**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**.

**2. 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.

**3. 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.

**4. 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**

**5. 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.

**6. 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**

**7. 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.

**8. rmdir command**

If you need to delete a directory, use the **rmdir** command. However, rmdir only allows you to delete empty directories.

**9. 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.

**10. 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.

**11. locate command**

You can use this command to **locate** a file, just like the search command in Windows. What’s more, using the **-i** argument along with this command will make it case-insensitive, so you can search for a file even if you don’t remember its exact name.

To search for a file that contains two or more words, use an asterisk **(\*)**. For example, **locate -i school\*note** command will search for any file that contains the word “school” and “note”, whether it is uppercase or lowercase.

**12. find command**

Similar to the **locate** command, using **find** also searches for files and directories. The difference is, you use the **find** command to locate files within a given directory.

As an example, find **/home/ -name notes.txt** command will search for a file called **notes.txt** within the home directory and its subdirectories.

Other variations when using the **find** are:

* To find files in the current directory use, **find . -name notes.txt**
* To look for directories use, **/ -type d -name notes. txt**

**13. grep command**

Another basic Linux command that is undoubtedly helpful for everyday use is **grep**. It lets you search through all the text in a given file.

**14. sudo command**

Short for “**SuperUser Do**”, this command enables you to perform tasks that require administrative or root permissions. However, it is not advisable to use this command for daily use because it might be easy for an error to occur if you did something wrong.

**15. df command**

Use **df** command to get a report on the system’s disk space usage, shown in percentage and KBs. If you want to see the report in megabytes, type **df -m**.

**16. du command**

If you want to check how much space a file or a directory takes, the **du** (Disk Usage) command is the answer. However, the disk usage summary will show disk block numbers instead of the usual size format. If you want to see it in bytes, kilobytes, and megabytes, add the **-h** argument to the command line.

**17. head command**

The**head** command is used to view the first lines of any text file. By default, it will show the first ten lines, but you can change this number to your liking. For example, if you only want to show the first five lines, type **head -n 5 filename.ext**.

**18. tail command**

This one has a similar function to the head command, but instead of showing the first lines, the **tail** command will display the last ten lines of a text file. For example, **tail -n filename.ext.**

**19. diff command**

Short for difference, the **diff** command compares the contents of two files line by line. After analyzing the files, it will output the lines that do not match. Programmers often use this command when they need to make program alterations instead of rewriting the entire source code.

The simplest form of this command is **diff file1.ext file2.ext**

**20. tar command**

The **tar** command is the most used command to archive multiple files into a **tarball** — a common Linux file format that is similar to zip format, with compression being optional.

This command is quite complex with a long list of functions such as adding new files into an existing archive, listing the content of an archive, extracting the content from an archive, and many more.

**21. chmod command**

**chmod** is another Linux command, used to change the read, write, and execute permissions of files and directories.

**22. chown command**

In Linux, all files are owned by a specific user. The **chown** command enables you to change or transfer the ownership of a file to the specified username. For instance, **chown linuxuser2 file.ext** will make **linuxuser2** as the owner of the **file.ext**.

**23. jobs command**

**jobs** command will display all current jobs along with their statuses. A job is basically a process that is started by the shell.

**24. kill command**

If you have an unresponsive program, you can terminate it manually by using the **kill** command. It will send a certain signal to the misbehaving app and instructs the app to terminate itself.

There is a total of 64 signals that you can use, but people usually only use two signals:

* **SIGTERM (15)** — requests a program to stop running and gives it some time to save all of its progress. If you don’t specify the signal when entering the kill command, this signal will be used.
* **SIGKILL (9)** — forces programs to stop immediately. Unsaved progress will be lost.

Besides knowing the signals, you also need to know the process identification number (PID) of the program you want to **kill**. If you don’t know the PID, simply run the command **ps ux**.

After knowing what signal you want to use and the PID of the program, enter the following syntax:

**kill [signal option] PID**.

**25. ping command**

Use the **ping** command to check your connectivity status to a server. For example, by simply entering **ping google.com**, the command will check whether you’re able to connect to Google and also measure the response time.

**26. wget command**

The Linux command line is super useful — you can even download files from the internet with the help of the **wget** command. To do so, simply type **wget** followed by the download link.

**27. uname command**

The **uname** command, short for Unix Name, will print detailed information about your Linux system like the machine name, operating system, kernel, and so on.

**28. top command**

As a terminal equivalent to Task Manager in Windows, the **top** command will display a list of running processes and how much CPU each process uses. It’s very useful to monitor system resource usage, especially knowing which process needs to be terminated because it consumes too many resources.

**29. history command**

When you’ve been using Linux for a certain period of time, you’ll quickly notice that you can run hundreds of commands every day. As such, running **history** command is particularly useful if you want to review the commands you’ve entered before.

**30. man command**

Confused about the function of certain Linux commands? Don’t worry, you can easily learn how to use them right from Linux’s shell by using the **man** command. For instance, entering **man tail** will show the manual instruction of the tail command.

**31. echo command**

This command is used to move some data into a file. For example, if you want to add the text, “Hello, my name is John” into a file called name.txt, you would type **echo Hello, my name is John** **>> name.txt**

**32. zip, unzip command**

Use the **zip** command to compress your files into a zip archive, and use the **unzip** command to extract the zipped files from a zip archive.

**33. hostname command**

If you want to know the name of your host/network simply type **hostname**. Adding a **-I** to the end will display the IP address of your network.

**34. useradd, userdel command**

Since Linux is a multi-user system, this means more than one person can interact with the same system at the same time. **useradd** is used to create a new user, while **passwd** is adding a password to that user’s account. To add a new person named John type, **useradd John** and then to add his password type, **passwd 123456789.**

To remove a user is very similar to adding a new user. To delete the users account type, **userdel UserName**

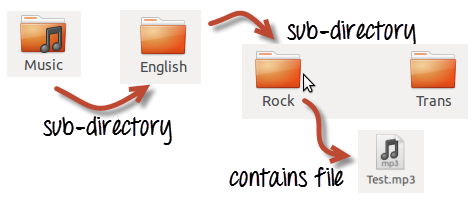
## Listing files (ls)

If you want to see the list of files on your UNIX or Linux system, use the '**ls'** command.  
  
It shows the files /directories in your current directory.  

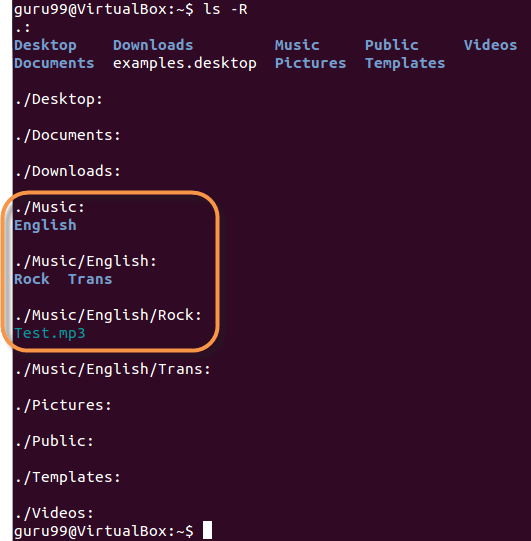

Note:

* Directories are denoted in blue color.
* Files are denoted in white.
* You will find similar color schemes in different flavors of Linux.

Suppose, your "Music" folder has following sub-directories and files.



You can use **'ls -R' to shows all the files not only in directories but also subdirectories**

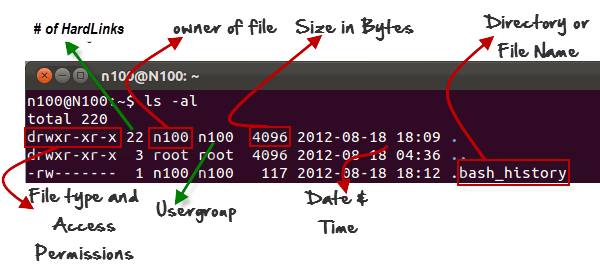


NOTE: The command is case-sensitive. If you enter, "**ls - r**" you will get an error.

**'ls -al'** gives detailed information of the files. The command provides information in a columnar format. The columns contain the following information:

|  |  |
| --- | --- |
| **1st Column** | **File type and access permissions** |
| **2nd Column** | # of HardLinks to the File |
| **3rd Column** | Owner and the creator of the file |
| **4th Column** | Group of the owner |
| **5th Column** | File size in Bytes |
| **6th Column** | Date and Time |
| **7th Column** | Directory or File name |

Let's see an example -

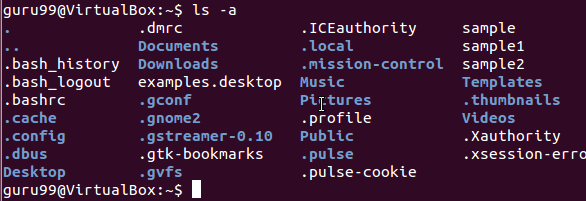


## Listing Hidden Files

Hidden items in UNIX/Linux begin with -[Must Know Linux/Unix Commands](https://www.guru99.com/images/period_symbol(2).png) at the start, of the file or directory.

Any Directory/file starting with a '.' will not be seen unless you request for it.  To view hidden files, use the command.

ls -a

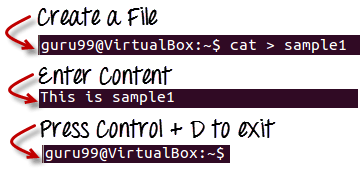


## Creating & Viewing Files

The 'cat' server command is used to display text files. It can also be used for copying, combining and creating new text files.  Let's see how it works.

To create a new file, use the command

1. cat > filename
2. Add content
3. Press 'ctrl + d' to return to command prompt.



How to create and view files in Linux/Unix

To view a file, use the command -

cat filename

Let's see the file we just created -

Creating a file using Linux/Unix Commands

Let's see another file sample2

Creating a file using Linux/Unix Commands

The syntax to combine 2 files is -

cat file1 file2 > newfilename

Let's combine sample 1 and sample 2.

[Creating a file using Linux/Unix Commands](https://www.guru99.com/images/cat_combine.png)

As soon as you insert this command and hit enter, the files are concatenated, but you do not see a result. This is because **Bash Shell (Terminal) is silent type**.  Shell Commands will never give you a confirmation message like "OK" or "Command Successfully Executed". It will only show a message when something goes wrong or when an error has occurred.

To view the new combo file "sample" use the command

cat sample

[Viewing a file using Linux/Unix Commands](https://www.guru99.com/images/cat_combo.png)

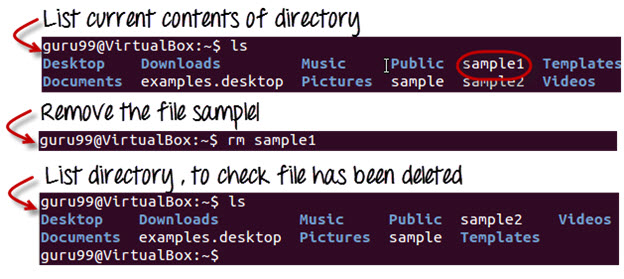
**Note:**Only text files can be displayed and combined using this command.

## Deleting Files

The 'rm' command removes files from the system without confirmation.

To remove a file use syntax -

rm filename

How to delete files using Linux/Unix Commands

## Moving and Re-naming files

To move a file, use the command.

mv filename new\_file\_location

Suppose we want to move the file "sample2" to location /home/guru99/Documents. Executing the command

***mv sample2 /home/guru99/Documents***

How to move a file using Linux/Unix Commands

mv command needs super user permission. Currently, we are executing the command as a standard user. Hence we get the above error. To overcome the error use command.

sudo command\_you\_want\_to\_execute

Sudo program allows regular users to run programs with the security privileges of the superuser or root.

Sudo command will ask for password authentication. Though, you do not need to know the root password. You can supply your own password. After authentication, the system will invoke the requested command.

Sudo maintains a log of each command run. System administrators can trackback the person responsible for undesirable changes in the system.

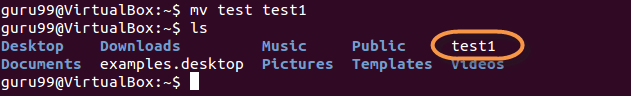
guru99@VirtualBox:~$ sudo mv sample2 /home/quru99/Documents

[sudo] password for guru99: \*\*\*\*

guru99@VirtualBox:~$

For renaming file:

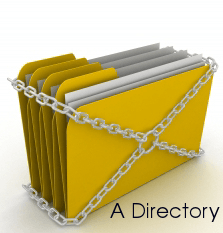
mv filename newfilename



**NOTE**: By default, the password you entered for sudo is retained for 15 minutes per terminal. This eliminates the need of entering the password time and again.

You only need root/sudo privileges, only if the command involves files or directories not owned by the user or group running the commands

## Directory Manipulations

Directory Manipulation in Linux/Unix

Enough with File manipulations! Let's learn some directory manipulation Linux basic commands.

Creating Directories

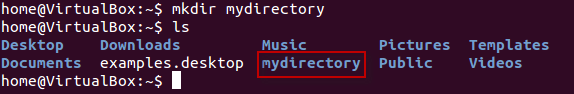
Directories can be created on a Linux operating system using the following command

mkdir directoryname

This command will create a subdirectory in your present working directory, which is usually your "Home Directory".

For example,

mkdir mydirectory



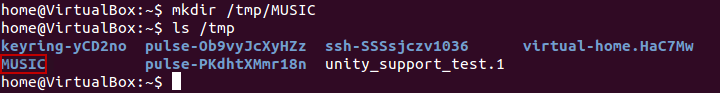
If you want to create a directory in a different location other than 'Home directory', you could use the following command -

mkdir

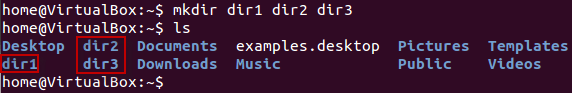
For example:

mkdir /tmp/MUSIC

will create a directory 'Music' under '/tmp' directory



You can also create more than one directory at a time.



## Removing Directories

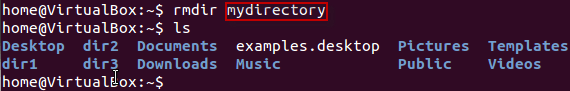
To remove a directory, use the command -

rmdir directoryname

Example

rmdir mydirectory

will delete the directory mydirectory



**Tip**: Ensure that there is no file / sub-directory under the directory that you want to delete. Delete the files/sub-directory first before deleting the parent directory.

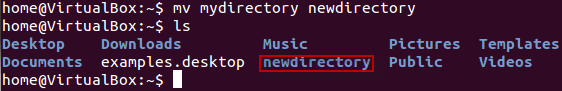
Directory Manipulation in Linux/Unix

## Renaming Directory

The 'mv' (move) command (covered earlier) can also be used for renaming directories. Use the below-given format:

mv directoryname newdirectoryname

Let us try it:



How to rename a directory using Linux/Unix Commands

## ****Other Important Commands****

## The 'Man' command

Man stands for manual which is a reference book of a Linux Operating System. It is similar to HELP file found in popular software.

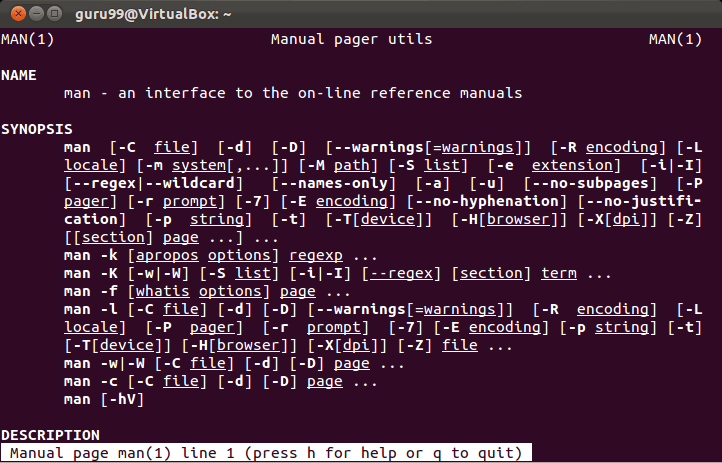
To get help on any command that you do not understand, you can type

man

The terminal would open the manual page for that command.

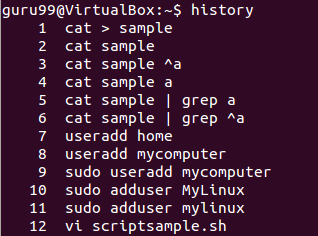
For an example, if we type man man and hit enter; terminal would give us information on man command

Man command in Linux/Unix



## The History Command

History command shows all the basic commands in Linux that you have used in the past for the current terminal session. This can help you refer to the old commands you have entered and re-used them in your operations again.



## ****The clear command****

This command clears all the clutter on the terminal and gives you a clean window to work on, just like when you launch the terminal.



|  |
| --- |
|  |

## Linux Command List

Below is a Cheat Sheet of Linux commands list we have learned in this Linux commands tutorial

|  |  |
| --- | --- |
| **Command** | **Description** |
| ls | Lists all files and directories in the present working directory |
| ls - R | Lists files in sub-directories as well |
| ls - a | Lists hidden files as well |
| ls - al | Lists files and directories with detailed information like permissions, size, owner, etc. |
| cat > filename | Creates a new file |
| cat filename | Displays the file content |
| cat file1 file2 > file3 | Joins two files (file1, file2) and stores the output in a new file (file3) |
| mv  file "new file path" | Moves the files to the new location |
| mv filename new\_file\_name | Renames the file to a new filename |
| sudo | Allows regular users to run programs with the security privileges of the superuser or root |
| rm filename | Deletes a file |
| man | Gives help information on a command |
| history | Gives a list of all past basic Linux commands list typed in the current terminal session |
| clear | Clears the terminal |
| mkdir directoryname | Creates a new directory in the present working directory or a at the specified path |
| rmdir | Deletes a directory |
| mv | Renames a directory |
|  |  |
|  |
|  |  |

**Removing files and directories (rm command)**

To remove files and directories, use the [rm](https://linuxize.com/post/rm-command-in-linux/) command.

By default, when executed without any option, rm doesn’t remove directories. It also doesn’t prompt the user for whether to proceed with the removal of the given files.

To delete a file or a symlink, use the rm command followed by the file name as an argument:

rm file.txt

rm accepts one or more file or directory names as its arguments.

The -i option tells rm to prompt the user for each given file before removing it:

rm -i file.txt

rm: remove regular empty file 'file.txt'?

Use the -d option to remove one or more empty directories:

rm -d dirname

To remove non-empty directories and all the files within them recursively, use the -r (recursive) option:

rm -rf dirname

The -f option tells rm never to prompt the user and to ignore nonexistent files and arguments.

**Copying files and directories (cp command)**

The [cp](https://linuxize.com/post/rm-command-in-linux/) command allows you to copy files and directories.

To copy a file in the current working directory, use the source file as a first argument and the new file as the second:

cp file file\_backup

To copy a file to another directory, specify the absolute or the relative path to the destination directory. When only the directory name is specified as a destination, the copied file will have the same name as the original file.

cp file.txt /backup

By default, if the destination file exists, it will be overwritten.

To copy a directory, including all its files and subdirectories, use the -R or -r option:

cp -R Pictures /opt/backup

**Moving and renaming files and directories (mv command)**

The [mv](https://linuxize.com/post/how-to-move-files-in-linux-with-mv-command/) command (short from move) is used to rename and move and files and directories from one location to another.

For example, to move a file to a directory, you would run:

mv file.txt /tmp

To rename a file you need to specify the destination file name:

mv file.txt file1.txt

The syntax for moving directories is the same as when moving files.

To move multiple files and directories at once, specify the destination directory as the last argument:

mv file.tx1 file1.txt /tmp

## File Ownership and Permissions

In Linux, access to the files is managed through file permissions, attributes, and ownership. This ensures that only authorized users and processes can access files and directories.

In Linux, each file is associated with an owner and a group and assigned with permission access rights for three different classes of users:

* The file owner.
* The group members.
* Everybody else.

Three permissions types apply to each class:

* The read permission.
* The write permission.
* The execute permission.

This concept allows you to specify which users can read the file, write to the file, or execute the file.

To view the file owner and permissions, use the ls -l command.

### Changing permissions (chmod command)

The [chmod](https://linuxize.com/post/chmod-command-in-linux/) command allows you to change the file permissions. It works in two modes, symbolic and numeric.

When using the numeric mode, you can set the permissions for the owner, group, and all others. Each write, read, and execute permissions have the following number value:

* r (read) = 4
* w (write) = 2
* x (execute) = 1
* no permissions = 0

The permissions number of a specific user class is represented by the sum of the values of the permissions for that group.

For example, to give the file’s owner read and write permissions and only read permissions to group members and all other users, you would run:

chmod 644 filename

Only root, the file owner, or user with sudo privileges can change the permissions of a file.

To recursively operate on all files and directories under a given directory, use the chmod command with the -R, (–recursive) option:

chmod -R 755 dirname

Be extra careful when recursively changing the files’ permissions.

### Changing ownership (chown command)

The chown command allows you to change the user and group ownership of a given file, directory, or symbolic link.

To change the owner of a file, use the chown command followed by the user name of the new owner and the target file:

chown username filename

To change both the owner and the group of a file, invoke the chown command followed by the new owner and group separated by a colon (:) with no intervening spaces and the target file:

chown username:groupname filename

Use the -R (--recursive) option, to recursively operate on all files and directories under the given directory:

chown -R username:groupname dirname

## Managing Users and Groups

Linux is a multi-user system, which means that more than one person can interact with the same system at the same time. Groups are used to organize and administer user accounts. The primary purpose of groups is to define a set of privileges such as reading, writing, or executing permission for a given resource shared among the users within the group.

### Creating users (useradd and passwd Commands)

The  useradd command allows you can create new users.

To create a new user account, invoke the useradd command followed by the username:

useradd newuser

Once the user is created, set the user password by running the passwd command:

passwd newuser

### Removing users (userdel Command)

In Linux, you can delete a user account using the  userdel command.

To delete a user account named pass the user name to the userdel command:

userdel newuser

Use the -r (–remove) option to remove the user’s home directory and mail spool:

userdel -r newuser

### Managing groups (groupadd and groupdel Command)

To create a new group, invoke the groupadd command followed by the group name:

groupadd mygroup

To remove a group, use the groupdel command with the group name as argument:

groupdel mygroup

### Adding users to groups (usermod Command)

To add an existing user to a group, use the  usermod command followed by the -G option and the name of the group:

usermod -a -G sudo linuxize

Usually, your hard disk drive divided into one or more logical disks called partitions. This division is described in the partition table found in sector 0 of the hard disk. The device is usually /dev/sda, /dev/sdb or so on. A device name refers to the entire disk, and the device name will be as follows:

1. /dev/hd\* – IDE disks. /dev/hda will be first IDE hard disk, /dev/hdb will be second IDE hard disk, and so on.
2. /dev/sd\* – SCSI or SATA disks including SSDs. /dev/sda will be first SATA/SCSI hard disk, /dev/sdb will be second SATA/SCSI hard disk, and so on.
3. /dev/nvme\* – NVM Express (NVMe) pci SSD. /dev/nvme0n1 will be first NVMe SSD, /dev/nvme1n1 will be second NVMe SSD, and so on.

**1. View all Disk Partitions in Linux**

The following basic command list all existing disk partition on your system. The ‘**-l**‘ argument stand for (listing all partitions) is used with fdisk command to view all available partitions on Linux. The partitions are displayed by their device’s names. For example: **/dev/sda**, **/dev/sdb** or **/dev/sdc**.

[root@tecmint.com ~]# fdisk -l

Disk /dev/sda: 637.8 GB, 637802643456 bytes

255 heads, 63 sectors/track, 77541 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/sda1 \* 1 13 104391 83 Linux

/dev/sda2 14 2624 20972857+ 83 Linux

/dev/sda3 2625 4582 15727635 83 Linux

/dev/sda4 4583 77541 586043167+ 5 Extended

/dev/sda5 4583 5887 10482381 83 Linux

/dev/sda6 5888 7192 10482381 83 Linux

/dev/sda7 7193 7845 5245191 83 Linux

/dev/sda8 7846 8367 4192933+ 82 Linux swap / Solaris

/dev/sda9 8368 77541 555640123+ 8e Linux LVM

**2. View Specific Disk Partition in Linux**

To view all partitions of specific hard disk use the option ‘**-l**‘ with device name. For example, the following command will display all disk partitions of device **/dev/sda**. If you’ve different device names, simple write device name as **/dev/sdb** or **/dev/sdc**.

[root@tecmint.com ~]# fdisk -l /dev/sda

Disk /dev/sda: 637.8 GB, 637802643456 bytes

255 heads, 63 sectors/track, 77541 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/sda1 \* 1 13 104391 83 Linux

/dev/sda2 14 2624 20972857+ 83 Linux

/dev/sda3 2625 4582 15727635 83 Linux

/dev/sda4 4583 77541 586043167+ 5 Extended

/dev/sda5 4583 5887 10482381 83 Linux

/dev/sda6 5888 7192 10482381 83 Linux

/dev/sda7 7193 7845 5245191 83 Linux

/dev/sda8 7846 8367 4192933+ 82 Linux swap / Solaris

/dev/sda9 8368 77541 555640123+ 8e Linux LVM

**3. Check all Available fdisk Commands**

If you would like to view all commands which are available for fdisk. Simply use the following command by mentioning the hard disk name such as **/dev/sda**as shown below. The following command will give you output similar to below.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help):

Type ‘**m**‘ to see the list of all available commands of fdisk which can be operated on **/dev/sda** hard disk. After, I enter ‘**m**‘ on the screen, you will see the all available options for fdisk that you can be used on the **/dev/sda** device.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): **m**

Command action

a toggle a bootable flag

b edit bsd disklabel

c toggle the dos compatibility flag

d delete a partition

l list known partition types

m print this menu

n add a new partition

o create a new empty DOS partition table

p print the partition table

q quit without saving changes

s create a new empty Sun disklabel

t change a partition's system id

u change display/entry units

v verify the partition table

w write table to disk and exit

x extra functionality (experts only)

Command (m for help):

**4. Print all Partition Table in Linux**

To print all partition table of hard disk, you must be on command mode of specific hard disk say **/dev/sda**.

[root@tecmint ~]# fdisk /dev/sda

From the command mode, enter ‘**p**‘ instead of ‘**m**‘ as we did earlier. As I enter ‘**p**‘, it will print the specific **/dev/sda** partition table.

Command (m for help): **p**

Disk /dev/sda: 637.8 GB, 637802643456 bytes

255 heads, 63 sectors/track, 77541 cylinders

Units = cylinders of 16065 \* 512 = 8225280 bytes

Device Boot Start End Blocks Id System

/dev/sda1 \* 1 13 104391 83 Linux

/dev/sda2 14 2624 20972857+ 83 Linux

/dev/sda3 2625 4582 15727635 83 Linux

/dev/sda4 4583 77541 586043167+ 5 Extended

/dev/sda5 4583 5887 10482381 83 Linux

/dev/sda6 5888 7192 10482381 83 Linux

/dev/sda7 7193 7845 5245191 83 Linux

/dev/sda8 7846 8367 4192933+ 82 Linux swap / Solaris

/dev/sda9 8368 77541 555640123+ 8e Linux LVM

Command (m for help):

**5. How to Delete a Partition in Linux**

If you would like to delete a specific partition (i.e **/dev/sda9**) from the specific hard disk such as **/dev/sda**. You must be in fdisk command mode to do this.

[root@tecmint ~]# fdisk /dev/sda

Next, enter ‘**d**‘ to delete any given partition name from the system. As I enter ‘**d**‘, it will prompt me to enter partition number that I want to delete from **/dev/sda** hard disk. Suppose I enter number ‘**4**‘ here, then it will delete partition number ‘**4**‘ (i.e. **/dev/sda4**) disk and shows free space in partition table. Enter ‘**w**‘ to write table to disk and exit after making new alterations to partition table. The new changes would only take place after next reboot of system. This can be easily understood from the below output.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): **d**

Partition number (1-4): **4**

Command (m for help): **w**

The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.

The kernel still uses the old table. The new table will be used at

the next reboot or after you run partprobe(8) or kpartx(8)

Syncing disks.

You have new mail in /var/spool/mail/root

**Warning** : Be careful, while performing this step, because using option ‘**d**‘ will completely delete partition from system and may lost all data in partition.

**6. How to Create a New Partition in Linux**

If you’ve free space left on one of your device say **/dev/sda** and would like to create a new partition under it. Then you must be in fdisk command mode of **/dev/sda**. Type the following command to enter into command mode of specific hard disk.

[root@tecmint ~]# fdisk /dev/sda

After entering in command mode, now press “**n**” command to create a new partition under **/dev/sda** with specific size. This can be demonstrated with the help of following given output.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): **n**

Command action

e extended

p primary partition (1-4)

**e**

While creating a new partition, it will ask you two options ‘**extended**‘ or ‘**primary**‘ partition creation. Press ‘**e**‘ for extended partition and ‘**p**‘ for primary partition. Then it will ask you to enter following two inputs.

1. First cylinder number of the partition to be create.
2. Last cylinder number of the partition to be created (Last cylinder, +cylinders or +size).

You can enter the size of cylinder by adding “**+5000M**” in last cylinder. Here, ‘**+**‘ means addition and **5000M** means size of new partition (i.e **5000MB**). Please keep in mind that after creating a new partition, you should run ‘**w**‘ command to alter and save new changes to partition table and finally reboot your system to verify newly created partition.

Command (m for help): **w**

The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.

The kernel still uses the old table. The new table will be used at

the next reboot or after you run partprobe(8) or kpartx(8)

Syncing disks.

**7. How to Format a Partition in Linux**

After the new partition is created, don’t skip to format the newly created partition using ‘**mkfs**‘ command. Type the following command in the terminal to format a partition. Here **/dev/sda4** is my newly created partition.

[root@tecmint ~]# mkfs.ext4 /dev/sda4

**8. How to Check Size of a Partition in Linux**

After formatting new partition, check the size of that partition using flag ‘**s**‘ (displays size in blocks) with fdisk command. This way you can check size of any specific device.

[root@tecmint ~]# fdisk -s /dev/sda2

5194304

**9. How to Fix Partition Table Order**

If you’ve deleted a logical partition and again recreated it, you might notice ‘**partition out of order**‘ problem or error message like ‘**Partition table entries are not in disk order**‘.

For example, when three logical partitions such as (**sda4**, **sda5** and **sda6**) are deleted, and new partition created, you might expect the new partition name would be **sda4**. But, the system would create it as **sda5**. This happens because of, after the partition are deleted, **sda7** partition had been moved as **sda4** and free space shift to the end.

To fix such partition order problems, and assign **sda4** to the newly created partition, issue the ‘**x**‘ to enter an extra functionality section and then enter ‘**f**‘ expert command to fix the order of partition table as shown below.

[root@tecmint ~]# fdisk /dev/sda

WARNING: DOS-compatible mode is deprecated. It's strongly recommended to

switch off the mode (command 'c') and change display units to

sectors (command 'u').

Command (m for help): **x**

Expert command (m for help): **f**

Done.

Expert command (m for help): **w**

The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.

The kernel still uses the old table. The new table will be used at

the next reboot or after you run partprobe(8) or kpartx(8)

Syncing disks.

After, running ‘**f**‘ command, don’t forget to run ‘**w**‘ command to save and exit from fdisk command mode. Once it fixed partition table order, you will no longer get error messages.