>	Displays the current directory
<code>touch</code>	Updates a file's time stamp

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Pregled Linux komandi

Finding Files	
<code>find</code>	Finds files based on specified criteria, such as name and size
<code>locate</code>	Finds files using a periodically updated filename database. (The database is created by the <code>updatedb</code> program.)
<code>whereis</code>	Finds files based in the typical directories where <i>executable</i> (also known as <i>binary</i>) files are located
<code>which</code>	Finds files in the directories listed in the PATH environment variable

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Pregled Linux komandi

Processing Files	
cat	Displays a file on standard output (can be used to concatenate several files into one big file)
cut	Extracts specified sections from each line of text in a file
dd	Copies blocks of data from one file to another (used to copy data from devices)
diff	Compares two text files and finds any differences
expand	Converts all tabs into spaces
file	Displays the type of data in a file
fold	Wraps each line of text to fit a specified width
grep	Searches for regular expressions within a text file
less	Displays a text file one page at a time (can go backward, also)
lpr	Prints files
more	Displays a text file, one page at a time (goes forward only)
nl	Numbers all nonblank lines in a text file and prints the lines to standard output

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Pregled Linux komandi

<code>paste</code>	Concatenates corresponding lines from several files
<code>patch</code>	Updates a text file using the differences between the original and revised copy of the file
<code>sed</code>	Copies a file to standard output while applying specified editing commands
<code>sort</code>	Sorts lines in a text file
<code>split</code>	Breaks up a file into several smaller files with specified size
<code>tac</code>	Reverses a file (last line first and so on)
<code>tail</code>	Displays the last few lines of a file
<code>tr</code>	Substitutes one group of characters for another throughout a file
<code>uniq</code>	Eliminates duplicate lines from a text file
<code>wc</code>	Counts the number of lines, words, and characters in a text file
<code>zcat</code>	Displays a compressed file (after decompressing)

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Pregled Linux komandi

Archiving and Compressing Files

<code>compress</code>	Compresses files
<code>cpio</code>	Copies files to and from an archive
<code>gunzip</code>	Decompresses files compressed with GNU Zip (<code>gzip</code>)
<code>gzip</code>	Compresses files using GNU Zip
<code>tar</code>	Creates an archive of files in one or more directories (originally meant for archiving on tape)
<code>uncompress</code>	Decompresses files compressed with <code>compress</code>

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Pregled Linux komandi

Managing Processes	
<code>bg</code>	Runs an interrupted process in the background
<code>fg</code>	Runs a process in the foreground
<code>free</code>	Displays the amount of free and used memory in the system
<code>halt</code>	Shuts down Linux and halts the computer
<code>kill</code>	Sends a signal to a process (usually used to terminate a process)
<code>ldd</code>	Displays the shared libraries needed to run a program
<code>nice</code>	Runs a process with lower priority (referred to as <code>nice</code> mode)
<code>ps</code>	Displays a list of currently running processes
<code>printenv</code>	Displays the current environment variables
<code>pstree</code>	Similar to <code>ps</code> but shows parent-child relationships clearly
<code>reboot</code>	Stops Linux and then restarts the computer
<code>shutdown</code>	Shuts down Linux
<code>top</code>	Displays a list of most processor- and memory-intensive processes
<code>uname</code>	Displays information about the system and the Linux kernel

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Pregled Linux komandi

Managing Users

<code>chsh</code>	Changes the <i>shell</i> (command interpreter)
<code>groups</code>	Prints the list of groups that include a specified user
<code>id</code>	Displays the user and group ID for a specified user name
<code>passwd</code>	Changes the password
<code>su</code>	Starts a new shell as another user or <code>root</code> (when invoked without any argument)

Working with Date and Time

<code>cal</code>	Displays a calendar for a specified month or year
<code>date</code>	Shows the current date and time or sets a new date and time

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Pregled Linux komandi

Managing the File System	
<code>df</code>	Summarizes free and available space in all mounted storage devices
<code>du</code>	Displays disk usage information
<code>fdformat</code>	Formats a diskette
<code>fdisk</code>	Partitions a hard drive
<code>fsck</code>	Checks and repairs a file system
<code>mkfs</code>	Creates a new file system
<code>mknod</code>	Creates a device file
<code>mkswap</code>	Creates a swap space for Linux in a file or a hard drive partition
<code>mount</code>	Mounts a device (for example, the CD-ROM) on a directory in the file system
<code>swapoff</code>	Deactivates a swap space
<code>swapon</code>	Activates a swap space
<code>sync</code>	Writes <i>buffered</i> (saved in memory) data to files
<code>tty</code>	Displays the device name for the current terminal
<code>umount</code>	Unmounts a device from the file system

— Preuzimanje administratorskih prava

- Ako želimo da izvršimo neki zadatak na sistemu koji zahteva privilegije najvišeg nivoa, tada moramo da postanemo superuser/root. Prilikom pokretanja sistema obično se logujemo kao običan korisnik. Tokom rada možemo da preuzmemo root privilegije sledećom komandom:

su -

☒ nakon koje sistem od nas traži root lozinku.

- Nakon završenog rada komandom exit napuštamo root nalog.

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Upravljanje fajlovima

- `cd` – (change directory) – Menja radni direktorijum
- `pwd` – (print working directory) . prikazuje radni direktorijum
- `ls` – (list) – lista radni direktorijum

```
$ ls -l
total 0
drwxr-xr-x 2 user user 48 2007-05-26 12:07 dir
-rw-r--r-- 1 user user  0 2007-05-26 12:07 file1
-rw-r--r-- 1 user user  0 2007-05-26 12:07 file2
```

— Upravljanje fajlovima

➡ ls – (list) – lista radni direktorijum

☑ Format izlaza ls komande je sledeći:

```
drwxr-xr-x 2 user user 48 2007-05-26 12:07 dir
***** ^ **** ^^^^ ** ^^^^^^^^^^^^^^^^^^ ***
1  2      3  4      5  6      7      8
```

➡ 1) Označava tip fajla. Najčešće se susrećemo sa četiri oznake: d – direktorijum, c – character device, b – block device, “-” - običan fajl, s – socket, p – pipe, l – link

➡ 2) Dozvole. Postoje tri tipa dozvola: r – čitanje, w – upis, x – izvršavanje, “-” - ukinuta dozvola. Dozvole idu redom, po po tri za vlasnika (owner), vlasničku grupu (owner group) i svi ostali (others).

➡ 3) Broj veza ka fajlu

➡ 4) Vlasnik fajla.

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Upravljanje fajlovima

➤ ls – (list) – lista radni direktorijum

☑ Format izlaza ls komande je sledeći:

```
drwxr-xr-x 2 user user 48 2007-05-26 12:07 dir
^***** ^ **** ^^^^ ** ^^^^^^^^^^^^^^^^^^ ***
1  2      3  4      5  6          7          8
```

➤ 5) Vlasnička grupa.

➤ 6) Veličina fajla.

➤ 7) Vreme poslednje promene fajla.

➤ 8) Ime fajla.

— Upravljanje fajlovima

➤ Na UNIX like sistemima, sakriveni fajlovi imaju imena koja počinju tačkom.
Da bi smo prikazali sakrivene fajlove koristimo komandu: `ls -a`

➤ Opcije:

- ☑ `-d` lista direktorijume umesto njihovog sadržaja
- ☑ `-l` daje prošireni isting
- ☑ `-a` prikazuje sakrivene fajlove
- ☑ `-i` prikazuje inode pored fajlova
- ☑ `-R` lista direktorijumme rekurzivno
- ☑ `-s` sa opcijom `-l` daje veličinu fajlova u blokovima
- ☑ `-h` sa opcijom `-l` daje veličinu u čitljivom formatu.

—Upravljanje fajl sistemom

➤ Rukovanje fajl sistemom podrazumeva kopiranje, pomeranje i brisanje fajlova.

➤ cp – služi za kopiranje fajlova.

— cp oldfile newfile

— cp -r olddir newdir

— cp file1 file2 file3 dir

➤ Opcije

☑ -r kopira direktorijume

☑ -p čuva vlasništvo, prava pristupa i vremenske oznake fajla.

☑ -u (--update) kopira fajl samo ako je noviji od odredišnog fajla.

—Upravljanje fajl sistemom

➡ mv služi za pomeranje fajlova

```
mv oldfile newfile
```

➡ rm briše fajlove koje smo mu zadali kao parametre. Ako želimo da obrišemo direktorijum, moramo da zadamo opciju -r, pri čemu će izbrisati direktorijum i sve fajlove u njemu.

☒ -r Briše rekurzivno direktorijume

☒ -f Ignoriše nepostojeće fajlove i ništa ne pita.

—Rad sa dozvolama

- Svaki fajl na Linux-u poseduje prava pristupa za vlasnika, vlasničku grupu i sve ostale.
- U slučaju direktorijuma, pravo čitanja nam dozvoljava da listamo direktorijum, pravo upisa da kreiramo i brišemo fajlove u njemu i pravo izvršavanja da pristupimo direktorijumu.
- `chmod` menja prava pristupa nekom fajlu.

`chmod [option]... {MOD | --reference=reference_file} file1 ...`

- Dozvole možemo navesti u oktalnom i simboličkom obliku. Oktalni oblik je pogodan kada želimo da izmenimo sva prava nad fajlom. Kada se prava zadaju u simboličkom obliku, možemo da menjamo pojedina prava a da ne utičemo na ostala prava nad tim fajlom.

—Rad sa dozvolama

- Kod simboličkog zadavanja prava prisupa koristimo oznake da bi naznačili čija prava menjamo:
 - ☒ u – vlasnik
 - ☒ g – vlasnička grupa
 - ☒ o – svi ostali
 - ☒ a – sve kategorije
- `chmod u+r file`
- `chmod gu-w file`
- `chmod u+r,o-w file`
- `chmod u=rwx,go= file`
- `chmod 641 file`

— Rad sa dozvolama

- Ako želimo da promenimo prava pristupa direktorijumu i svim fajlovima u okviru tog direktorijuma koristimo opciju -R. Pri zadavanju prava pristupa u oktalno boliku treba biti pažljiv kako ne bismo zadali pogrešna prava direktorijumu i fajlovima. Preporučuje se da se prava zadaju u simboličkom obliku.

```
chmod -R go-rwx dir
```

—Rad sa dozvolama

✚ Menjanje vlasništva nad fajlovima

- ☑ Pored toga što možemo da menjamo prava pristupa za sve tri kateogrije korisnika, ponekad je potrebno da se menja vlasnik i vlasnička grupa fajla.
- ☑ Samo root ima pravo da menja vlasnika fajla, dok sam vlasnik fajla može da menja vlasničku grupu.

✚ Promena vlasništva se vrši uz pomoć komandi chown i chgrp.

```
$ ls -l fajl
-rw-r--r-- 1 user user 0 2009-03-25 17:28 fajl
$ sudo chown other_user fajl
$ ls -l fajl
-rw-r--r-- 1 other_user user 0 2009-03-25 17:28 fajl
```

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

— Rad sa dozvolama

➤ Promena vlasništva se vrši uz pomoć komandi `chown` i `chgrp`.

```
$ ls -l fajl
-rw-r--r-- 1 user user 0 2009-03-25 17:28 fajl
$ groups
user adm dialout cdrom video plugdev lpadmin admin sambashare
$ chgrp adm fajl
$ ls -l fajl
-rw-r--r-- 1 user adm 0 2009-03-25 17:28 fajl
```

— Sistemske komande

- `df` – prikazuje slobodan prostor na svim mount-ovanim particijama. Uz opciju `-h` prikazuje veličinu slobodnog prostora u B.

```
$ df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/hdb3	21G	2.3G	17G	12%	/

- `du` – komanda za procenu korišćenja fajl sistema. Komanda `ls` nam daje veličinu nekog direktorijuma u vidu fajla u kome se skladište podaci o strukturi direktorijuma. Komanda `du` omogućava procenu veličine prostora koji zauzimaju svi fajlovi u nekom direktorijumu.

```
$ du -sh /home/user/
```

```
709M      /home/user/Desktop
```

- Opcijom `-s` dobijamo sumu samo za date argumente, a opcijom `-h` dobijamo veličinu u B.

— Sistemske komande

- ↪ ako želimo da vidimo veličinu fajlova samo u direktorijumima koje smo naveli možemo da koristimo opciju `--max-depth`.

```
$ du -h --max-depth=1 /home/user/
```

```
36K  /home/user/.smplayer
4.0K /home/user/.update-notifier
46M  /home/user/.cpan
1.1M /home/user/.gconf
768K /home/user/.tomboy
21G  /home/user/Videos
30M  /home/user/.local
7.0M /home/user/.themes
174M /home/user/.mozilla
64K  /home/user/.dbus
4.0K /home/user/.gpilotd
....'
```

Upotreba komandnog interpretera

1010011
1110100
1100001
1010110

—Sistemske komande

➤ Komanda `free` daje prikaz slobodnog i zauzetog memorijskog prostora.

➤ Opcija `-m` daje prikaz u MB a ne u kB.

`$ free`

	<i>total</i>	<i>used</i>	<i>free</i>	<i>shared</i>	<i>buffers</i>	<i>cached</i>
<i>Mem:</i>	1036644	993340	43304	0	52332	703704
<i>-/+ buffers/cache:</i>		237304	799340			
<i>Swap:</i>	947792	0	947792			

➤ Komanda `top` daje informaciju o trenutnom korišćenju resursa sistema.

Upotreba komandnog interpretera

—Sistemske komande

- Komanda `top` daje informaciju o trenutnom korišćenju resursa sistema.
- Izlazak iz prikaza se vrši pritiskom na taster `q`.

```
top - 12:07:07 up 55 min, 6 users, load average: 0.10, 0.05, 0.02
Tasks: 135 total, 2 running, 133 sleeping, 0 stopped, 0 zombie
Cpu(s): 1.6%us, 0.3%sy, 0.0%ni, 96.0%id, 1.3%wa, 0.4%hi, 0.3%si, 0.0%st
Mem: 1029076k total, 907884k used, 121192k free, 44876k buffers
Swap: 10241428k total, 0k used, 10241428k free, 470136k cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6391	user	15	0	290m	41m	20m	S	6	4.1	1:05.54	ktorrent
5975	root	15	0	105m	37m	10m	R	4	3.8	0:11.63	Xorg
6411	user	15	0	125m	7668	5072	S	2	0.7	0:00.02	kio_file
6679	user	15	0	19080	1232	876	R	2	0.1	0:00.01	top
1	root	18	0	3960	888	624	S	0	0.1	0:00.95	init
2	root	10	-5	0	0	0	S	0	0.0	0:00.00	kthreadd

- Linux shell ima veliki broj malih komandi koje nam omogućavaju administraciju sistema.
 - Shell skriptovanje omogućava automatizaciju izvršavanja određenih operacija.
 - Svaki shell skript pocinje pozivom komandnog interpretera koji se koristi za izvršavanja komandi: `#!/bin/sh`
 - Primer konverzija tipa fajla:
 - ☑ Kreiramo tekst fajl pod nazivom totif
- ```
#!/bin/sh
xwdtopnm < $1.xwd > $1.pnm
pnmtotiff < $1.pnm > $1.tiff
rm $1.pnm
```
- Fajl postavljamo da bude izvršni sa: `chmod +x totif`
  - Skript pozivamo sa: `./totif figure1`