



## All Basic/Advanced Linux Commands for Linux Administrators Interview Questions PDF

### Contents [\[hide\]](#)

- 1 40 Basic Linux Commands
- 2 Important Linux Commands
- 3 Learn Linux Commands
- 4 Useful Linux Commands
- 5 UNIX Basic Commands
- 6 About PuTTY
- 7 40 Basic Linux Commands List
- 8 File System Structure in Linux
- 9 Linux Commands for Beginners
- 10 What is Linux Commands?
  - 10.1 Manipulational Related Linux Commands
  - 10.2 File System Space Monitoring Linux Commands
  - 10.3 User Management Linux Commands
  - 10.4 Performance Management Linux Commands
- 11 Linux Commands Cheatsheet
- 12 Navigational Linux Commands
  - 12.1 man command in Linux
  - 12.2 pwd command in Linux
  - 12.3 cd command in Linux
  - 12.4 ls command in Linux
  - 12.5 locate command in Linux
  - 12.6 find command in Linux
  - 12.7 grep command in Linux
  - 12.8 cat command in Linux
  - 12.9 clear command in Linux
  - 12.10 echo command in Linux
- 13 Manipulational Linux Commands
  - 13.1 touch command in Linux
  - 13.2 vi command in Linux
  - 13.3 mkdir command in Linux
  - 13.4 rmdir command in Linux
  - 13.5 rm command in Linux
  - 13.6 cp command in Linux
  - 13.7 mv command in Linux
- 14 File System Space Monitoring Linux Commands
  - 14.1 du command in Linux
  - 14.2 df command in Linux
  - 14.3 head command in Linux
  - 14.4 tail command in Linux
- 15 User Management Related Linux Commands
  - 15.1 useradd command in Linux
  - 15.2 userdel command in Linux
  - 15.3 usermod command in Linux
  - 15.4 passwd command in Linux
  - 15.5 chage command in Linux
  - 15.6 groupadd command in Linux
  - 15.7 groupdel command in Linux
  - 15.8 sudo command in Linux
  - 15.9 chmod command in Linux
  - 15.10 chown command in Linux
  - 15.11 diff command in Linux
  - 15.12 sdiff command in Linux



### Categories

### Recent Posts

[How to increase shared memory in RHEL7](#)

[Best Linux Apps for Every Kind of User](#)

[What is Linux ?](#)

[How to use help command in linux](#)

[Netbackup Tutorials : Troubleshooting Tips to resolve backup failures](#)

### Sign for Latest Updates

#### First Name

#### Last Name

#### Email address:

- 15.13 tar command in Linux
- 15.14 gzip and gunzip command in Linux
- 16 Performance Management Based Linux Commands
  - 16.1 top command in Linux
  - 16.2 ps command in Linux
  - 16.3 kill command in Linux
  - 16.4 free command in Linux
- 17 CONCLUSION

### Share on Social Media



### Follow us on Social Media



## 40 Basic Linux Commands

The general assumption when someone hears about Linux is that all users think of a very complicated operating system and it is mostly used by developers or programmers. But it's not as complicated as it is assumed. To break that myth I am going to share **basic linux commands** in this article. I will try to show in this article that how easy to memorize these **basic linux commands** as they exactly have similar names with their functionality.

### Important Linux Commands

Linux is basically derived from Unix but it is absolutely open-source Unix Operating System that is known as Linux Kernel. There are many popular Linux-based systems like Fedora, Ubuntu, Mint, Debian, CentOS, and many more proprietary Linux based systems like Red Hat Linux and SUSE, etc. As you might be aware Linux was released in **1991** first time and it is on expansion to date. Linux became so much popular due to its open-source nature. Anyone is free to modify and redistribute it under their own brand name.

### Learn Linux Commands

Organizations like Red Hat taken full advantage of its open-source feature and they have developed their own Linux which has become world-famous and popularly known as Red Hat Enterprise Linux and they have their development environment popularly known as Fedora. Same as many other organizations have used and developed Proprietary Linux and established as big brands globally.

### Useful Linux Commands

The general idea about all the Linux users is that all the users who log in to the Linux system must need to have a shell. By default, all Linux users have a bash shell. Shell is basically an interface between user and Linux systems. Shell provides an interface that gives access to the operating system's services. Most of the Linux distributions have a graphic user interface as their shell which is for making Linux easy for their users.

While it has a GUI interface but CLI Command Line Interface is too much popular among all Linux users as it is more powerful and very much effective.

### UNIX Basic Commands

Simply you can say if any task requires many steps through GUI which can be done in a matter of seconds by typing Linux commands via CLI.

So if you are going to use Linux or already started using Linux for any reason either for a job or for reading or for business or anything learning or going through these Linux

#### Storage

LINUX RHEL

### How to increase shared memory in RHEL7

Raj Rai January 11, 2021

### Best Linux Apps for Every Kind of User

RSY Digital World January 8, 2021

### What is

### Linux ?

Raj Rai January 8, 2021

#### Net Backup

LINUX RHEL

### How to increase shared memory in RHEL7

Raj Rai January 11, 2021

### Best Linux Apps for Every Kind of User

RSY Digital World January 8, 2021

### What is

### Linux ?

Raj Rai January 8, 2021

Commands give you a fair idea of how to accomplish any specific task and these Linux commands are going to help you in a long way.

## **About PuTTY**

PuTTY is open-source as well and delivered as freeware useful for SSH and telnet. Putty is being used by all business organizations and many global organizations keep it in their desktop/laptop build as inbuilt tools.

Anyhow if you do not have it on your desktop or laptop you can simply [download PuTTY](#) here.

One of the most important things to consider as a Linux user either a newbie or an experienced user you need to have logs of your activity. Shell is even having the option of history but if you have logged it is always better and provides you safety in many odd situations.

You can refer to the process of [how to configure putty to save the log](#).

## **40 Basic Linux Commands List**

We will try to share a list of the *40 basic Linux commands* in a single go to allow you to better understand and have more familiarity with Linux. Even though each Linux command is too descriptive and has many switches to use.

All switches can not be covered in a single post. I hope you must agree with that but we will try our best to show sample output as much as possible.

The general concept about Linux is that there are many ways to accomplish the same tasks via various methods and all yield the same output or results you can same.

You will find minute differences in different distributions. If you have a fair idea about anyone's distribution you will have more idea of how you can use man pages which will be your best friend to get the exact syntax.

Basically, commands are nothing but scripts which is already compiled and incorporated with operating systems. While creating Linux commands human readability and memorability have been taken into consideration to make it easy to remember for users.

## **File System Structure in Linux**

If you have some knowledge about the file system structure of any Linux related distribution. As mentioned earlier all commands are compiled scripts with Linux Operating System. In technical terms, they are called binaries.

Most of the Linux distributions prefer RBAC principles. RBAC stands for Role-Based Access Control which is very important for the safety of the Linux system and keeps the operating system integrity intact.

Based on RBAC they are normal users of the Linux system who logs into the system and perform exactly what they are assigned for and can not make any changes to the Linux system.

## **Linux Commands for Beginners**

Another hand there are power users like UNIX system administrators or Linux system administrators who are supposed to make changes to the Linux system according to the need of organizations.

## **What is Linux Commands?**

Linux commands are pre-compiled bash scripts delivered with Operating System Bundle. Based on that binaries are divided into two basic categories. There is always a directory

called /usr which stands for users which is common for all categories of users to Linux systems. all binaries are kept under a short form of binaries termed as a bin.

All pre-compiled binaries for normal users kept under /usr/bin directory and all the special privileges like system administrators binaries which are able to make changes to Linux systems are kept under /usr/sbin. s stands for special and a bin is the same binaries.

We will discuss this later while going through commands about what special means. Technically it is called privilege.

I can sense it is going to be a big read but do not worry. It will help you out to familiarize yourself with 40 important Linux commands you must know and which is going to allow you to work on any Linux or even UNIX-based systems either it is Desktop Level or Server Level.

Now let us go through **40 basic Linux commands**. I am going to start with a man command. See how easy to remember it for men or anyone. It can be also categorized as below.

**Navigational Linux Commands**

**Manipulational Related Linux Commands**

**File System Space Monitoring Linux Commands**

**User Management Linux Commands**

**Performance Management Linux Commands**

## **Linux Commands Cheatsheet**

*Important Linux commands we are going to share in these posts are as below.*

```
man  pwd  cd   ls   locate find  grep  cat
clear echo  touch vi  mkdir rmdir rm   cp
mv    du   df   head tail diff  sdiff tar
passwd chage sudo  chmod chown top   ps   kill
free  gunzip useradd userdel usermod groupadd
groupdel
```

### **Navigational Linux Commands**

#### **man command in Linux**

**man command in Linux** refers to the word manual. Normally we all know what a manual consists of. It has all the details about the products that's why different items have different manuals. The same is the role of man command in Linux. the man command is used to have details about particular Linux command.

Syntax of man command is

```
$ man <linux commands>
$ man ls
```

Always manual is too much descriptive as it will show all the options you can avail from that **Linux commands**. It is not feasible to copy here considering the objective of this share. You can simply type the details in your Linux system to see all the details.

## ***pwd* command in Linux**

***PWD* command in Linux** stands for the Present Working Directory. As the name suggests ***PWD* command** is used to find out the path in the Linux system where currently you are.

Normally if you just logged into any Linux system you must be there in your home folder. So if you run the ***PWD*** Linux command system will print the present working directory means your home folder path.

```
$ pwd
/export/home/ramesh
```

## ***cd* command in Linux**

***cd* command in Linux** where stands for Change Directory. So if you want to navigate into Linux system files and folders you can use the ***CD*** Linux command to navigate.

You can use the ***cd*** command in a different way. Just remember while navigation that Linux is case sensitive operating system. So you always need to consider this while navigating.

```
$ cd /path
$ cd /var
$ pwd
/var
$ cd /           - It will change to / root directory.
$ cd ../        - It will change you to one directory back.
$ cd ../tmp     - It will go one directory back and go to another directory
                in one go.
```

If you want to see all the options with a ***cd*** or any other commands we are going to share through this share you can use the ***man*** command.

## ***ls* command in Linux**

***ls* command in Linux** is related to the word list. It is used for listing the contents of any directory. It provides similar functionality to DIR in DOS-based systems.

This is having too much switch to using with ***ls*** Linux command. If you simply use the ***ls*** command so wherever you are present it will show the contents of that directory. You can refer to the ***PWD*** command to find out where exactly you are in the directory structure of [Linux](#) systems.

```
$ ls
```

So if you have just logged in and have not done anything it means you are in your home folder so a simple type of ***ls*** command will display the contents of your home folder.

As I said it is having too many switches and it is not possible to discuss all the switches. One write up will not be enough for the ***ls*** command itself and we have to discuss all the basic commands here. So we will mention only a few switches here.

If you have been used Windows Based Systems either Desktop or Laptop or Even Servers any version starting from DOS to Latest Windows Versions there is always one feature in windows called hidden files. Similar concepts are also available in Linux System as well.

There are some sensitive files that need prevention normally kept under this category so by default they are kept as hidden files. You need to enable view hidden files.

-a -a switch you can understand it as all including hidden files as well.

-R -R switch is relative to recursive which means it will include directory and subdirectory as well.

-l -l switch is related to long listing views of files and folders in directory.

-h -h is related to human readable format.

-t -t switch will shows the timestamp of file last changed.

You can used these switches in combinations or individually based on your query you have.

```
$ ls -l
```

```
$ ls -lt
```

```
$ ls -ltr
```

```
$ ls -ltrh
```

```
$ ls -lart
```

## **locate command in Linux**

**locate command in Linux** is your real best friend to locate what is the path of any commands in the Linux system if you do not remember. It is really helpful in case if you do not know where exactly binary exists if you find any error like "command not found".

As the name suggests it locates the path of files in the Linux system. This is really helpful as it is not found in UNIX systems.

The general syntax of the **locate** is like below.

```
$ locate <filename>
```

```
$ locate vgs
```

```
/sbin/vgscan
```

```
/usr/sbin/vgs
```

```
/usr/sbin/vgscan
```

```
/usr/sbin/vgsplit
```

```
/usr/share/man/man8/vgs.8.gz
```

```
/usr/share/man/man8/vgscan.8.gz
```

```
/usr/share/man/man8/vgsplit.8.gz
```

## **find command in Linux**

**find command in Linux** is another very useful command to find the files in the Linux system. As the name suggests find **Linux commands** are used to find files. This has the same objective as the **locate** command. find is also available in UNIX systems.

These are really very very helpful if you have to find out files that are consuming more space in case of file system full or in case of file change investigations like incident.

Again you can see all the options with find using man command as mentioned above. This also available with too many switch and you have to familiarize more with the find command. It

can be used in conjunction with other so-called dangerous commands of UNIX or Linux like `rm` we will show details of `rm` down the lane.

#### Few Examples of `find` command

```
$ find /tmp -name core -type f -print | xargs /bin/rm -f
```

This command will do two function first it will find files named `core` in `/tmp` directory and then deletes it. This you can use for your house keeping purpose to have `/tmp` clean in form of scheduled task to keep your system healthy.

```
$ find $HOME -mtime 0
```

This will find files in your home directory which got modified within 0 days means last twenty-four hours.

It is so vast. Just do man and exercise them very carefully if you are using with `rm` especially otherwise it will only fetch details without changing anything if no `rm` conjunction used.

### *grep command in Linux*

`grep` stands for global regular expression print. As the name suggests **grep command in Linux** is used to catch specific regular expressions from a file. This is very useful in the daily life of Linux users and also heavily used in automation scripts to direct output and performing specific operations on segregated output of `grep`.

```
$ grep red students.txt
```

For example, students have been marked in red and green. Failed students marked red and passed students marked green.

You can use the above command to have a list of students who have failed without touching or updating the existing file and perform activity whatever needed.

### *cat command in Linux*

`cat` is an abbreviation for concatenating. **cat command in Linux** is the most frequently used commands in Linux. It lists the contents of a file on the standard output.

Syntax of the `cat` command is like below

```
$ cat <filename>
```

It is also very handy for concatenation as its name suggest. It redirects output to another location for performing any operations without touching original file for any work.

```
$ cat /etc/passwd > /tmp/passwd.txt
```

It is also very useful during file system full troubleshooting to nullify any file which is the reason of file system full like below. If you have backup you can nullify file to gain space related to `/var` file system full.

```
$ cat /dev/null > messages
```

### *clear command in Linux*

As the name suggests it clears the terminal screen. As you know Linux is terminal-based and sometimes you continuously work on the terminal and it becomes difficult to track the output of any *Linux commands* you may have run. In that scenario, the *clear command* is *Linux* becomes really handy.

```
$ clear
```

>>This will clear your screen and output will go to top of terminal.

## **echo command in Linux**

As the name suggests *echo command in Linux* echoes means its display and it is used to move data to file.

```
$ echo Hello; My name is SRK >> SRK.txt
```

So here it echo's Words and updates SRK.txt file.

```
$ echo $PATH
```

```
/export/opt/oracle
```

If PATH variable is set correctly I mean defined it will echo's the value of path.

## **Manipulational Linux Commands**

### **touch command in Linux**

*touch command in Linux* is used to touch or you can say blank file. For example, if you have created any file system and to make sure all looks good with the file system and it is writable. You can use the *touch* command to touch the file system and create a blank file to test the writability to the file system.

```
$ touch test
```

If you are able to create a file that confirms the file system is writable otherwise the command will fail and will show write error kind of error.

So basically touch is used for testing the writability of either existing or new file system as the name suggests it touch file system and test it.

### **vi command in Linux**

*vi command in Linux* is basically abbreviated from the visual editor which is the default editor delivered with Linux operating system. It is basically an editor that can be used to create a new file and do the stuff via that file.vi is a full-screen editor and has two modes. Two modes are command mode and insert mode.

```
$ vi <filename>
```

If file exists it will open that file and if it does not exists it will open new file which you can edit and save as per your requirement.

To save file you need to use exit sequence like below

```
<Esc>:q! -- To exit from file without saving.
```



```
<Esc>: wq! -- To exit from file with save option.
```

There are many other options you can learn in vi editors once you start practicing. Normally people find it a little bit difficult and there are many other popular editors that can be used. But believe me, once you do your hand gets dirty with vi editors you will find it really easy.

## **mkdir command in Linux**

As the name suggests **mkdir command in Linux** used to [create a directory in Linux](#). mkdir resembles from make and directory word. So **mkdir command in Linux** used to create a directory. You must be aware of what is a directory.

The folder where you keep the same category or related category files and subfolders is known as the directory. A directory can hold files and a directory which is called a subdirectory in case if it exists under any parent directory. Everything under UNIX or Linux system has one parent directory called root directory and written as /.

```
$ sudo mkdir /ramesh  
$ sudo mkdir /ramesh/suresh
```

In this parent directory for Ramesh and Suresh is / and Ramesh is a parent directory for Suresh while Suresh is a subdirectory of Ramesh. It is also termed a child directory.

## **rmdir command in Linux**

Again **rmdir command**, where rm stands for remove and dir, stands for the directory. **rmdir command in Linux** is safer command compared to rm commands which we will see below because rmdir commands can remove the only blank directory.

```
$ sudo rmdir <directory name>
```

## **rm command in Linux**

As you have seen rmdir is incapable to remove a directory that has contents there here it comes to the rm devil which is also known as [rm command in Linux](#) which is one of the **most dangerous commands in Linux** and to be used very wisely. Extra space between **rm** command may lead to disaster and end up with the removal of entire contents.

It has the capability of deleting files and folders also. You can use syntax like below.

```
$ sudo rm <filename> - It removes mentioned file  
  
$ sudo rm -r <directory name> - It removed mentioned directory including  
subdirectory and files as well in directory and  
subdirectory.  
  
$ sudo rm -rf <directory> -If you are sure what you are intended to delete -f  
option also can be used which refers to forcibly  
delete entire directory structure mentioned  
with command means same again directory, file  
and subdirectory and files under sub directory  
which means entire directory structure.
```

## **cp command in Linux**

**cp command in Linux** is a very much used command in the everyday life of Linux users. cp is derived from word copy. cp command is used to copy files from the source directory to the target directory.

```
$ sudo cp /ramesh/test.txt /suresh/
```

Mentioning whole path is also known as absolute path in unix linux world . These also have variant like it can copy all the contents from one directory to another.

```
$ sudo cp /ramesh/* /suresh
```

\* represents all in ramesh directory. It gets copied to suresh directory but it still remains in suresh directory .That implies it does not remove source and make a copy of source contents to another location.

```
$ sudo cp -pr /ramesh /suresh
```

This will also copy entire contents of ramesh directory to suresh and have ramesh directory unchanged but it will inherit parent permissions.

## mv command in Linux

**mv command in Linux** also *very important Linux commands* almost similar to cp but the difference is that **cp command** leaves copy intact at source but **mv command** does not leave files and folders at source it simply moves to the new location.

```
$ sudo mv /ramesh/test.txt /suresh
```

It will move test.txt file to suresh directory and test.txt will be no more in ramesh directory it will gets moved permanently to suresh directory.

So that is the main difference in cp and mv commands you can realize cp leave duplicate content and mv not. So cp will waste your space while mv not.

**mv command in linux** you can also use to rename the file or folder which will have syntax like below

```
$ sudo mv file1 file2
```

```
$ sudo mv dir1 dir2
```

## File System Space Monitoring Linux Commands

### du command in Linux

**du command in Linux** is derived from disk usage. the **du command** is used to find how much space is used by the directory and files.

By default du output shows in bytes and it is difficult to understand or you need to do calculations to find out the space used in KB and MB to have a clear idea you can use the **-sh** option which shows output in human-readable format and is easy to understand. It will show Kilobyte, Megabyte & even Gigabyte.

```
$ du -sh
12K  account
34M  cache
```

```
230M log
```

```
2.0G www
```

```
$
```

If you are troubleshooting file system full issue and exactly looking some big files in GB or MB which is filling file system it becomes very handy with dig down file or folder using space quickly.

```
$ cd /directory which have alerts
```

```
$ du -sh *|egrep "M|G"
```

This will only show files in MB or GB in that directory by navigating same way you can easily file the culprit one.

## df command in Linux

**df command in Linux** is derived from disk free. df command is used to see the reports of the file system how much free space you have. Again you need to use the -h option to see the report in a human-readable format.

You can refer to the below output. It will display in an easy to understand report. Which can be used for capacity planning as if a business needs file system expansion. Also, file systems always get monitored and the threshold is getting defined to be notified or alerted via incidents in organizations to keep them healthy and continue business as usual.

```
$ df -h
```

| Filesystem        | Size | Used | Avail | Use% | Mounted on |
|-------------------|------|------|-------|------|------------|
| /dev/cciss/c0d0p3 | 9.7G | 6.0G | 3.3G  | 65%  | /          |
| /dev/cciss/c0d0p5 | 24G  | 14G  | 8.2G  | 64%  | /var       |
| /dev/cciss/c0d0p1 | 170M | 38M  | 124M  | 24%  | /boot      |
| tmpfs             | 5.9G | 8.0K | 5.9G  | 1%   | /dev/shm   |

```
$
```

## head command in Linux

**head command in Linux** you can symbolize with topmost. head command is used to display the top lines of any text file and by default, it displays the first ten lines.

```
$ head messages
```

Even you can use switch to display number of lines as per your requirements using syntax below.

To see top 5 lines from messages file.

```
$ head -n 5 messages
```

## tail command in Linux

tail command in Linux is similar to head but tail reports from down and head reports from up. This also displays the last 10 lines.

```
$ tail -n messages
```

These are really handy in case of troubleshooting any issues and to understand better what is going wrong with linux or unix system.

```
$ tail -20 messages
```

-It will display last 20 messages so you can understand if any issue going on with your linux system.

## User Management Related Linux Commands

User Management in Linux is one of the crucial parts of Linux system administrators. Nowadays many organizations have a separate IAM team which is called Identity Access Management.

This is because of the complexity of access level and better management of users in the organization and use of various techniques like LDAP based authentication and AD-based authentication NIS based Authentication Local System based Authentication. IAM is almost an industry.

We are intended here with CLI and related **basic Linux commands** so we will be focused on Local System Based Authentication and related **basic Linux commands**.

There are mainly 4 files in Linux or UNIX systems that hold user details. Files are passwd, shadow, group, nsswitch.conf.

### useradd command in Linux

**useradd command in Linux** as it is clearly evident from the command name that it is intended to add users to the Linux or UNIX system.

When a useradd command runs successfully it adds user details in passwd file.

The syntax you can use to add users in one go properly with all options is like below. You can use it in a simple way also and it takes the default option but this syntax will clarify all the things in a single go and you can co-relate it with passwd file entry and parameters in passwd file.

```
$ sudo useradd -u UID -g GID GECOS Home Folder Shell
```

UID : Unique ID ----All users must have unique ID.

GID : Group ID ----All users must have unique group ID.

GECOS: About user details like Name or Reference Number if any

Home Folder : home folder for user where he lands into the system.

Shell : What shell user have by default bash shell for user.

### userdel command in Linux

**userdel command in Linux** as the name suggest userdel command is used for deleting users. the syntax for the **userdel** command is like below.

```
$ sudo userdel <username>
```

### usermod command in Linux

**usermod command in Linux** is another user management command which is used for changing or modifying user configuration. Even though there are many modifications for the user that can be done by **usermod**. Just have a man page to refer to all the modifications you can make using **usermod command in Linux**.

For example you can unlock user account using

```
$ sudo usermod -U <user id>
```

## passwd command in Linux

As the name suggests [passwd command in Linux](#) is used to set the password for the user, unlocking the user account, reset the password for the user, setting a non-expiring password, and many more.

**passwd command in Linux** modifies /etc/shadow file which stores all the credentials in encrypted form so you need to have root-level access for password management.

While a normal user can set his password but if you have to reset, unexpired, or any other password related stuff you need privilege access. It is also named a superuser.

Few examples of **passwd command in linux** is as below.

```
$ passwd <my id>           - Change password for yourself.
$ sudo passwd <user name>  - Change password for others.
$ sudo passwd -l <user id> - Locking password for user.
$ sudo passwd -u <user id> - Unlock user password.
```

## chage command in Linux

**chage command in Linux** is another user management related command called chage. It is a combination of Change and age. **chage command in Linux** is used to manage user's password validity related information's like current password expiry information [change password in Linux](#) expiry date Change last password change date set password expiry notice to the user and how to lock the account and many more.

Let us understand by some examples.

If you simply type **chage command** you will have like this which tells you that what all you can do with **chage command**.

```
$ sudo chage
Usage: chage [options] user
Options:
-d, --lastday LAST_DAY    set last password change to LAST_DAY
-E, --expiredate EXPIRE_DATE set account expiration date to EXPIRE_DATE
-h, --help                display this help message and exit
-I, --inactive INACTIVE  set password inactive after expiration to INACTIVE
-l, --list                show account aging information
-m, --mindays MIN_DAYS   set minimum number of days before password
-M, --maxdays MAX_DAYS  set maximum number of days before password
-W, --warndays WARN_DAYS set expiration warning days to WARN_DAYS
```

```
$ sudo chage -l sybase
Last password change           : Jul 24, 2020
Password expires               : never
Password inactive              : never
Account expires                : never
Minimum number of days between password change : -1
Maximum number of days between password change : -1
Number of days of warning before password expires : -1
```

```
$ sudo chage -M 99999 sybase
```

It will change maximum number of days between password change

## groupadd command in Linux

As the name suggests **groupadd command in Linux** is used for creating groups. The syntax for the **groupadd** command in Linux is as below

```
$ sudo groupadd [options] <group name>
```

groupadd command in linux writes details in /etc/group file.

## groupdel command in Linux

**groupdel command in Linux** is just for reversing group and deletes them and syntax is similar to groupadd.

```
$ sudo groupdel [options] <GROUP NAME>
```

## sudo command in Linux

**sudo command in Linux** is derived from two words su stands for superuser and do mean what superuser is doing. It is pronounced as a sudo in [Linux and UNIX](#). Exactly you can say sudo means superuser do. This command provides you administrative or root privilege. Mean whatever superuser can do I can also do.

It can be used with all system modifying commands. Like below

```
$ sudo cp /ramesh/ /suresh
```

## chmod command in Linux

**chmod command in Linux** is another Linux command derived from two words ch stands for change and mod stands for modifying which clearly says **chmod command** objectives.

There are three categories of users defined in Linux or the UNIX system and they are called user, group, and others.

There are three Permission is defined as read, write, and execute. There is some numerical value assigned to each permissions read=4 write=2 execute=1 and these are applicable to files as well as the directory.

Will try to summarize with the help of few examples

```
$ ls -l /etc/passwd
```

```
-r--r--r-- 1 root root 2376 Jan 28 2017 /etc/passwd
```

If you see the permission for password file in above output it is mentioned and marked in red they are for user, group and others and all have only read permission. So considering read value if you want to give only read permission to all for any file you can assign (u=4g=4o=4) means if you combine it it looks like 444.

```
$ sudo chmod 444 /etc/passwd
```

```
$ ls -l /var/log/messages
```

```
-rw----- 1 root root 2553995 Oct 18 00:08 /var/log/messages
```

In this example see the permission first three it is -rw and rest is - means only user have read and write permission group and others do not have any kind of permissions on it. So this you can achieve using  $u=4+2+0,0+0+0,0+0+0$  ie 600.

```
$ sudo chmod 600 /var/log/messages
```

Let us see another example. This we will understand from directory.

```
$ ls -ld /tmp
```

```
drwxrwxrwt 14 root root 8151040 Oct 18 00:11 /tmp
```

First letter starts from d which represents that it is a directory. Read next nine letters it is in group of three rwx and they represent for user group and others. For now forget about last t. So it means users group and others all have read write and execute permissions. So if you calculate  $4+2+1,4+2+1,4+2+1$  which you will get at 777. This can be achieved on any file or folder like below.

```
$ sudo chmod 777 /tmp
```

Curious what is t in the last ? OK it is called sticky bit which is used to provide more [security](#) to world writable files in linux system which prevents deletions of files in directory.

## chown command in Linux

**chown command in Linux** is another Linux command related to permission chown. As the name suggests ch change and own=owner. **chown command** is getting used for changing ownership.

For example, consider the HR department of an organization that is headed by Ramesh. They have their own HR-related folder on the server.

```
$ ls -ld /export/home/payout
drwxr-xr-x 3 ramesh HR 4096 /export/home/payout
```

So you can see here it is a directory with 755 permission as we discussed in **chmod command** and directory is owned by Ramesh and the group is HR. So Ramesh can read and write execute while the HR group can read and execute and others can read-only.

So think of Ramesh got retired and Suresh is onboarded and takes the change of the department and you have to incorporate the change in the folder which will be like below.

```
$ sudo chown suresh:HR /export/home/payout
$ ls -ld /export/home/payout
drwxr-xr-x 3 ramesh HR 4096 /export/home/payout
```

One more thing to add about **chown command in Linux** will change ownership of directory but it will not change the ownership of files and subdirectory inside. If you have to change it across the entire directory structure you have to use the -R option which means recursively.

```
$ sudo chown -R suresh:HR /export/home/payout
```

## diff command in Linux

**diff command in Linux** is the short name for difference. **diff command in Linux** is used to find the difference in two files.

Syntax of **diff command in linux** is as below

```
$ diff file1 file2
```

For example if you have changes IP or name of your linux system and you have saved hosts file backup before save. You can run below command to see the difference.

```
$ sudo diff /etc/hosts /etc/hosts.bkup
```

If any difference is there in file it will report if there is no output it simply means files are identical.

So diff command if focused on finding difference only.

## sdiff command in Linux

**sdiff command in Linux** is another important **Linux commands** for comparison of two files is sdiff command in Linux comparison of two files. It provides better readability as compared to diff command in Linux which only prints differences while sdiff will print both files in compare mode which will be easy to see if any difference.

```
$ sdiff hosts hosts.181020
```

```
# Do not remove the following line          # Do not remove the following line
# that require network functionality will fail. # that require network functionality will fail.
127.0.0.1      localhost.localdomain localhost  127.0.0.1  localhost.localdomain localhost
210.210.176.19 linuxq16 linuxq16.abcd.com    210.210.176.29 linuxq16 linuxq16.abcd.com
210.210.180.52 linuxpuredisk1 linuxpuredisk1-nest<
```

The output looks very much clear with sdiff and really good for file comparison to understand what is the difference between the two files.

## tar command in Linux

**the tar command in Linux** is mostly used command. The full form of tar command is the **Tape Archive.tar command in Linux** is used to archive multiple files and folders into a single file. The created file is popularly known as a tarball in the Linux and UNIX world. Further zip tools can be used to compress it. It becomes really helpful to save space in the system.

For example

If you have any folder which is not in use and consuming too much space and you need some space in file system you can create tar ball and compress it and keep it for use later if needed in future and you can take off entire directory



structure. So you can say it works like backup and restore tool as well.

There is many option available but mostly used options is creating tar ball and extracting tar ball like below.

```
$ sudo tar cvf /data/olddata.tar /data/olddata
```

Where c stand for copy v stands for verbose and f file source where you want to keep tar ball and target directory for which we want to archive.

**How to extract tar file in linux ?**

```
$ cd to directory where you want to extract and run below
```

```
$ sudo tar xvf olddata.tar
```

## gzip and gunzip command in Linux

**gzip and gunzip command in Linux** is another very good and frequently used Linux command utility used for compressing files to save disk space.

Consider the tar command example once you create a tar file you can zip it using gzip to save space. gzip is a zipping tool like WinZip.

```
$ sudo gzip /data/olddata.tar
```

It will create file with .gz extension. Later you can unzip that whenever needed.

```
$ sudo gunzip /data/olddata.tar.gz
```

## Performance Management Based Linux Commands

### top command in Linux

**top command in Linux** is one of the most popular and heavily used processes monitoring tools in form of **Linux commands**. It provides an overall display of programs running and resources in terms of CPU Memory Load being used. You will have a very good idea about the Linux system that is exactly happening.

The **top command** comes with a variety of options you need to read it and practice it to become familiar with the **top utility in Linux**.

```
$ top
```

```
top - 12:54:21 up 179 days, 17:31, 21 users, load average: 0.06, 0.08, 0.08
```

```
Tasks: 322 total, 1 running, 320 sleeping, 1 stopped, 0 zombie
```

```
Cpu(s): 0.5%us, 1.0%sy, 0.0%ni, 98.5%id, 0.0%wa, 0.0%hi, 0.0%si, 0.0%st
```

```
Mem: 12170084k total, 11971828k used, 198256k free, 148816k buffers
```

```
Swap: 35405424k total, 1068136k used, 34337288k free, 665496k cached
```

```
PID USER PR NI VIRT RES SHR S %CPU %MEM TIME+ COMMAND
```

```
18242 root 18 0 26400 1564 472 S 2.3 0.0 495:22.57 cmaferfd
```

```
19967 root  15  0 384m 22m 8604 S 2.3 0.2 6214:40 BESClient
13270 root  15  0 19448 1484 1004 R 1.0 0.0 0:00.11 top
3501 root   11 -5  0  0  0 S 0.3 0.0 1421:14 vxfs_thread
9265 root  15  0  0  0  0 S 0.3 0.0 0:18.32 pdflush
  1 root   15  0 10376 504 472 S 0.0 0.0 0:53.24 init
  2 root   RT -5  0  0  0 S 0.0 0.0 2:19.46 migration/0
  3 root   34 19  0  0  0 S 0.0 0.0 2:39.85 ksoftirqd/0
  4 root   RT -5  0  0  0 S 0.0 0.0 0:00.00 watchdog/0
  5 root   RT -5  0  0  0 S 0.0 0.0 2:33.28 migration/1
  6 root   34 19  0  0  0 S 0.0 0.0 0:40.05 ksoftirqd/1
  7 root   RT -5  0  0  0 S 0.0 0.0 0:00.00 watchdog/1
  8 root   RT -5  0  0  0 S 0.0 0.0 1:06.07 migration/2
  9 root   34 19  0  0  0 S 0.0 0.0 4:23.08 ksoftirqd/2
10 root   RT -5  0  0  0 S 0.0 0.0 0:00.00 watchdog/2
11 root   RT -5  0  0  0 S 0.0 0.0 1:15.92 migration/3
12 root   34 19  0  0  0 S 0.0 0.0 1:05.78 ksoftirqd/3
13 root   RT -5  0  0  0 S 0.0 0.0 0:00.00 watchdog/3
14 root   10 -5  0  0  0 S 0.0 0.0 0:08.66 events/0
15 root   10 -5  0  0  0 S 0.0 0.0 0:08.26 events/1
16 root   10 -5  0  0  0 S 0.0 0.0 0:18.82 events/2
17 root   10 -5  0  0  0 S 0.0 0.0 0:16.72 events/3
18 root   10 -5  0  0  0 S 0.0 0.0 0:00.29 khelper
121 root   10 -5  0  0  0 S 0.0 0.0 0:00.92 kthread
129 root   10 -5  0  0  0 S 0.0 0.0 0:25.41 kblockd/0
130 root   10 -5  0  0  0 S 0.0 0.0 6:48.29 kblockd/1
131 root   10 -5  0  0  0 S 0.0 0.0 2:26.26 kblockd/2
132 root   10 -5  0  0  0 S 0.0 0.0 2:16.48 kblockd/3
133 root   16 -5  0  0  0 S 0.0 0.0 0:00.00 kacpid
310 root   16 -5  0  0  0 S 0.0 0.0 0:00.00 cqueue/0
```

```
$
```

## ps command in Linux

**ps command in Linux** is used to report a snapshot of the current processes running on the system. It is really handy to find out the process which is really consuming too many resources during the performance issue investigation.

Again **ps command in Linux** also has too many switches. Again you can have a look at all options using man pages for ps.

```
$ man ps
```

Few very much used options are as below. Simple **ps** will give details about yourself what process you are running with your terminal details and PID as well.

```
$ ps
```

```
PID TTY      TIME CMD
615 pts/22  00:00:00 root
25014 pts/22  00:00:00 ps
```

```
$ ps -ef
```

It will provide extended list of processes with full path of programs with PID and user who is running that program.

```
$ ps -ef |grep root
```

You can use **grep** to sort process with specific user.

```
$ ps -eo comm,pcpu --sort -pcpu | head -5
```

```
COMMAND      %CPU
dataserver    2.9
BESClient     2.4
dataserver    2.2
dataserver    0.9
```

This will provide you top 5 process CPU consumers in easily understandable report. This is from quite cool server nothing serious on it right now but it will really help to find out by which you can find out

## kill command in Linux

**kill command in Linux** is a really helpful command to keep the Linux system in good health. If the Linux system not performing well or going through sluggish performance. You need to find the culprit process using other **Linux commands** and once you are aware of the process you can kill the process using a Process ID called PID to restore the Linux system back to the normal state.

If the Linux system has any unresponsive or defunct process you can terminate them manually using the **kill command in Linux**. Kill has more than 50 signals but two signals are mostly used 9 and 15. Singal 9 is known as SIGKILL and signal 15 is known as SIGTERM.

Linux system has provisions to keep the system healthy and it also sends a certain signal to problematic applications and tells them to terminate themselves to have a system in a good state.

```
$ sudo kill -9 <PID>
```

## free command in Linux

**free command in Linux** is one of the most used commands to have details about how much memory is used and free in Linux systems. **free command in Linux** definitely deserves a place in this list of 40 **Important Linux commands** you should know.

Sample output you can see below. By default without any option it reports in KB and the same output by using `free -k`.

```
$ free
Total      used      free      shared  buffers  cached
Mem:    12170084 11948676  221408    0      144492  649952
-/+ buffers/cache: 11154232 1015852
Swap:   35405424 1068052  34337372

$ free -k
total      used      free      shared  buffers  cached
Mem:    12170084 11947884  222200    0      144512  649964
-/+ buffers/cache: 11153408 1016676
Swap:   35405424 1068052  34337372

$ free -m
total      used      free      shared  buffers  cached
Mem:    11884    11668    216     0       141     634
-/+ buffers/cache: 10892     991
Swap:    34575    1043    33532

$ free -g
total      used      free      shared  buffers  cached
Mem:         11     11     0     0     0     0
-/+ buffers/cache:     10     0
Swap:        33     1     32
```

## CONCLUSION

All the mentioned *Important Linux commands* above are really *basic & advanced useful linux commands* and getting used by Linux administrators as well as normal Linux users DevOps for their routine works. I hope you will like this collection of *40 basic linux commands for beginners and advanced users also*.

I sincerely thank you for going to this collection of the best and most used *basic linux commands for beginners and advanced users*.

If you think any commands missed in this list that deserves space in this collection feel free to comment in the comment section and share details for the benefit of readers of this blog.

It will really help our readers to have additional information in form of other *important linux commands* which may make their life even easier.

### Share on Social Media



### Follow us on Social Media



40 basic Linux commands • advanced linux commands pdf • all linux commands • all linux commands pdf • basic linux commands • basic linux commands list • basic linux commands pdf • basic linux interview questions • basic unix commands • best linux commands you must know • cheat sheet linux • command in linux • common linux commands • important linux commands • in linux commands • interview questions on linux commands • learn linux commands • linux admin interview questions • linux administrator interview questions • linux advanced commands pdf • linux all commands • linux basic commands • linux basic commands pdf • linux basics interview questions • linux cheat sheet • linux command • linux command cheat sheet • linux command interview questions • linux command list • linux command pdf • linux commands • linux commands cheat sheet • Linux Commands cheat sheet pdf • linux Commands for beginners • Linux Commands for devops • linux Commands interview questions • linux commands list • linux commands list pdf • linux Commands pdf • linux commands tutorial • Linux Commands with examples • Linux Commands with examples and syntax • Linux Commands with examples and syntax pdf • linux commands with syntax and examples pdf • linux interview questions • linux interview questions and answers • linux interview questions commands • list of linux commands • Manipulation Related Linux Commands • unix basic commands • unix basic commands pdf • unix command interview questions • unix commands • useful linux commands • vi linux commands • what is in linux • what is in linux command