**Basic Linux Commands for beginners**

* #sudo su (super user do - switch user)
* #cat (concatenate) -> list the content of file into the terminal.
* CTRL+d (EXIT SHORT KEY)
* #hostname
* #ifconfig
* #cat /etc/os-release
* #ln -s (soft link)
* #ln( hard link backup)
* #chkconfig http on
* #chown (change the owner and group owner of a file).
* #curl (Retrieve information from URL)
* #df (Show size, used space, and available space) df –h –x
* #diff (compare two txt files)
* #echo (print a string into terminal)
* #exit
* #find (track down file that you know exist)
* #free (summary of the memory)
* #grep (Used for searching) 
* #groups(tells you which groups a user is a member of)
* #gzip (compress the file)
* #head (give first 10 lines of the file)
* #history (pervious commands)
* #kill(terminate the process)
* #less (view file without opening an editor)
* #ls (list the files and folders in the current directory)
* #mkdir (create new directories in the file system)
* #mv (move file and directory from directory to directory)
* #passwd (use for the change of password)
* #ping (verify that you have network connectivity)
* #ps(running process)
* #pwd(print working directory)
* #shutdown
* #ssh(make a connection to remote linux computer)
* #tail(last 10 lines of the file)
* #w (currently logged user)
* #who (info about logged user)
* #uname (tell about system info)
* #touch (create new empty file) + update the timestamp of the file (access time, modify time and change time) #stat file name
* #cp (used copy a file into directory)
* #wc(used to count a words in text file)
* #more (control the display od output)
* #wget(used to download web page)
* #ps (current running process with PID)-🡪 ps –aux | grep ftp🡪 To capture specific process
* #netstat –atnp | grep 3306 🡪 To capture Specific Port Number
* #passwd 🡪 Change Password
* #more filename (To Read Big Files)
* #kill -9 PID 🡪 Forcefully Kill Process
* #gzip folder/filename 🡪 Compress file/folder (Not Recommended Use Tar to avoid mistakes)
* #unzip –l foldername.zip (Read without uncompressing)
* #init 0 (Shutdown)
* #init 6 (Restart)
* #shutdown –h now
* #shutdown –h +20
* #shutdown –r now
* #tr (translating and deleting characters)
* #sort
* #useradd
* #groupadd
* #gpasswd -a -M(multiple user add into group)
* #cat /etc/passwd # /etc/group
* #cd (Change Directory)

**Linux Most Useful Commands According to Our Working Environment**

* #pwd (Print present working directory)
* #free –g 🡪 Check Free RAM
* #top--- press 1---CPU Usage Check
* #df –h 🡪 Hard disk Usage Check
* #fdisk –l 🡪 To Check Hard disk attached
* #mount 🡪 Mount Disk
* #whereis cat 🡪 To check location of command like CAT
* #/etc/sudoers---Remove last All 🡪 Revoke sudo access and give specific command to execute as sudo
* #ls –altrh (drwxr-xr-x   2 root root  20K Jul  2 15:56 bin) 🡪 List files with most information
* #cd (directory,..,../..) 🡪 Change Directory
* #man ls 🡪 Show Manual of any command for usage
* #cp file1 file2 🡪 Copy File
* #mv file1 file2 🡪 Move File or Rename
* #mkdir foldername 🡪 Create Folder
* #chmod –R user-group-other 🡪 Change Permission
* #date 🡪 Current Date
* #file file1 (Type of file)
* #tar –cvzf foldername.tar.gz 🡪 Create backup and can be used as zip (Recommanded)
* Grep (Search)
* #ssh user@ip –p portnumber 🡪 SSH session
* #telnet abc.com 443 🡪 Check Port connectivity
* #rm –rf folder/file (Most dangerous command)
* #whatis cat (Give 1 line description)
* #tail –n N filename (N number of lines) 🡪 Check From Bottom of file
* #head –n N 🡪 Check From Start of file
* #find 🡪 To find some file or even text in file
* #wget 🡪 TO Download Web Content
* ----------------------------------------------------------------------
* Shell Login Screen [username@machinename:~] $/# 🡪 When you login the screen it gives few useful information like username(username) from which it is logged in, Machine Name(machinename), Directory(~) in which it is logged in and $/# represent the type of user for $ its standard user and # means it’s a root user.
* #chmod(Set the file permissions) chmod –R 777. (owner+group+world)(nopermission+Execute+Write+WE+Read+RE+RW+RWE)

**Linux File Permissions**

In Linux, access to the files is controlled by the operating system using file permissions, attributes, and ownership. Understanding the Linux file system permissions model allows you to restrict access to files and directories only to authorized users and processes and make your system more secure.

Each file is owned by a particular user and a group and assigned with permission access rights for three different classes of users:

* The file owner.
* The group members.
* Others (everybody else).

There are three file permissions types that apply to each user class and allows you to specify which users are allowed to read the file, write to the file, or execute the file. The same permission attributes apply for both files and directories with a different meaning:

* The read permission.
  + The file is readable. For example, when the read permission is set, the user can open the file in a text editor.
  + The directory’s contents can be viewed. The user can list files inside the directory with the [ls](https://linuxize.com/post/how-to-list-files-in-linux-using-the-ls-command/) command.
* The write permission.
  + The file can be changed or modified.
  + The directory’s contents can be altered. The user can [create new files](https://linuxize.com/post/create-a-file-in-linux/) , [delete existing files](https://linuxize.com/post/how-to-remove-files-and-directories-using-linux-command-line/) , [move files](https://linuxize.com/post/how-to-move-files-in-linux-with-mv-command/) , [rename files](https://linuxize.com/post/how-to-rename-files-in-linux/) ..etc.
* The execute permission.
  + The file can be executed.
  + The directory can be entered using the [cd](https://linuxize.com/post/linux-cd-command/) command.

File permissions can be viewed using the [ls](https://linuxize.com/post/how-to-list-files-in-linux-using-the-ls-command/) command. Here is an example:

ls -l filename.txtCopy

-rw-r--r-- 12 linuxize users 12.0K Apr 8 20:51 filename.txt

|[-][-][-]- [------] [---]

| | | | | | |

| | | | | | +-----------> 7. Group

| | | | | +-------------------> 6. Owner

| | | | +--------------------------> 5. Alternate Access Method

| | | +----------------------------> 4. Others Permissions

| | +-------------------------------> 3. Group Permissions

| +----------------------------------> 2. Owner Permissions

+------------------------------------> 1. File Type

Copy

The first character shows the file type. It can be a regular file (-), directory (d), a [symbolic link](https://linuxize.com/post/how-to-create-symbolic-links-in-linux-using-the-ln-command/) (l), or any other special type of file.

The next nine characters represent the file permissions, three triplets of three characters each. The first triplet shows the owner permissions, the second one group permissions, and the last triplet shows everybody else permissions.

**Permission number**

File permission can be represented in a numeric or symbolic format. In this article, we’ll focus on the numeric format.

The permission number can consist of three or four digits, ranging from 0 to 7.

When 3 digits number is used, the first digit represents the permissions of the file’s owner, the second one the file’s group and the last one all other users.

The write, read, and execute permissions have the following number value:

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

The permissions digit of a specific user class is the sum of the values of the permissions for that class.

Each digit of the permissions number may be a sum of 4, 2, 1 and 0:

* 0 (0+0+0) – No permission.
* 1 (0+0+1) – Only execute permission.
* 2 (0+2+0) – Only write permission.
* 3 (0+2+1) – Write and execute permissions.
* 4 (4+0+0) – Only read permission.
* 5 (4+0+1) – Read and execute permission.
* 6 (4+2+0) – Read and write permissions.
* 7 (4+2+1) – Read, write, and execute permission.

For example, if the permission number is set to 750 it means that the file’s owner has read, write and execute permission, file’s group has read and execute permissions, and other users have no permissions:

* Owner: rwx=4+2+1=7
* Group: r-x=4+0+1=5
* Others: r-x=0+0+0=0

When the 4 digits number is used, the first digit has the following meaning:

* setuid=4
* setgid=2
* sticky=1
* no changes = 0

The next three digits have the same meaning as when using 3 digits number. If the first digit is 0 it can be omitted, and the mode can be represented with 3 digits. The numeric mode 0755 is the same as 755.

To view the file’s permissions in the numeric (octal) notation, use the [stat](https://linuxize.com/post/stat-command-in-linux/) command:

stat -c "%a" filename

Copy

644

Copy

**Never Use chmod 777**

Setting 777 permissions to a file or directory means that it will be readable, writable and executable by all users and may pose a huge security risk.

For example, if you recursively change the permissions of all files and subdirectories under the /var/www directory to 777, any user on the system will be able to create, delete or modify files in that directory.

If you experience permission issues with your web server, instead of recursively setting the permission to 777, change the file’s ownership to the user running the application and set the file’s permissions to 644 and directory’s permissions to 755.

File ownership can be changed using the [chown](https://linuxize.com/post/linux-chown-command/) command and permissions with the [chmod](https://linuxize.com/post/chmod-command-in-linux/) command.

Example

chown -R oracle:oinstall /u01/bkupdir

chmod -R 775 /u01

**Conclusion**

If you are managing a Linux system, it is crucial to know how the Linux permissions work.

You should never set 777 (rwxrwxrwx) permissions files and directories permissions. 777 means that anyone can do anything with those files.

Feel free to leave a comment if you have any questions.

CRON JOB IN LINUX

# rpm -qa | grep crontabs

# yum install crontabs

#systemctl start crond

#systemctl enable crond

#cat /etc/crontab

\* \* \* \* \* command to execute

│ │ │ │ │

│ │ │ │ │

│ │ │ │ └───── day of week (0 - 6) (0 to 6 are Sunday to Saturday, or use names; 7 is Sunday, the same as 0)

│ │ │ └────────── month (1 - 12)

│ │ └─────────────── day of month (1 - 31)

│ └──────────────────── hour (0 - 23)

└───────────────────────── min (0 - 59)

#crontab -e

#crontab -l