



Vodafone
India
Foundation

_VOIS



Shell Commands



Vodafone
India
Foundation

_VOIS



Contents

- What is a Shell?
- Shell commands



Why to use CLI over GUI?

- Less resources (graphics components add to more required memory)
- High precision
- Repetitive tasks friendly
- Powerful



Vodafone
India
Foundation

_VOIS



But always remember.....

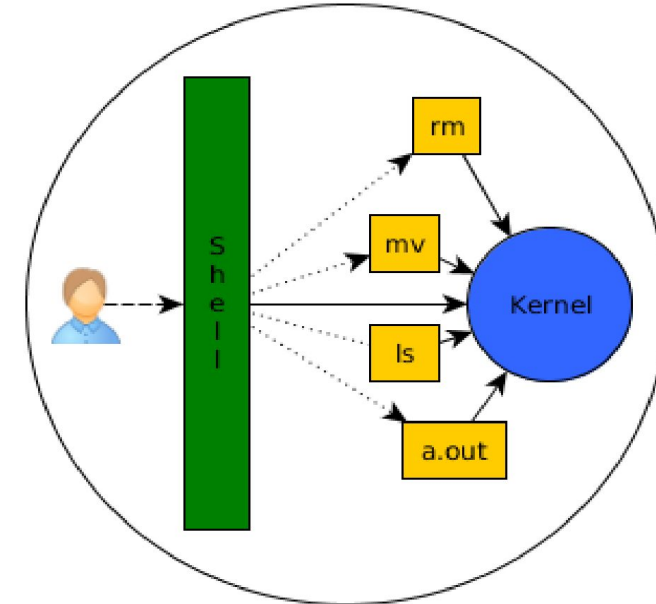


Image source - <https://knowyourmeme.com/memes/with-great-power-comes-great-responsibility>



What is a Shell ?

- Shell is a command language interpreter that executes commands read from the standard input device (keyboard) or any file.
- Shell is not a part of system kernel but uses the kernel to execute programs.
- **cat /etc/shells** command will give the various shells in our system.
- **Sh – simple shell**
- **BASH – Bourne Again Shell**
- **KSH – Korne Shell**
- **CSH – C Shell**
- **SSH – Secure Shell**





Vodafone
India
Foundation

_VOIS



Want to check which type of Shell you are using?

```
echo $SHELL
```



How to check what a particular command do?

- `man <command>`
 - e.g. `man ls`



How to check the list of files in a folder?

- **ls -a <folder>** (all) Lists all the files (including .*files)
- **ls -l <folder>** (long) Long listing (type, date, size, owner, permissions)
- **ls -t <folder>** (time) Lists the most recent files first
- **ls -S <folder>** (size) Lists the biggest files first
- **ls -r <folder>** (reverse) Reverses the sort order
- **ls -ltr <folder>** (options can be combined) Long listing, most recent files at the end



Vodafone
India
Foundation

_VOIS



File name pattern substitutions

ls *txt

The shell first replaces `*txt` by all the file and directory names ending by txt (including .txt), except those starting with `.`, and then executes the `ls` command line. **This will work when you are inside the directory.**

ls -d .*

Lists all the files and directories starting with `.` and `d` tells `ls` not to display the contents of directories.

That's great that it worked but what was `.` And `..` ?

cat ?.log

Displays all the files which names start by 1 character and end by .log



Working with directories

command	function
<code>pwd</code>	print working directory
<code>mkdir dirName</code>	make directory
<code>mkdir -p dirName</code>	create parent directory
<code>rmdir dirName</code>	remove directory
<code>rm -r dirName</code> or <code>rm -ri dirName</code> or <code>rm -rf dirName</code>	remove dir recursively (add f for force), -i with permission
<code>cd /dirName/dirPath</code>	go to a specific dir
<code>cd</code>	takes you to home directory
<code>cd ~</code>	home directory
<code>cd ..</code>	previous directory
<code>cd -</code>	previous dir with path



Practice: working with directories

- Display your current directory.
- Change to the /etc directory.
- Now change to your home directory using only three key presses.
- Change to the /boot/grub directory using only eleven key presses.
- Go to the parent directory of the current directory.
- Go to the root directory.
- List the contents of the root directory.
- List a long listing of the root directory.
- Stay where you are, and list the contents of /etc.
- Stay where you are, and list the contents of /bin and /sbin.
- Stay where you are, and list the contents of ~.
- List all the files (including hidden files) in your home directory.
- List the files in /boot in a human readable format.
- Create a directory testdir in your home directory.
- Change to the /etc directory, stay here and create a directory newdir in your home directory.
- Create in one command the directories ~/dir1/dir2/dir3 (dir3 is a subdirectory from dir2, and dir2 is a subdirectory from dir1).
- Remove the directory testdir.



Working with files

“Everything in Linux is a file, Everything in Linux that is not a file is a process”

Let's learn how to recognise, create, remove, copy and move files using commands:

command	function
touch filename1 fileName2 fileName3	easy way to create a file
cp file1 file2 or cp fileName filePath	copy content of a file to other file
mv fileName1 fileName2 or mv fileName1 filePath/dirPath	move file content or file to other dir
cp -r dirName1 dirName2	recursive copy
cp file1 file2 file3 file4 file5 /dirName	cp multiple files to directory
head	display first 10 lines
tail	display last 10 lines
cat	display content
cat > fileName	create file with concatenated content (ctrl+d for EOL)



File, Users, Groups and Permissions





Users, Groups and Permissions

- Use **ls -l** to check file access rights
- 3 types of access rights
 - Read access (r)
 - Write access (w)
 - Execute rights (x)
- 3 types of access levels
 - **User (u)**: for the owner of the file
 - **Group (g)**: each file also has a “group” attribute, corresponding to a given list of users
 - **Others (o)**: for all other users



Vodafone
India
Foundation

_VOIS



Changing permissions

`chmod <permissions> <files>`

`sudo chmod --- fileName`

1. Numerical values

Read = 4 , write = 2 , execute = 1

e.g.

`-r--r--r-- 1 sagar sagar 7 Jul 21 21:14 te2.txt`

`-rw-rw-rw- 1 sagar sagar 7 Jul 21 21:14 te2.txt`



Changing permissions

- 2. Symbolic way
 - **chmod go+r:** add read permissions to group and others.
 - **chmod u-w:** remove write permissions from user.
 - **chmod a-x:** (a: all) remove execute permission from all.
 - **chmod R a+rX linux/:** Makes linux and everything in it available to everyone!
- R: apply changes recursively
 - X: x, but only for directories and files already executable



File ownership

- Particularly useful in (embedded) system development when you create files for another system.
- We should have multiple users to change ownership, add a user first:
 - `sudo adduser username`
 - `sudo adduser sco`
 - to delete a user:
 - `sudo deluser username`

Now create a file/dir and change owner

- **`chown -R sco /home/linux/src`** (R: recursive) Makes user sco the new owner of all the files in /home/linux/src.
- **`chgrp -R empire /home/askywalker`**: Makes empire the new group of everything in /home/askywalker.



The root account

- In case you really want to use root.
- If you have the root password:
- **sudo -s**(switch user)
- **sudo** command gives you access to some root privileges with your own user password.
- for example **sudo chmod 777 fileName.sh**



Standard I/O, redirections, pipes

- More about command output
- All the commands outputting text on your terminal do it by writing to their standard output.
- **echo** content - outputs the strings it is being passed as arguments
- **>** redirection operator a standard output can be written (redirected) to a file
- **>>** Standard output can be appended to an existing file

```
echo "Welcome to NSTI, Bangalore" > sample.txt
```

```
echo "how's your day going?" >> sample.txt
```



Vodafone
India
Foundation

_VOIS



command	function
grep pattern fileName	searches a file or files for lines that have a certain pattern
 pipe	Pipes ' ' send the output of one command as input of another command.
sort	'sort' command sorts out the content of a file alphabetically



Vodafone
India
Foundation

_VOIS



- e.g.
- `grep "123" te4.txt | cat > te7.txt`



Process and jobs

- “Everything in Unix is a file , Everything in Unix that is not a file is a process”
- Instances of a running programs
- Several instances of the same program can run at the same time
- Data associated to processes:
- Open files, allocated memory, stack, process id, parent, priority, state...
- PID: Process id
- VSZ: Virtual process size (code + data + stack)
- RSS: Process resident size: number of KB currently in RAM
- TTY: Terminal
- STAT: Status: R (Runnable), S (Sleep), W (paging), Z (Zombie)...



ps ux	Lists all the processes belonging to the current user
top	Displays most important processes, sorted by cpu percentage
kill <pids>	Sends an abort signal to the given processes. Lets processes save data and exit by themselves
kill 9 <pids>	Sends an immediate termination signal
ps tree	display a tree of processes



Environment variables

Shells let the user define variables. They can be reused in shell commands.

Convention: lower case names

variables that are also visible within scripts or executables called from the shell.

Convention: upper case names.

env	Lists all defined environment variables and their value.
SHELL	Current shell name
TERM	Current terminal type
USER	Current user name
HOSTNAME	Name of the local machine
PATH	Command search path



Vodafone
India
Foundation

_VOIS



- **~/.bashrc file**
- Shell script read each time a bash shell is started
- You can use this file to define
- Your default environment variables (PATH, EDITOR...).
- Your aliases.
- A greeting message.



. Miscellaneous

du -h <file>	returns size on disk of the given file
df -h	Returns disk usage and free space for the filesystem containing the given directory.
tar -cvf <archive> <files or directories> tar -cvf sample.tar sample	c: create v: verbose. Useful to follow archiving progress. f: file. Archive created in file (tape used otherwise).
tar -xvf <archive>	x - extract
gzip fileName	create gz file (gnu zip)



- **Getting information about users**

- ▶ **who**

- Lists all the users logged on the system.

- ▶ **whoami**

- Tells what user I am logged as.

- ▶ **groups**

- Tells which groups I belong to.

- ▶ **groups <user>**

- Tells which groups <user> belongs to.

- ▶ **finger <user>**

- Tells more details (real name, etc) about <user>

- Disabled in some systems (security reasons).



Vodafone
India
Foundation

_VOIS



- **text editors**

- vi
- vim
- nano
- gedit