Introduction to Linux

Linux Commands

Basic Linux Commands

Linux provides a **CLI** (Command Line Interface) to communicate with the OS. Here are the most basic of the Linux Commands.

1. pwd

Displays the current working directory of the terminal.

syntax:

```
$ pwd
```

```
edureka@edureka:~$ pwd
/home/edureka
```

2. echo

Writes its arguments to standard output.

syntax:

```
$ echo "<text>"
```

```
edureka@edureka:~$ echo "Hello Learner"
Hello Learner
```

3. clear

Clears the terminal screen. Contents will not actually be deleted in this case, only scrolled down. You can also clear the screen by pressing ***Ctrl+L*** on the keyboard.

syntax:

```
$ clear
```

Working with Files

1. cp

Copies files and directories. A copy of the file/directory copied, still remains in the working directory.

syntax:

```
$ cp <flag> {filename} /pathname/
```

```
edureka@edureka:~$ cp testfile.txt /home/edureka/Public/
edureka@edureka:~$ ls /home/edureka/Public/
testfile.txt
edureka@edureka:~$ ls
Desktop    Downloads         Music     Public     testfile.txt
Documents  examples.desktop  Pictures  Templates  Videos
```

Command	Explanation
cp -i	Enters interactive mode; CLI asks before overwriting files
cp -n	Does not overwrite the file
cp -u	Updates the destination file only when the source file is different from the destination file
cp -r	Recursive copy for copying directories; Copies even hidden files
cp -v	Verbose; Prints informative messages

2. mv

Moves files and directories from one directory to another. The file/directory once moved, is deleted from the working directory.

syntax:

```
$ mv <flag> {filename} /pathname/
```

```
edureka@edureka:~$ mv testfile.txt /home/edureka/Public/
edureka@edureka:~$ ls /home/edureka/Public/
testfile.txt
edureka@edureka:~$ ls
Desktop     Downloads         Music     Public     Videos
Documents   examples.desktop  Pictures  Templates
```

Command	Explanation
mv -i	Enters interactive mode; CLI asks before overwriting files
mv -u	Updates the destination file only when the source file is different from the destination file
mv -v	Verbose; Prints source and destination files

3. rm

Removes files from a directory. By default, the rm command does not remove directories. Once removed, the contents of a file cannot be recovered.

syntax:

```
$ rm <flag> {filename}
```

```
edureka@edureka:~$ ls
Desktop     Downloads         Music     Public     testfile.txt
Documents   examples.desktop  Pictures  Templates  Videos
edureka@edureka:~$ rm testfile.txt
edureka@edureka:~$ ls
Desktop     Downloads         Music     Public     Videos
Documents   examples.desktop  Pictures  Templates
```

Command	Explanation
rm –r	Removes even non-empty directories.
rm –rp	Removes non-empty directories including parent and subdirectories.

4. grep

Searches for a particular string/word in a text file. This is similar to "Ctrl+F" but, executed via a CLI.

syntax:

```
$ grep <flag or element_to_search> {filename}
```

```
edureka@edureka:~$ cat greptestfile.txt
test test test file1 file2 file3 test test test
edureka@edureka:~$ grep file greptestfile.txt
test test test file1 file2 file3 test test test
```

Command	Explanation
grep -i	Returns the results for case insensitive strings
grep -n	Returns the matching strings along with their line number
grep -v	Returns the result of lines not matching the search string
grep -c	Returns the number of lines in which the results matched the search string

5. cat

Used to read, modify or concatenate text files. It also displays file contents.

syntax:

```
$ cat <flag> {filename}
```

```
edureka@edureka:~$ cat cattestfile.txt
This is a test file to check for concatenate command.
```

Command	Explanation
cat -b	This is used to add line numbers to non-blank lines
cat -n	This is used to add line numbers to all lines
cat -s	This is used to squeeze blank lines into one line
cat –e	Show $ at the end of line

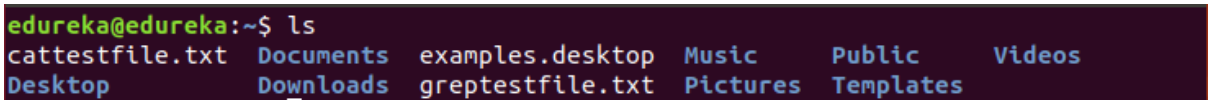
Working with Directories

1. ls

Lists all the contents in the current working directory.

syntax:

```
$ ls <flag>
```



Command	Explanation
ls <path name>	By specifying the path after ls, the content in that path will be displayed
ls –l	Using 'l' flag, lists all the contents along with its owner settings, permissions & time stamp (long format)
ls –a	Using 'a' flag, lists all the hidden contents in the specified directory
ls –author	Using '–author' flag, lists the contents in the specified directory along with its owner
ls –s	Using 's' flag, sorts and lists all the contents in the specified directory by size
ls *.html	Using '*' flag, lists only the contents in the directory of a particular format
ls –lS > file.txt	Using '>' flag, copies the result of ls command into a text file

2. cd

Used to change the current working directory of the user.

syntax:

```
$ cd /pathname/
```

```
edureka@edureka:~$ cd /home/edureka/Public/
edureka@edureka:~/Public$
```

Command	Explanation
cd ~	This command also changes the directory to home directory
cd /	Changes the directory to root directory
cd ..	Changes the directory to its parent directory
cd 'xx yy'	We specify the folder name in inverted commas because there is a space in the folder name

3. sort

Sorts the results of a search either alphabetically or numerically. Files, file contents and directories can be sorted using this command.

syntax:

```
$ sort <flag> {filename}
```

```
edureka@edureka:~$ cat sorttestfile.txt
alias
cat
cd
rm
mv
cp
man + help
ls
ssh
edureka@edureka:~$ sort sorttestfile.txt
alias
cat
cd
cp
ls
man + help
mv
rm
ssh
```

Command	Explanation
sort -r	the flag returns the results in reverse order;
sort -f	the flag does case insensitive sorting
sort -n	the flag returns the results as per numerical order

4. mkdir

Creates a new directory.

syntax:

```
$ mkdir <flag> {directoryname} /pathname/
```

```
edureka@edureka:~$ mkdir newdirectory
edureka@edureka:~$ ls
cattestfile.txt  Downloads         Music        Public            Videos
Desktop          examples.desktop  newdirectory sorttestfile.txt
Documents        greptestfile.txt  Pictures     Templates
```

Command	Explanation
mkdir -p	Creates both a new parent directory and a sub-directory

mkdir –p <filename1>/{f1,f2,f3}	This is used to create multiple subdirectories inside the new parent directory

5. rmdir

Removes a specified directory. Although by default, it can only remove an empty directory, there are flags which can be deployed to delete the non-empty directories as well.

syntax:

```
$ rmdir <flag> {directoryname}
```

```
edureka@edureka:~$ ls
cattestfile.txt  Downloads         Music         Public            Videos
Desktop          examples.desktop  newdirectory  sorttestfile.txt
Documents        greptestfile.txt  Pictures      Templates
edureka@edureka:~$ rmdir newdirectory
edureka@edureka:~$ ls
cattestfile.txt  Downloads         Music     sorttestfile.txt
Desktop          examples.desktop  Pictures  Templates
Documents        greptestfile.txt  Public    Videos
```

Command	Explanation
rmdir –p	Removes both the parent and child directory
rmdir –pv	Removes all the parent and subdirectories

User Permissions

1. su

Switches to root-user so that superuser permissions can be used to execute commands.

syntax:

```
$ su
```

2. su <username>

Switches to a different user whose name is passed as the argument.

syntax:

```
$ su <username>
```

3. sudo

Executes only that command with root/ superuser privileges.

syntax:

```
$ sudo <command>
```


Command	Explanation
sudo useradd <username>	Adding a new user
sudo passwd <username>	Setting a password for the new user
sudo userdel <username>	Deleting the user
sudo groupadd <groupname>	Adding a new group
sudo groupdel <groupname>	Deleting the group
sudo usermod -g <groupname> <username>	Adding a user to a primary group

4. chmod

Used to change the access permissions of files and directories. Consider the example below.

```
edureka@edureka:~$ cat chmodtestfile.sh
#!/bin/sh

echo "This is a test file to test permissions"
edureka@edureka:~$ ./chmodtestfile.sh
bash: ./chmodtestfile.sh: Permission denied
```

On trying to run the newly created file named ***chmodtestfile.sh***, an error is thrown. After modifying the permissions of the file using the said Linux command, it turns executable.

syntax:

```
$ chmod <permissions of user,group,others> {filename}
```

```
edureka@edureka:~$ chmod 777 chmodtestfile.sh
edureka@edureka:~$ ./chmodtestfile.sh
This is a test file to test permissions
edureka@edureka:~$
```

The permissions associated with each digit is as follows.

Number	read	write	execute
0	–	–	–
1	–	–	yes
2	–	yes	–
3	–	yes	yes
4	yes	–	–
5	yes	–	yes
6	yes	yes	–
7	yes	yes	yes

Installing Packages

Stable versions of most software's will already be available in Linux repositories. Here are the Linux Commands to install them.

1. Install packages

For an RHEL based system;

syntax:

```
$ sudo yum install package-name
```

For a Debian based system;

syntax:

```
$ sudo apt-get install package-name
```

For a Fedora based system;

syntax:

```
$ sudo dnf install package-name
```


Zipped Files

When you download a package from the internet, the downloaded file comes in compressed form. Here are a few commands to decompress and compress files in Linux.

1. tar

Zips/Compresses files of ***.tar*** format.

syntax:

```
$ tar -cvf tar-filename source-folder-name
```

Unzips/Decompresses files of ***.tar*** format.

syntax:

```
$ tar -xvf tar-file-name
```

Secure Shell For Remote Access

1. ssh
The following command refers to a cryptographic network protocol for operating network services securely over an unsecured network. Typical use-cases include remote command-line execution, but any network service can be secured with SSH.

This command, on running at the slave node, will give remote access to the master.

syntax:

```
$ ssh <master's ip>
```

This command, on running at the master, will give remote access to the slave node.

syntax:

```
$ ssh <slave's ip>
```