----------SK-Technologies----------

Contents:

1. **Unix Basic Commands----------**

Unix is not an Operating system; it is an Organization. Under Unix there are 4 flavors which are i. Aix

ii. Linux

iii. Solaris

iv. HP-UX

Linux:

There are multiple distributers available for Linux. RedHat(IBM now), CentOS, Ubuntu, Fedora, SeLinux …etc.

Each Distributer having their own infrastructure on the OS. Most of the commands are similar and working fine in their functionalities.

We will focus on the Linux commands why because in the Real time most of the application are running on the Linux servers only.

Small History of Linux:

Linux**, computer operating system created in the early 1990s by Finnish software engineer Linus Torvalds and the Free Software Foundation (FSF). in 1991 he released version 0.02; Version 1.0 of the Linux kernel, the core of the operating system, was released in 1994. About the same time, American software developer Richard Stallman and the FSF made efforts to create an open-source UNIX-like operating system called GNU. In contrast to Torvalds, Stallman and the FSF started by creating utilities for the operating system first. These utilities were then added to the Linux kernel to create a complete system called GNU/Linux, or, less precisely, just Linux.**

LINUX BASIC COMMANDS

Before we go on to the list of commands, you need to open the command line first. If you are still unsure about the command-line interface, check out this **CLI tutorial**. Although the steps may differ depending on the distribution that you're using, you can usually find the command line in the **Utilities** section.
Here is a list of basic Linux commands:

1. pwd command

Use the **pwd** command to find out the path of the current working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that starts with a forward slash **(/)**. An example of an absolute path is **/home/username**.
Syn: $pwd

2. uname command

The uname command which display the OS name of the server, like Linux, Aix or Solaris etc.
Syn: $uname

3. whoami command

The whoami commands which tell us the login session is running by which user.

Syn: $whoami

4. touch command

The **touch** command allows you to create a blank new file through the Linux command line. As an example, enter
touch **/home/username/Documents/Web.html** to create an HTML file entitled **Web** under the **Documents** directory.
Syn: $touch <filename>

5. ls command

The **ls** command is used to view the contents of a directory. By default, this command will display the contents of your current working directory.
If you want to see the content of other directories, type **ls** and then the directory's path. For example, enter **ls /home/username/Documents** to view the content of **Documents**.
There are variations you can use with the **ls** command:

- **ls -R** will list all the files in the sub-directories as well
- **ls -a** will show the hidden files
- **ls -al** will list the files and directories with detailed information like the permissions, size, owner, etc.

Syn: $ls

6. vi command

The vi command allows you to write something into the file. Also it is used to modify the data in the file.

Vi contains the "Insert" mode which means only special keys will work. Esc button plays crucial role here.

There are some basic keys, Esc+i -- Insertmode

Esc+:w -- save

Esc+:q -- quit

Esc+:wq -- save& quit

Esc+:w! -- Forcefully saving

Esc+:q! -- Forcefully quit

Esc+:wq! -- Forcefully save&quit

Syn: $vi <Filename>

7. cat command

cat (short for concatenate) is one of the most frequently used commands in Linux. It is used to list the contents of a file on the standard output (sdout). To run this command, type **cat** followed by the file's name and its extension. For instance: **cat file.txt**.
Here are other ways to use the **cat** command:

- **cat > filename** creates a new file
- **cat filename1 filename2>filename3** joins two files (1 and 2) and stores the output of them in a new file (3)
- to convert a file to upper or lower case use, **cat filename | tr a-z A-Z >output.txt**

Syn: $cat <Filename>

8. cp command

Use the **cp** command to copy files from the current directory to a different directory. For instance, the command **cp scenery.jpg /home/username/Pictures** would create a copy of **scenery.jpg** (from your current directory) into the **Pictures** directory.
Syn: $cp <sourcefile> <Destinationpath>

9. mv command

The primary use of the **mv** command is to move files, although it can also be used to rename files.
The arguments in mv are similar to the cp command. You need to type **mv**, the file's name, and the destination's directory. For example: **mv file.txt /home/username/Documents**.
To rename files, the Linux command is **mv oldname.ext newname.ext**

Syn: $mv <sourcefile> <Destinationpath>

10. mkdir command

Use **mkdir** command to make a new directory — if you type **mkdir Music** it will create a directory called **Music**.
There are extra **mkdir** commands as well:

- To generate a new directory inside another directory, use this Linux basic command **mkdir Music/Newfile**
- use the **p** (parents) option to create a directory in between two existing directories. For example, **mkdir -p Music/2020/Newfile** will create the new "2020" file.

Syn: $mkdir <Directory_Name>

11. cd command

To navigate through the Linux files and directories, use the **cd** command. It requires either the full path or the name of the directory, depending on the current working directory that you're in.
Let's say you're in **/home/username/Documents** and you want to go to **Photos**, a subdirectory of **Documents**. To do so, simply type the following command: **cd Photos**.
Another scenario is if you want to switch to a completely new directory, for example,**/home/username/Movies**. In this case, you have to type **cd** followed by the directory's absolute path: **cd /home/username/Movies**.
There are some shortcuts to help you navigate quickly:

- **cd ..** (with two dots) to move one directory up
- **cd** to go straight to the home folder
- **cd-** (with a hyphen) to move to your previous directory

On a side note, Linux's shell is case sensitive. So, you have to type the name's directory exactly as it is.

Syn: $cd

12. rmdir command

If you need to delete a directory, use the **rmdir** command. However, rmdir only allows you to delete empty directories.
Syn: $rmdir <Empty_Directoryname>

13. rm command

The **rm** command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to rmdir — use **rm -r**.
Note: Be very careful with this command and double-check which directory you are in. This will delete everything and there is no undo.
Syn: $rm <filename>
$rm -rf <Directoryname or filename>
Note: Be careful while using this command, we don't have recycle-bin concept in the Unix environments.

14. head command

The "head" command used to display the top 10 lines of the file content. Also, you can display the more than 10 lines by the option added it.
Syn: $head <filename>
$head -15 <filename> {top 15 lines it will display}

15. tail command

The "tail" command used to display the bottom 10 lines of the file content. Also, same as top command with option you can display multiple lines from bottom by applying the "-" option.
Syn: $tail <filename>
$tail -20 <filename>
How to display the running logs?
$tailf -100 <log_filename>
$tail -n 100 >log_filename>

16. more command

The “more” command display the content of the file as line by line. But there is a small draw back with this which we can’t move the page into upwards.
To over come this we can use the “less” command.

17. less command

The “less” command, as mentioned in the previous command to overcome the more we are using the less command. It can display the file content with line by line and we can move the content in both sides up & down.

18. date command

The “date” commands which prints the time stamp of the server. By using the date command, we can reset the time also.

19. grep command

The “grep’ command very regularly we are using in the realtime. By using it, we can trip the output of command. Lets take an example, $cat filename, it will display the complete content of the file. But if you want to display some specific portion of the content then you can use the “grep” command and you can append this to cat command.

Syn: $cat <filename> | grep <argument>

$cat <filename> | grep -i <argument> {to ignore the case sensitive}

20. ps -ef command

The “ps -ef” we regularly using to display the processes which are running in the background. It will display the process id, parent id, by which user it running and on which command based it is running all this info it showing.

Logic question: How many processes are running in the machine?

21. kill command

The “kill” used for deleting the processes. There are many sign to append with kill command. But most commonly using the -9 option.

Syn: $kill -9 <processID>.

22. useradd command

The “useradd” command very popular command which is used to create a users.

Make sure you would be a Super user/root user, By the normal user you can't execute the useradd command. Which means only root user can create users.

Syn: #useradd <username> {The moment you are creating a user the group also will create with the same name}

#id <username>

23. passwd command

By using the "passwd" command we can set or reset the password for user accounts.

Syn: #passwd <username>

24. systemctl command

This is very imp command. Regularly we are using it. By using this command we can change the state of the daemons/services. There is slight difference between service and process but both are running in the background.

There are many options available in the systemctl command each option having their own functionalities.

Syn: #systemctl {start/stop/restart/status/enable..etc} <servicename>

Note: all services are available under "/etc/systemd/system".

25. zip command

The "zip" command is very useful by using this we can make archiving the files which means compressing the files. By doing this we can reduce our actual file sizes.

$zip <filename> {sometimes you will get an error like zip command is not found which means in the background zip rpm is not running you have to install it by "#yum install zip"}

26. unzip command

The unzip command we are using for extracting the zipped files.

$unzip <file.zip>

27. gzip & gunzip command

This is also same as zip command. But the suffix will be vary. Normally the suffix is .gz

In the real-time most of the people are using the .gz files only.

$gzip <filename>

$gunzip <file.gz>

29. tar command

The “tar” command very useful, by using this we can make archive on the directories.

Syn: $tar -cvf <newname.tar> <directory_name>

$tar -xvf <file.tar>

30. wget command

By using this wget command we can download something from the browser.

$wget <URL>

31. yum commands

The “yum” command very imp, if you know yum then you can install/remove/update packages in Linux server. This command will works on Redhat, CentOS, Amzon Linux etc.

The following are the couple of yum commands:

Syn: #yum <options> <arguments>

#yum <install/remove/update/upgrade/….etc>

#yum <install/remove/update/upgrade/….etc> <packagename>

yum repolist

#yum list

#yum list <packagename>

#yum install <packageName>

#yum remove <packagename>

#yum clean all

#yum update

#yum update <packagename>

#yum info <packagename>

Note: The configuration file of yum is “/etc/yum.repos.d/base.repo

How to setup Repo-configuration?

Create or download .repo file under /etc/yum.repos.d/

Import the gpg key

yum clean all & yum repolist

What is the content of the .repo file?

#cat /etc/yum.repos.d/jenkins.repo

32. apt-get commands

apt is a command-line utility for installing, updating, removing, and otherwise managing deb packages on Ubuntu, Debian, and related Linux distributions. It combines the most frequently used commands from the apt-get and apt-cache tools with different default values of some options.

The apt or apt-get command is used to manage the packages in Ubuntu OS. Same as how we are performing with yum command the same functionalities it contains but in the Ubuntu operating system.

If you are running by normal user you can use the “prefix” as sudo.

Syn: $sudo apt update or $sudo apt-get update

$sudo apt-get upgrade

$sudo apt-get install <packagename>

$sudo apt-get remove <packagename>

$sudo apt auto remove <packagename>

$sudo apt list

$sudo apt list |grep -I <packagename>

$sudo apt list –installed

$sudo search <packagename>

$sudo apt show <packagename>

Note: the configuration files is “/etc/apt/apt.conf

How to list repo configuration files in Ubuntu?

`/etc/apt/sources.list` file and all files under `/etc/apt/sources.list.d/` directory.

33. rpm command

RPM (Red Hat Package Manager) is an default open source and most popular package management utility for Red Hat based systems like (RHEL, CentOS and Fedora). The tool allows system administrators and users to install, update, uninstall, query, verify and manage system software packages in Unix/Linux operating systems. The RPM formerly known as .rpm file, that includes compiled software programs and libraries needed by the packages. This utility only works with packages that built on .rpm format.

Syn: #rpm –checksig <packagename>

#rpm -ivh < packagename >

#rpm -qpR < packagename >

#rpm -ivh –nodeps < packagename >

#rpm -qa < packagename >

#rpm -qa |grep -i < packagename >

#rpm -Uvh < packagename >

#rpm -evv < packagename >

#rpm -ev –nodeps < packagename >

#rpm -qi < packagename>

34. wc command

The wc command which is used for count if you are specifying the options based on it it will count.

Syn: $wc -l <filename>

$wc -c <filename>

$wc -a <filename>

35. hostname command

The “hostname” command prints the server’s hostname.

Syn: $hostname

36. ifconfig command

The “ifconfig” command is regularly we are using it will display the IP address and few more info.

The popular command is “ifconfig -a”

Syn: $ifconfig -a

37. df -h command

The “df -h” command used to display the mounted file systems and its properties/utilizations details.

Syn: $df -h

38. du -sh command

This is most popular command every day we are using this command to check the files/directories size

Syn: $du -sh * {it means it will display the sizes of all files & directories which are in present directory}