**UNIX/SHELL SCRIPTING**

<https://www.howtogeek.com/412055/37-important-linux-commands-you-should-know/>

**ls**  --------------> Listing all the files in the current directory

**ls –l** --------------> For long listing format of the particular file or directory

**man ls** -------------> To know all the options for modification

**ls –lr** --------------> To get reverse order

**ls –lt** --------------> To get files based on the order of time

**ls –a** --------------> To list all the files

**.file name** --------------> To hide the files

**..** --------------> One directory upwards(previous)

**When the permission starts with:**

**d** ------->directory

(**-**) ------>regular file

**l**  --------->symbolic link

**d** or **b** -------->block

**Next 9 characters:**

rwx----->**owner** or **user**

rwx----->**group**

rwx----->**others**

**To change the access of this permission we have 2 ways:**

**1)using chmod command**

Ex: chmod (g or u or o)+r/w/x (file name) or chmod g=rwx filename

\* **chmod g+w test1 or chmod g=rwx test1**.

**2)Octal notation**

read(r)=4

Write(w)=2

Execute(x)=1

ex: Q. -rwxr-xrw- test1

A**. chmod 756 test1**

**grep 993(grpid) group**------------> To know which group it belongs to.

**Types of shell:**

* Kornshell--ksh
* Bourne shell---bash
* Cshell----csh
* Simple shell----sh
* Turbo C shell-----tcsh

**X**---->It is the encrypted password which will be present in etc shadow file. Only root have the access to this. But it can only reset the password and cannot retrieve that.

**Id** ---------------------> To know group id and group name.

**Cd** --------------------->(change directory) To navigate between different directories

**Pwd** ------------------->(print working directory) To know the current directory

**mkdir** ----------------> To create directory

**man mkdir**---------------> To know the options in creating directory....ex=mkdir dir1

**mkdir –p /temp/b1/c1/a1/z1**--->To create more than 1 directory (subdirectories)

**touch filename**-------------> To create the file.....**ex**=**touch z1** , **touch “user test”**(to include space in the name) or **touch user\ test** (using escape character \ )

**ls -l filename**------------> To check the permission of file......**ex**=ls -l z1

**rm** --------------------> To remove or delete file

**rmdir** ----------------> To remove directory(it removes empty directory only)

If directory is not empty then ---🡪 rm –r <directory name> (recursive deleting)

**To give default permissions to new files which creates:**

Max value for permissions = 666

Value u want to give for ex :rw-r--r-- = 644

so umask value is = 022

**umask 022**----------------> To make the default value for new files.

**Wild Characters (**<https://kompjuteras.com/en/wildcards-in-linux-commands-with-practical-examples/>)( <http://tldp.org/LDP/GNU-Linux-Tools-Summary/html/x11655.htm>)

Consider there are four directories/files created: test1 test2 ttst task devops spring java.

* (**\***)--------> It shows anything.

**Example** : When we want all the files which starts with t then the command will be

**ls –l t\***

* (**?**)--------> It denotes single character**.**

**Example**: When we want all the files in which 3rd  letter is S then the command will be

**ls –l ??s\***

* ( **[ ]** )---------> matches any occurrence of character enclosed in the square brackets.inside bracket only 1 character is counted.

**Example** **: ls –l t[te]\*** ( it denotes the file starting from t and which contain t or e as 2nd character and continue one or more characters)

* ( **!** )------> excludes from serach.

**Example** : **ls -l [!st]\*** (excluding files which starts from s and t.

**ls –l [!d-S]\*** (excluding files which starts from letters between d to s.

* ( **\** )------🡪 escape character.

Ex : to list files having ? we can use this.

* **[[:named:]]** ----------🡪 Named character classes…will list named values.i,e as follows:

Alpha – all letters

Alnum – letters and numbers

Digit – only numbers

Lower – lowercase

Space – files with spaces

Upper – uppercases

Example:

1. This will list all files who begins with letters

**ls -l [[:alpha:]]\***

1. This will list all files who begins with letters or numbers

**ls -l [[:alnum:]]\***

1. This will list all files who have numbers in name

**ls -l \*[[:digit:]]\***

1. This will list all files who have spaces in name

**ls -l \*[[:space:]]\***

1. This will list all files who have UPERCASES

**ls -l \*[[:upper:]]\***

1. This will list all files who have lowercases

**ls -l \*[[:lower:]]\***

<https://www.computernetworkingnotes.com/rhce-study-guide/differences-between-absolute-path-and-relative-path-in-linux.html>

**What is an absolute path?**An absolute path is defined as the specifying the location of a file or directory from the root directory(/). In other words we can say absolute path is a complete path from start of actual filesystem from / directory.  
  
**Some examples of absolute path:**  
/var/ftp/pub  
/etc/samba.smb.conf  
/boot/grub/grub.conf

If you see all these paths started from / directory which is a root directory for every Linux/Unix machines.  
  
**What is the relative path?**  
Relative path is defined as path related to the present working directory(pwd). Suppose I am located in /var/log and I want to change directory to /var/log/kernel. I can use relative path concept to change directory to kernel  
  
changing directory to /var/log/kernel by using relative path concept.  
  
pwd  
/var/log  
cd kernel  
Note: If you observe there is no / before kernel which indicates it's a relative directory to present working directory.  
  
Changing directory to /var/log/kernel using absolute path concept.  
  
cd /var/log/kernel  
Note: We can use an absolute path from any location where as if you want to use relative path we should be present in a directory where we are going to specify relative to that present working directory.

**Command to view the files**

cat <filename>

more <filename>

less <filename>

* Cat is also used for concatenation of the files. It views the file in continuous manner
* The command for concatenation is **cat <file1> <file2>**
* More and less command view the file in the forms of pages.
* In More and less commands there is also a command for searching the patterns that is “ / “ .and for next type “n” and for previous type “N”.

**Commands to copy a file from directory to directory**

**Syntax : cp source target**

* Ex: cp test /tmp
* When we want to copy and change the file name in destination then **cp file1 /tmp/file2**.
* To keep the time unchanged we can use **–p (preserve)**

Ex: **cp –p testdir1 /testdir2**

* To copy the directory we can use **cp –r dir1 /dir2**

**Move commands**

Move commands are used to move the files and directory and also to rename the files.

Syntax: **mv source /target**

Example: **mv testdir /root**

**mv testdir /root/newdir**

**Redirection commands**

**>** --------------🡪 to override the file

**>>** ---------------🡪 To append the file

Example : **cat file1 file2 > newfile**

**Cat file1 file2 >> newfile**

**Egrep “redhat|fedora” file? > output1**

**<** --------------------🡪 input redirector

Ex : **cat < demo2**

**Search pattern commands**

* To search something we use a command called **grep**

Example : **grep fedora file?**

* To know the line in wich the searched content is present

Example : **grep –n fedora file?**

* This is case sensitive. So, to make case insensitive use the command –i.

Example : **grep -I fedora file?**

* We use –v as a negate command

Example : **grep –v redhat file?**

* To pass the multiple search content we can use E (extended) command

Example : **grep –E “redhat|fedora” file?**

Or

**egrep “redhat|fedora” file?**

* Same way fgrep command searches for the fixed character strings in a files.

<https://www.geeksforgeeks.org/fgrep-command-in-linux-with-examples/>

**File Descriptors**

There are 2 standard file descriptors: Standard output which indicates 1 and standard error which indicates 2.

* When we want to save only the output : ls demo?

**ls \*txt 1> output1**

* When we want to save only the error into a file :

**ls demo? \*txt 2> error1**

* When we want to save both error and output : ls demo?

**ls \*txt > combined 2>&1**

* When we want to save separately but in a single command :

**ls demo? \*txt 1> o1 2> err1**

* When we want to append the file :

**ls \*txt >> combined 2>&1**

**Pipe ( | )**---------🡪 It is used to give the output of one command as the input to the other command.

Ex : **ls | grep txt**

**Head**-------🡪It is used to display the first 10 lines of the file.

Ex : **head demo1** , **head -5 demo1**

**Tail**---------🡪It is used to display the last 10 lines of the file.

Ex : **tail demo1** , **tail -5 demo1**

**Editors :** vi editor emacs nano pico nedit gedit

**Vi editor Commands**

There are 3 modes in vi editor:

**1) Command** or **colon** mode is for executing the command for save quit etc.

**2) Insert** mode is for entering the data.

**3) Escape** mode is for navigation.

**Commands for using vi editor:**

* **vi <filename>** to view the file.
* To enter the text go to insert mode using **i**
* Enter the data and press **esc** to return to the command mode.
* Then **:wq** use this command to save and exit the vi
* **:q!** is for quit without saving.
* **:se nu** is to see the line number.
* **:se list** is to see the list
* **Ctrl + f** is for going to next page.
* **Ctrl + b** is for previous page
* **/** search in forward direction and **?** is for searching in backward direction.
* **n** is for nextand **N** is for previous.
* **Ctrl + h** is for find and replace

Syntax---- :%s /word to be replced /replacble word/g(globally)

* **U** is for undo.
* **X** is for delete.
* **Xdd** is for deleting whole line (x---no of line to be deleted).
* **Xyy** is for copying. (x is for no of lines to be copied.)
* **P** is for paste.
* **^** is for beginning of a line.