The topics:

- **Checking system activity**
- **Exiting the shell**
- **Getting Help with Using the Shell**
- **file permission**
- **Change permission on a file – numeric**
- **Change permission on a file – symbolic**

Checking system activity:

$ ps

Reports the process status
The ps command is used to provide information about the currently running processes and monitoring system usage.
The basic syntax of ps is : `ps [options]`

$ ps -au

$ ps au

USE	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	2164	0.0	0.8	1908	1100	ttyp0	S	14:50	0:00	login - - jake
jake	2147	0.0	0.7	1836	1020	ttyp0	S	14:50	0:00	-bash
jake	2310	0.0	0.7	2592	912	ttyp0	R	18:22	0:00	ps au

a : option tells ps to list the processes of all users on the system
u : option tells ps to provide detailed information about each process.

$ kill [PID]
To stop any process by PID i.e. to kill process
e.g. : $ kill 2310
$ kill 0
To stop all process except your shell.

Some time we use the signal (-9) to kill the process : kill -9 2310

Exiting the shell

$ exit

To exit the shell

Getting Help with Using the Shell

- Use the ***help*** command (**$ help**)
- Use the ***man command*** ($ man)

Man :Short for **Manual**

$ **man** : display the manual of a command.

To reading a manual and return to the shell prompt within the open terminal ➔ press q

Understanding file permission

Why Are File Permissions Important ?

- Keep users from accessing other users' private files
- To protect important system files

permissions bits:

rwx rwx rwx

Owner's| Group | Others

r = read w = write x = execute

For Files:

- "**Read**" means to be able to open and view the file
- "**Write**" means to overwrite or modify the file
- "**eXecute**" means to run the file as a binary
  files are executable only if they are programs and shell scripts, not useful for data files.

For Directories:

- "**Read**" means to be able to view the contents of the directory
- "**Write**" means to be able to create new files/directories or delete files/directories within the directory
- "**eXecute**" means to be able to "Change Directory" (cd) into the directory = permission to access the directory.

- **How to view the permission for a file or directory?**

  $ ls -al

Default Permissions on Linux

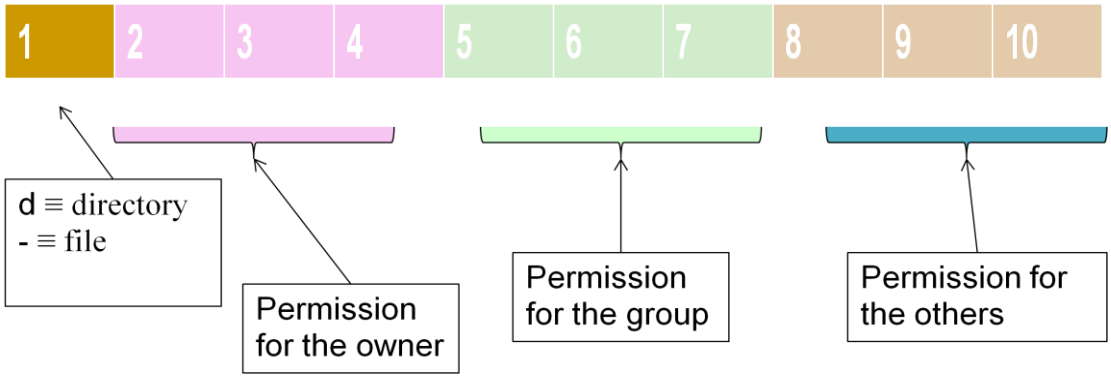
- **In most Linux Distributions, When a new file is created, by default its permissions will set to:**
  - **rw by the owner of the file.**
  - **r by group members and others.**

  rw- r-- r--

- **Default permission for an new directory:**
  - **rwx by the owner of the directory**
  - **rx by group members and others.**

  rwx r-x r-x

```
$ ls –al
-rw-rw-r--   1   chris   sales   1024   May  10   01:49   ..
drwxr-xr-x   2   chris   sales   2204   May  18   21:30   .bash_history
```



If the permission field is - ➔ the permission is not given

- Only the **owner** of a file can change its permission.

How to set file permission?

Use the command **chmod**(**ch**ange file **mod**e bits).

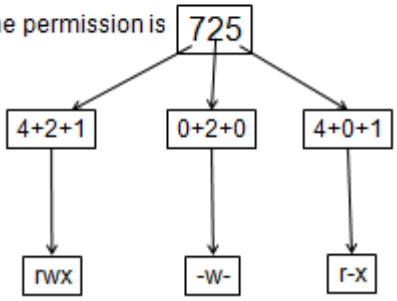
chmod has two notations:

- ❑ Numeric (Octal) Notation.
- ❑ Symbolic Notation.

Change permission on a file - numeric

- the file permissions aren't represented by characters. Instead, they are represented by a three-digit octal number.
- **4** = read (r)
  2 = write (w)
  1 = execute (x)
  0 = no permission (-)

If the permission is 725



Octal#	(421)	Binary	permission
0	0+0+0	000	---
1	0+0+1	001	--x
2	0+2+0	010	-w-
3	0+2+1	011	-wx
4	4+0+0	100	r--
5	4+0+1	101	r-x
6	4+2+0	110	rw-
7	4+2+1	111	rwx

Change permission on a file - symbolic

- Permissions are represented by characters rwx

chmod	who	+ / -	permission	filename

- This gives "**who**" the specified **permissions** for a given **filename.**
- The "who" is a list of letters re going to be giving permissions to. These may be specified in any order.

u	The **u**ser who owns the file (this means "you.")
g	The **g**roup the file belongs to.
o	The **o**ther users not in the file's group.
a	**a**ll of the above (an abbreviation for ugo)

- + ➔ add the selected permission.
- - ➔ remove the selected permission.

Change permission on a file

r = 4 w = 2 x = 1

$ chmod 777 file-name → **rwxrwxrwx**

$ chmod 755 file-name → **rwxr-xr-x**

owner(u) group(g) other(o) all(a) **rwxrwxrwx**

$ chmod a-w file-name → **r-xr-xr-x**

$ chmod go-rwx file-name → **rwx------**

owner(u) group(g) other(o) all(a) **---------**

$ chmod u+rw file-name → **rw-------**

http://www.computerhope.com/unix.htm#05

http://www.linfo.org/command_index.html