

# BASIC UNIX COMMAND LINE ASSIGNMENT

## I

- 1) ls
- 2) ls -l
- 3) ls -lh
- 4) ls -a
- 5) ls -S
- 6) ls -lt
- 7) whereis
- 8) cd
- 9) env
- 10) echo
- 11) echo \$PATH
- 12) Yes
- 13) No
- 14) whoami
- 15) ~~su~~ su -user.name
- 16) exit
- 17) reboot
- 18) shutdown [option] [time]
- 19) uname -r

20) `uname -m`

21) `uname`

22) `man uname`

23) `pwd`

24) `cd /dir-name`

25) `cd ..`

26) `cd -`

27) `cd ~`

28) `history`

29) `~/.bashrc` file

30) 1000, to change the number we must change the value of the `$HISTSIZE` and `$HISTFILESIZE` variable

31) we can modify bash's history behaviour by changing the `~/.bashrc` file.

32) `up` arrow key

### 33) Different types of shell

- Bourne shell : It's the preferred shell for shell programming because of its compactness and speed
- C shell : Incorporated features such as alias and history also built-in arithmetic and C like expression syntax
- Korn Shell : It's a superset of Bourne shell. Features comparable to C shell. Runs script written for Bourne shell.
- Bourne-Again shell : Incorporate features from the Korn and C shell. Compatible to the Bourne shell (BASH)

34) A login shell is started after a successful login. Login shell is the first process that executes under our user ID when we log into a session. A non-login shell is started by a program without a login. The program just passes the name of the shell executable.

35) When Bash is invoked as an interactive login shell it first reads and executes commands from the file `/etc/profile`, if that file exists. After that it looks for `~/.bash-profile`, `~/.bash-login`, and `~/.profile` in that order.

36) When a non-login shell is started, Bash reads and executes commands from `~/.bashrc`,

37) Shell configuration files are the files which are automatically executed when you log in and log out of a shell. They initialize and configure a shell upon login and perform cleanup operations upon logout.

38)



39) Shell variable is a variable that is set by the shell and is required by the shell in order to function correctly

- history (command history)
- LANG (set the locale)
- LPDEST (users default printer)
- MAIL (users mail)
- MANPATH (hierarchies of man page)

40) printenv

41) env | sort

42) new\_var = value

43)

44) Local variables has scope only within the file that is declared. Global can be accessed by any program started by the terminal session.

~~\*5) Quoting is used to remove the special meaning of certain characters to the shell. It disables special treatment for special chars~~

48) It's the feature that is used to match specific pattern. It's mainly used to match filenames or content in a file.

49) `ls -l *.sh`

50) `ls -l [0-9]*`

51) `ls -l [a-z]*[0-9]`

52) `ls -l ??????*`

53) Quoting is used to remove special meaning of certain characters to the shell. It disables special treatment of special characters.

54) Double quoting

→ to print Hello world

`echo Hello world`

Hello world

→ `echo "Hello world"`

Hello world

escape characters

→ to print " within an echo,

`echo "Hi \" is hello"`

~~Hi~~ Hi " hello

single quoting

-> `$echo "$echo "ug")"`

`ug`

`$echo '$(echo "ug")'`

`$(echo "ug")`

55) `find` : is used to search by a particular criteria and also manipulate files

`locate` : is used to scan the whole system quickly for something

`whereis` : simply returns the location of the executables, man pages & the source of a program

56) Globbing is used to find pattern occurrences while, `find`, `locate` & `whereis` ~~gives the location~~ used to search for a file by name.

57) Linux file system: is used for data management. It helps to arrange files on the disk storage. The file system has a hierarchical structure as it contains root directory and its subdirectory. All directories can be accessed from root directory.



58) An absolute path specifies location of a file or directory from the root directory. A relative path is defined as a path related to the pwd.

59) touch:

`$touch filename`  
creates a new file

→ cat:

`$cat > filename`  
if filename doesn't exist, then a new file is created

→ echo:

`$ echo "Hi" > filename`  
if filename is not there, then it's created

→ ~~vim~~ vim:

`$ vim filename`  
the text editor creates a new file if doesn't exist.



60) `rm {file-name}`

`$rm file-name`

→ removes the file  
`rm {f-1} {f-2}`

`$rm f_1 f_2`

removes multiple files

→ `unlink {file-name}`

`$unlink file-name`

removes single file permanently

61) `$tar cf tar.file.tar new.txt`

→ creates a tar file `file.tar` & archives `new.txt`

`$tar cf new.tar *.txt`

→ archives all `.txt` files.

62) `$tar xf new.tar *.txt`

→ extracts all `.txt` files from `new.tar`

45) `gnome-terminal -x bash -c`

46) store the variable in `/etc/environment` file

47) `export new = 5`

`echo $new`

`-> 5`

38) files in Unix system are organized into multilevel hierarchy structure known as a directory tree. At the very top of the file system is directory called root which is represented by a `"/"`. All other files are descendants of root.

```

[ec2-user@ip-172-31-46-226 ~]$ ls -a
.  ..  .bash_logout  .bash_profile  .bashrc  .ssh
[ec2-user@ip-172-31-46-226 ~]$ ls -al
total 12
drwx----- 3 ec2-user ec2-user 74 Nov 11 12:06 .
drwxr-xr-x 3 root     root     22 Nov 11 12:06 ..
-rw-r--r-- 1 ec2-user ec2-user 18 Jul 15 2020 .bash_logout
-rw-r--r-- 1 ec2-user ec2-user 193 Jul 15 2020 .bash_profile
-rw-r--r-- 1 ec2-user ec2-user 231 Jul 15 2020 .bashrc
drwx----- 2 ec2-user ec2-user 29 Nov 11 12:06 .ssh
[ec2-user@ip-172-31-46-226 ~]$ ls -as
total 12
0 . 0 .. 4 .bash_logout 4 .bash_profile 4 .bashrc 0 .ssh

```

```

[ec2-user@ip-172-31-46-226 ~]$ ls -lat
total 12
drwx----- 3 ec2-user ec2-user 74 Nov 11 12:06 .
drwx----- 2 ec2-user ec2-user 29 Nov 11 12:06 .ssh
drwxr-xr-x 3 root     root     22 Nov 11 12:06 ..
-rw-r--r-- 1 ec2-user ec2-user 18 Jul 15 2020 .bash_logout
-rw-r--r-- 1 ec2-user ec2-user 193 Jul 15 2020 .bash_profile
-rw-r--r-- 1 ec2-user ec2-user 231 Jul 15 2020 .bashrc
[ec2-user@ip-172-31-46-226 ~]$ whereis

```

Usage:

```
whereis [options] [-BMS <dir>... -f] <name>
```

Locate the binary, source, and manual-page files for a command.

Options:

```

-b          search only for binaries
-B <dirs>   define binaries lookup path
-m          search only for manuals and infos
-M <dirs>   define man and info lookup path
-s          search only for sources
-S <dirs>   define sources lookup path
-f          terminate <dirs> argument list
-u          search for unusual entries
-l          output effective lookup paths

```

For more details see whereis(1).

```
[ec2-user@ip-172-31-46-226 ~]$ env
XDG_SESSION_ID=1
HOSTNAME=ip-172-31-46-226.us-east-2.compute.internal
TERM=linux
SHELL=/bin/bash
HISTSIZE=1000
SSH_CLIENT=3.16.146.0 8431 22
SSH_TTY=/dev/pts/0
USER=ec2-user
LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33:cd=40;33:or=40;31:mi=01;05;37;41:su=37;41:sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;31:*.arc=01;31:*.arj=01;31:*.taz=01;31:*.lha=01;31:*.lz4=01;31:*.lzh=01;31:*.lzma=01;31:*.tlz=01;31:*.txz=01;31:*.tzo=01;31:*.t7z=01;31:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lrz=01;31:*.lz=01;31:*.lzo=01;31:*.xz=01;31:*.bz2=01;31:*.bz=01;31:*.tbz=01;31:*.tbz2=01;31:*.tz=01;31:*.deb=01;31:*.rpm=01;31:*.jar=01;31:*.war=01;31:*.ear=01;31:*.sar=01;31:*.rar=01;31:*.alz=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31:*.7z=01;31:*.rz=01;31:*.cab=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.webm=01;35:*.ogm=01;35:*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.ogx=01;35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.mid=01;36:*.midi=01;36:*.mka=01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.axa=01;36:*.oga=01;36:*.spx=01;36:*.xspf=01;36:
MAIL=/var/spool/mail/ec2-user
PATH=/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/ec2-user/.local/bin:/home/ec2-user/bin
PWD=/home/ec2-user
LANG=en_US.UTF-8
HISTCONTROL=ignoredups
SHLVL=1
```

```
[ec2-user@ip-172-31-46-226 ~]$ echo $PATH
/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin:/home/ec2-user/.local/bin:/home/ec2-user/bin
[ec2-user@ip-172-31-46-226 ~]$ whoami
ec2-user
[ec2-user@ip-172-31-46-226 ~]$ uname -r
4.14.248-189.473.amzn2.x86_64
[ec2-user@ip-172-31-46-226 ~]$ uname -m
x86_64
[ec2-user@ip-172-31-46-226 ~]$ pwd
/home/ec2-user
[ec2-user@ip-172-31-46-226 ~]$ cd-
-bash: cd-: command not found
[ec2-user@ip-172-31-46-226 ~]$ cd ~
[ec2-user@ip-172-31-46-226 ~]$
```

```
[ec2-user@ip-172-31-46-226 ~]$ history
 1  ls -a
 2  ls -al
 3  ls -as
 4  ls -lt
 5  ls -lat
 6  whereis
 7  cd
 8  env
 9  clear
10  echo $PATH
11  whoami
12  uname -r
13  uname -m
14  pwd
15  cd~
16  cd ~
17  clear
18  history
[ec2-user@ip-172-31-46-226 ~]$
```