**Listing files (ls)**

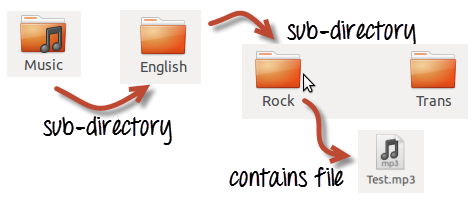
If you want to see the list of files on your UNIX or Linux system, use the '**ls'** command.  
  
It shows the files /directories in your current directory.

[](https://www.guru99.com/images/ls(1).png)

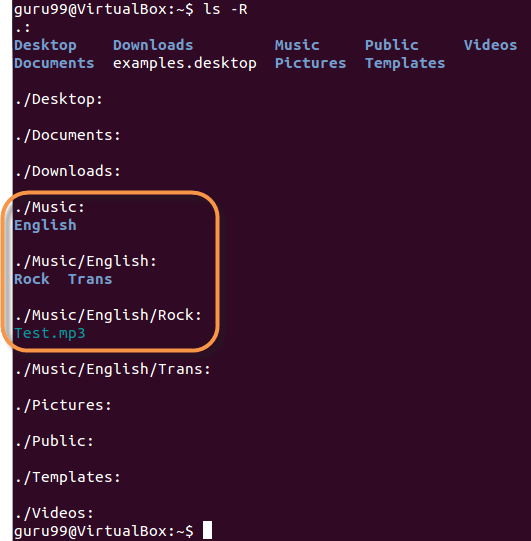
Note:

* 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.

[](https://www.guru99.com/images/sub-directory(1).png)

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

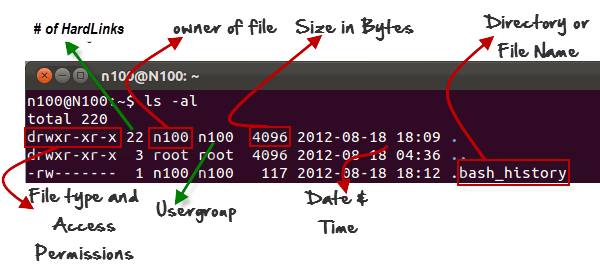
[](https://www.guru99.com/images/ls-R(1).png)

|  |  |
| --- | --- |
| **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 |

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:

Let's see an example -

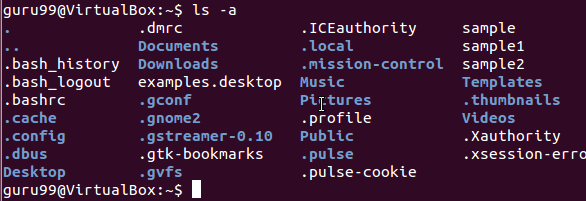
[](https://www.guru99.com/images/ls-al(2).png)

**Listing Hidden Files**

Hidden items in UNIX/Linux begin with - 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

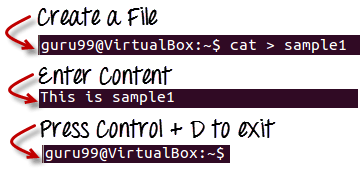
[](https://www.guru99.com/images/ls-a(2).png)

**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.

[](https://www.guru99.com/images/cat_filename(1).png)

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](https://www.guru99.com/images/cat_view_a_file(1).png)

Let's see another file sample2

[Creating a file using Linux/Unix Commands](https://www.guru99.com/images/cat_sample2.png)

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)

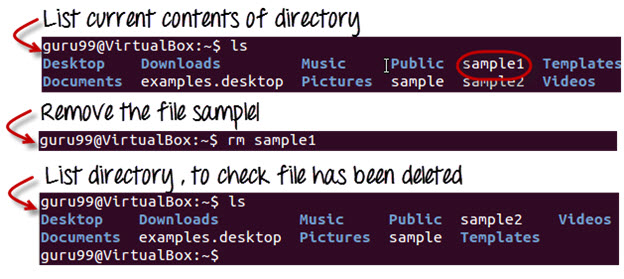
**Note:**Only text files can be displayed and combined using this command.

**Deleting Files**

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

To remove a file use syntax -

rm filename

[](https://www.guru99.com/images/linux_rm_command.jpg)

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/sathya/Documents. Executing the command

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

[How to move a file using Linux/Unix Commands](https://www.guru99.com/images/mv_error.png)

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.

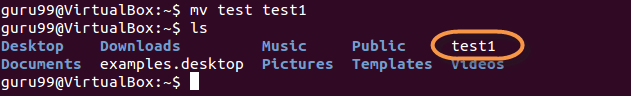
guru99@VirtualBox:~$ sudo mv sample2 /home/quru99/Documents

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

guru99@VirtualBox:~$

For renaming file:

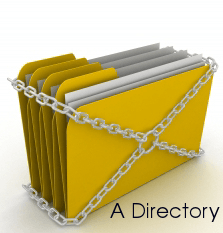
mv filename newfilename

[](https://www.guru99.com/images/mv(1).png)

**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**

[](https://www.guru99.com/images/Direct.png)

Directory Manipulation in Linux/Unix

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

Creating Directories

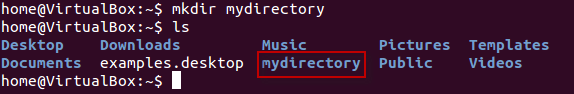
Directories can be created on a Linux operating system using the following command

mkdir directoryname

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

For example,

mkdir mydirectory

[](https://www.guru99.com/images/MKdir-1.png)

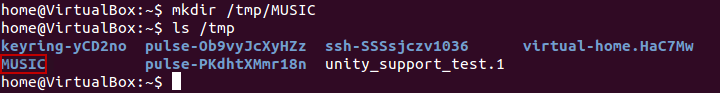
If you want to create a directory in a different location other than 'Home directory', you could use the following command -

mkdir

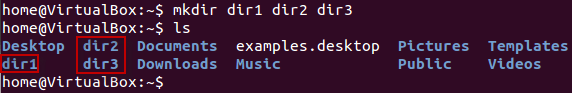
For example:

mkdir /tmp/MUSIC

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

[](https://www.guru99.com/images/8-2016/linux-5-1.png)

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

[](https://www.guru99.com/images/8-2016/linux-5-2.png)

**Removing Directories**

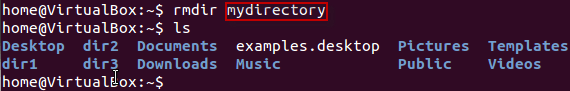
To remove a directory, use the command -

rmdir directoryname

Example

rmdir mydirectory

will delete the directory mydirectory

[](https://www.guru99.com/images/rmdir.png)

**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.

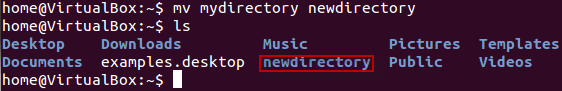
[Directory Manipulation in Linux/Unix](https://www.guru99.com/images/rmdir1.png)

**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:

[](https://www.guru99.com/images/8-2016/linux-5-3.png)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](https://www.guru99.com/introduction-linux.html). It is similar to HELP file found in popular software.

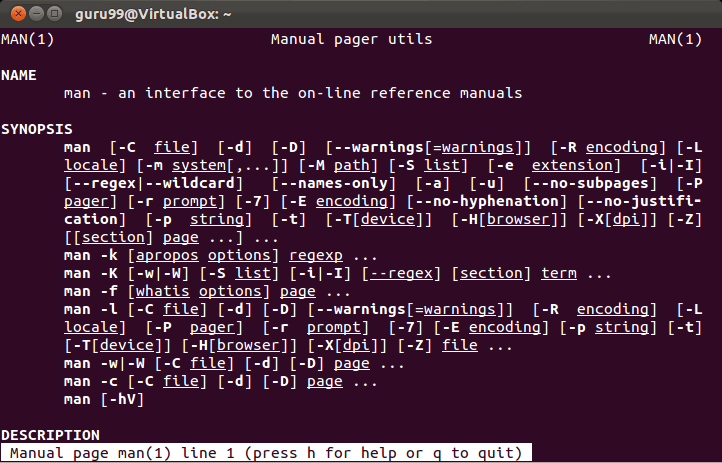
To get help on any command that you do not understand, you can type

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](https://www.guru99.com/images/man_man.png)

[](https://www.guru99.com/images/man_man_1.png)

**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.

[](https://www.guru99.com/images/history.png)

**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.

[](https://www.guru99.com/images/clear.png)

**Pasting commands into the terminal**

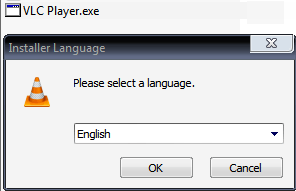
Many times you would have to type in long commands on the Terminal. Well, it can be annoying at times, and if you want to avoid such a situation then copy, pasting the commands can come to rescue.

For copying, the text from a source, you would use **Ctrl + c,** but for pasting it on the Terminal, you need to use **Ctrl + Shift + p**. You can also try **Shift + Insert or select Edit>Paste on the menu**

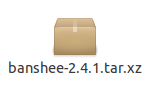
NOTE: With Linux upgrades, these shortcuts keep changing. You can set your preferred shortcuts via Terminal> Edit> Keyboard Shortcuts.

**Installing Software**

In windows, the installation of a program is done by running the setup.exe file. The installation bundle contains the program as well various dependent components required to run the program correctly.

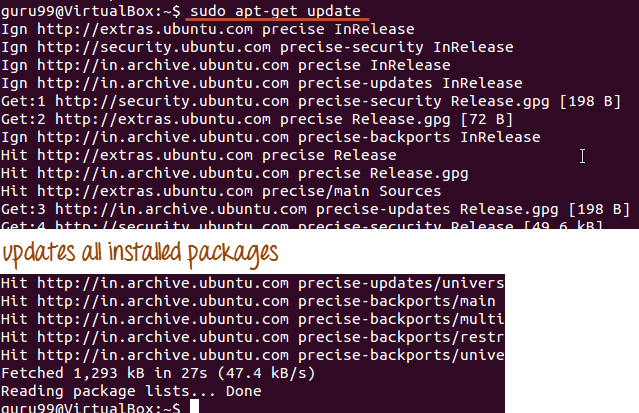
[](https://www.guru99.com/images/VLCPlayer.png)

Using Linux/Unix basic commands, installation files in Linux are distributed as packages. But the package contains only the program itself. Any dependent components will have to be installed separately which are usually available as packages themselves.

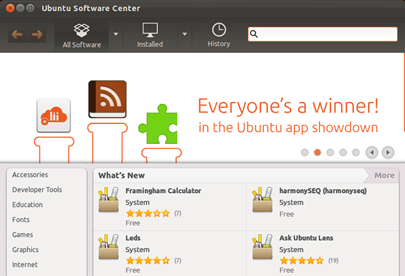
[](https://www.guru99.com/images/Banshee.png)

You can use the **apt** commands to install or remove a package. Let's update all the installed packages in our system using command -

sudo apt-get update

[](https://www.guru99.com/images/apt.png)

The easy and popular way to install programs on Ubuntu is by using the Software center as most of the software packages are available on it and it is far more secure than the files downloaded from the internet.

[](https://www.guru99.com/images/SoftwareCenter.png)

**Linux Mail Command**

For sending mails through a terminal, you will need to install packages 'mailutils'.

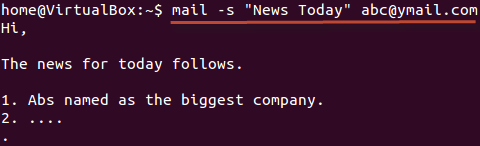
The command syntax is -

sudo apt-get install packagename

Once done, you can then use the following syntax for sending an email.

mail -s 'subject' -c 'cc-address' -b 'bcc-address' 'to-address'

This will look like:

[](https://www.guru99.com/images/mail.png)

Press Cntrl+D you are finished writing the mail. The mail will be sent to the mentioned address.

**Summary**

* You can format and print a file directly from the terminal. The formatting you do on the files does not affect the file contents
* In Unix/Linux,  software is installed in the form of packages. A package contains the program itself. Any dependent component needs to be downloaded separately.
* You can also send e-mails from terminal using the **'mail'** network commands. It is very useful Linux command.

**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 |
| pr -x | Divides the file into x columns |
| pr -h | Assigns a header to the file |
| pr -n | Denotes the file with Line Numbers |
| lp -nc lpr c | Prints "c" copies of the File |
| lp -d lpr -P | Specifies name of the printer |
| apt-get | Command used to install and update packages |
| mail -s 'subject' -c 'cc-address' -b 'bcc-address' 'to-address' | Command to send email |
| mail -s "Subject" to-address < Filename | Command to send email with attachment |

**Ownership of Linux files**

Every file and directory on your Unix/Linux system is assigned 3 types of owner, given below.

**User**

A user is the owner of the file. By default, the person who created a file becomes its owner. Hence, a user is also sometimes called an owner.

**Group**

A user- group can contain multiple users. All users belonging to a group will have the same Linux group permissions access to the file. Suppose you have a project where a number of people require access to a file. Instead of manually assigning permissions to each user, you could add all users to a group, and assign group permission to file such that only this group members and no one else can read or modify the files.

**Other**

Any other user who has access to a file. This person has neither created the file, nor he belongs to a usergroup who could own the file. Practically, it means everybody else. Hence, when you set the permission for others, it is also referred as set permissions for the world.

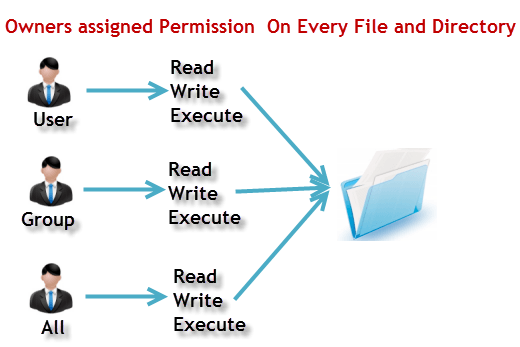
Now, the big question arises how does **Linux distinguish**between these three user types so that a user 'A' cannot affect a file which contains some other user 'B's' vital information/data. It is like you do not want your colleague, who works on your Linux computer, to view your images. This is where **Permissions** set in, and they define **user behavior**.

Let us understand the **Permission system** on Linux.

**Permissions**

Every file and directory in your UNIX/Linux system has following 3 permissions defined for all the 3 owners discussed above.

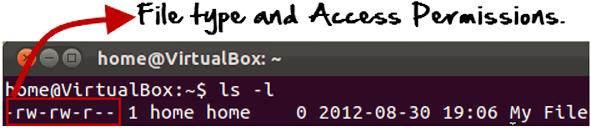
* **Read:** This permission give you the authority to open and read a file. Read permission on a directory gives you the ability to lists its content.
* **Write:**The write permission gives you the authority to modify the contents of a file. The write permission on a directory gives you the authority to add, remove and rename files stored in the directory. Consider a scenario where you have to write permission on file but do not have write permission on the directory where the file is stored. You will be able to modify the file contents. But you will not be able to rename, move or remove the file from the directory.
* **Execute:**In Windows, an executable program usually has an extension ".exe" and which you can easily run. In Unix/Linux, you cannot run a program unless the execute permission is set. If the execute permission is not set, you might still be able to see/modify the program code(provided read & write permissions are set), but not run it.

[](https://www.guru99.com/images/PermissionsConcept.png)File Permissions in Linux/Unix

Let's see file permissions in Linux with examples:

**ls - l** on terminal gives

ls - l

[](https://www.guru99.com/images/Permis_system.png)

Here, we have highlighted **'-rw-rw-r--'**and this weird looking code is the one that tells us about the Unix permissions given to the owner, user group and the world.

Here, the first '**-**' implies that we have selected a file.p>

[](https://www.guru99.com/images/its_a_file.png)

Else, if it were a directory, **d**would have been shown.

[File Permissions in Linux/Unix](https://www.guru99.com/images/Directory.png)

The characters are pretty easy to remember.

**r** = read permission  
**w** = write permission  
**x** = execute permission  
**-** = no permission

Let us look at it this way.

The first part of the code is **'rw-'**. This suggests that the owner 'Home' can:

[](https://www.guru99.com/images/no_execute.png)

* Read the file
* Write or edit the file
* He cannot execute the file since the execute bit is set to '-'.

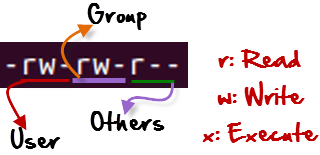
By design, many Linux distributions like Fedora, CentOS, Ubuntu, etc. will add users to a group of the same group name as the user name. Thus, a user 'tom' is added to a group named 'tom'.

The second part is **'rw-'.** It for the user group 'Home' and group-members can:

* Read the file
* Write or edit the file

The third part is for the world which means any user. It says **'r--'.** This means the user can only:

* Read the file

[](https://www.guru99.com/images/permission(1).png)

**Changing file/directory permissions with 'chmod' command**

Say you do not want your colleague to see your personal images. This can be achieved by changing file permissions.

We can use the '**chmod'** command which stands for 'change mode'. Using the command, we can set permissions (read, write, execute) on a file/directory for the owner, group and the world. **Syntax:**

chmod permissions filename

There are 2 ways to use the command -

1. **Absolute mode**
2. **Symbolic mode**

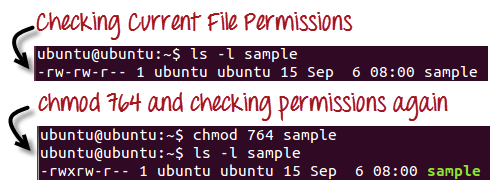
|  |  |  |
| --- | --- | --- |
| **Number** | **Permission Type** | **Symbol** |
| 0 | No Permission | --- |
| 1 | Execute | --x |
| 2 | Write | -w- |
| 3 | Execute + Write | -wx |
| 4 | Read | r-- |
| 5 | Read + Execute | r-x |
| 6 | Read +Write | rw- |
| 7 | Read + Write +Execute | rwx |

**Absolute(Numeric) Mode**

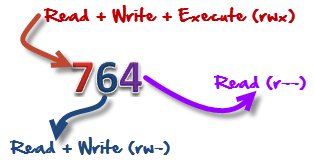
In this mode, file **permissions are not represented as characters but a three-digit octal number**.

The table below gives numbers for all for permissions types.

Let's see the chmod permissions command in action.

[](https://www.guru99.com/images/chmod_new(1).png)

In the above-given terminal window, we have changed the permissions of the file 'sample to '764'.

[](https://www.guru99.com/images/FilePermissions(1).png)

'764' absolute code says the following:

* Owner can read, write and execute
* Usergroup can read and write
* World can only read

**This is shown as '-rwxrw-r--**

This is how you can change user permissions in Linux on file by assigning an absolute number.

**Symbolic Mode**

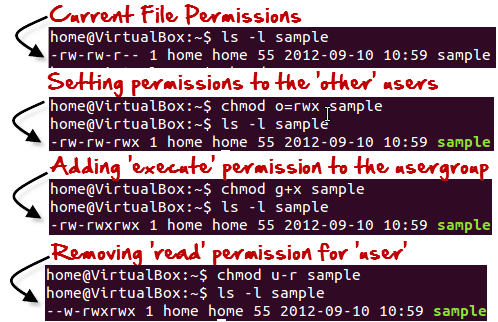
In the Absolute mode, you change permissions for all 3 owners. In the symbolic mode, you can modify permissions of a specific owner. It makes use of mathematical symbols to modify the Unix file permissions.

|  |  |
| --- | --- |
| **Operator** | **Description** |
| **+** | Adds a permission to a file or directory |
| **-** | Removes the permission |
| **=** | Sets the permission and overrides the permissions set earlier. |

The various owners are represented as –

|  |  |
| --- | --- |
| **User Denotations** | |
| u | user/owner |
| g | group |
| o | other |
| a | All |

We will not be using permissions in numbers like 755 but characters like rwx. Let's look into an example

[](https://www.guru99.com/images/Symbolic_Mode(1).png)

**Changing Ownership and Group**

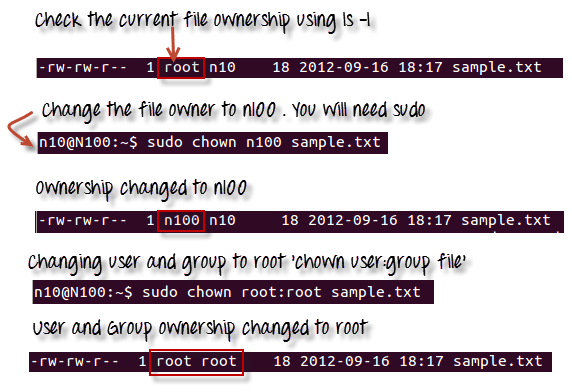
For changing the ownership of a file/directory, you can use the following command:

chown user

In case you want to change the user as well as group for a file or directory use the command

chown user:group filename

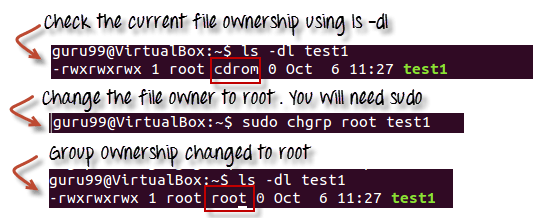
Let's see this in action

[](https://www.guru99.com/images/chown_comm(1).png)

In case you want to change group-owner only, use the command

chgrp group\_name filename

'**chgrp'** stands for change group.

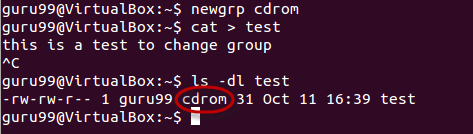
[](https://www.guru99.com/images/chgrp.png)

**Tip**

* The file /etc/group contains all the groups defined in the system
* You can use the command "groups" to find all the groups you are a member of

[File Permissions in Linux/Unix](https://www.guru99.com/images/groups.png)

* You can use the command newgrp to work as a member a group other than your default group

[](https://www.guru99.com/images/newgrp.png)

* You cannot have 2 groups owning the same file.
* You do not have nested groups in Linux. One group cannot be sub-group of other
* x- eXecuting a directory means Being allowed to "enter" a dir and gain possible access to sub-dirs
* There are other permissions that you can set on Files and Directories which will be covered in a later advanced tutorial

**Summary:**

* Linux being a multi-user system uses permissions and ownership for security.
* There are three user types on a Linux system viz. User, Group and Other
* Linux divides the file permissions into read, write and execute denoted by r,w, and x
* The permissions on a file can be changed by 'chmod' command which can be further divided into Absolute and Symbolic mode
* The 'chown' command can change the ownership of a file/directory. Use the following commands: chown user file or chown user:group file
* The 'chgrp' command can change the group ownership **chrgrp group filename**
* What does x - eXecuting a directory mean? A: Being allowed to "enter" a dir and gain possible access to sub-dirs.

# **Pipe, Grep and Sort Command in Linux/Unix with Examples**

In this tutorial, we will learn-

* [What is a Pipe in Linux?](https://www.guru99.com/linux-pipe-grep.html#1)
* ['pg' and 'more' commands](https://www.guru99.com/linux-pipe-grep.html#2)
* [The 'grep' command](https://www.guru99.com/linux-pipe-grep.html#3)
* [The 'sort' command](https://www.guru99.com/linux-pipe-grep.html#4)
* [What is a Filter?](https://www.guru99.com/linux-pipe-grep.html#5)

## What is a Pipe in Linux?

The Pipe is a command in Linux that lets you use two or more commands such that output of one command serves as input to the next. In short, the output of each process directly as input to the next one like a pipeline. The symbol '|' denotes a pipe.

Pipes help you mash-up two or more commands at the same time and run them consecutively. You can use powerful commands which can perform complex tasks in a jiffy.

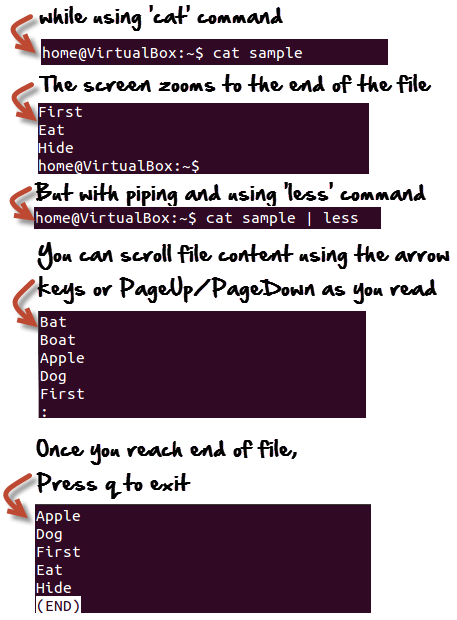
Let us understand this with an example.

When you use 'cat' command to view a file which spans multiple pages, the prompt quickly jumps to the last page of the file, and you do not see the content in the middle.

To avoid this, you can pipe the output of the 'cat' command to 'less' which will show you only one scroll length of content at a time.

cat filename | less

An illustration would make it clear.

[](https://www.guru99.com/images/Piping.png)

Click [here](https://www.guru99.com/faq.html#1) if the video is not accessible

## 'pg' and 'more' commands

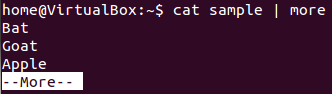
Instead of 'less', you can also use.

cat Filename | pg

or

cat Filename | more

And, you can view the file in digestible bits and scroll down by simply hitting the enter key.

[](https://www.guru99.com/images/more.png)

## The 'grep' command

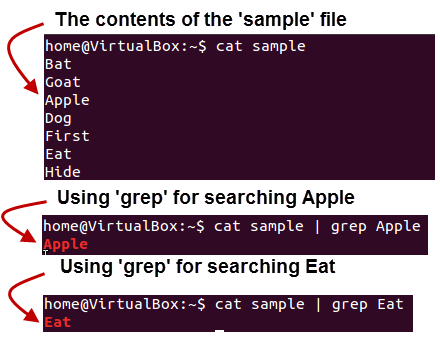
Suppose you want to search a particular information the postal code from a text file.

You may manually skim the content yourself to trace the information. A better option is to use the grep command. It will scan the document for the desired information and present the result in a format you want.

**Syntax:**

grep search\_string

Let's see it in action -

[](https://www.guru99.com/images/Grep_command.png)

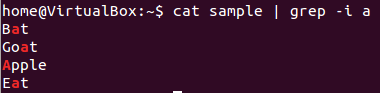
Here, **grep** command has searched the file 'sample', for the string 'Apple' and 'Eat'.

Following options can be used with this command.

|  |  |
| --- | --- |
| **Option** | **Function** |
| -v | Shows all the lines that do not match the searched string |
| -c | Displays only the count of matching lines |
| -n | Shows the matching line and its number |
| -i | Match both (upper and lower) case |
| -l | Shows just the name of the file with the string |

Let us try the first option **'-i'**on the same file use above -

Using the 'i' option grep has filtered the string 'a' (case-insensitive) from the all the lines.

[](https://www.guru99.com/images/grep_-i.png)

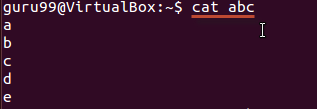
## The 'sort' command

This command helps in **sorting out the contents of a file alphabetically.**

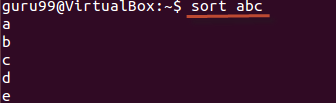
The syntax for this command is:

sort Filename

Consider the contents of a file.

[](https://www.guru99.com/images/cat_abc.png)

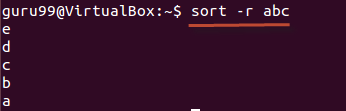
Using the sort command

[](https://www.guru99.com/images/sort_abc.png)

There are **extensions** to this command as well, and they are listed below.

|  |  |
| --- | --- |
| **Option** | **Function** |
| -r | Reverses  sorting |
| -n | Sorts numerically |
| -f | Case insensitive sorting |

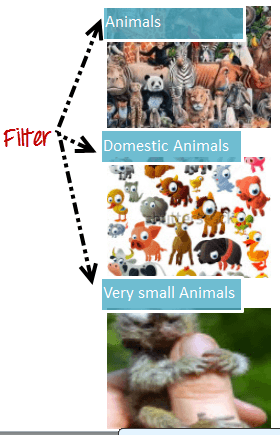
The example below shows reverse sorting of the contents in file 'abc'.

[](https://www.guru99.com/images/sort_-r_abc(1).png)

## What is a Filter?

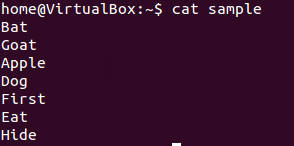
Linux has a lot of filter commands like awk, grep, sed, spell, and wc. A filter takes input from one command, does some processing, and gives output.

When you pipe two commands, the "filtered " output of the first command is given to the next.

[](https://www.guru99.com/images/Filter(1).png)

Let's understand this with the help of an example.

We have the following file 'sample'

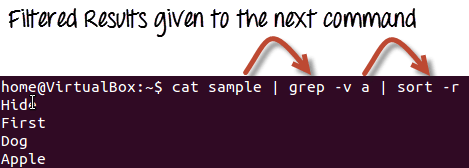
[](https://www.guru99.com/images/cat_sample.png)

**We want to highlight** only the lines that do not contain the character 'a', but the result should be in reverse order.

For this, the following syntax can be used.

cat sample | grep -v a | sort - r

Let us look at the result.

[](https://www.guru99.com/images/empirical_usage.png)

### Summary:

* Pipes '|' send the output of one command as input of another command.
* The Filter takes input from one command, does some processing, and gives output.
* The grep command can be used to find strings and values in a text document
* Piping through grep has to be one of the most common uses
* 'sort' command sorts out the content of a file alphabetically
* Less ,pg and more commands are used for dividing a long file into readable bits

# **Linux Regular Expression Tutorial: Grep Regex Example**

# **What are Linux Regular Expressions?**

**Linux Regular Expressions** are special characters which help search data and matching complex patterns. Regular expressions are shortened as 'regexp' or 'regex'. They are used in many Linux programs like grep, bash, rename, sed, etc.

## Types of Regular expressions

For ease of understanding let us learn the different types of Regex one by one.

* [Basic Regular expressions](https://www.guru99.com/linux-regular-expressions.html#1)
* [Interval Regular expressions](https://www.guru99.com/linux-regular-expressions.html#2)
* [Extended regular expressions](https://www.guru99.com/linux-regular-expressions.html#3)
* [Summary](https://www.guru99.com/linux-regular-expressions.html#4)

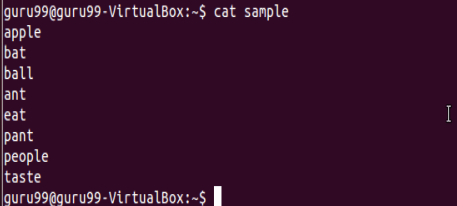
**Basic Regular expressions**

Some of the commonly used commands with Regular expressions are tr, sed, vi and grep. Listed below are some of the basic Regex.

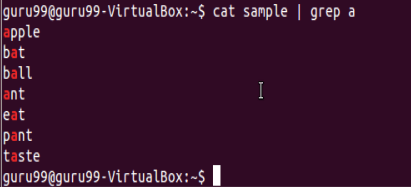
|  |  |
| --- | --- |
| **Symbol** | **Descriptions** |
| . | replaces any character |
| ^ | matches start of string |
| $ | matches end of string |
| \* | matches up zero or more times the preceding character |
| \ | Represent special characters |
| () | Groups regular expressions |
| ? | Matches up exactly one character |

Let's see an example.

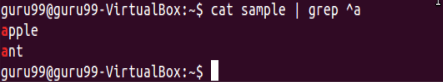
Execute cat sample to see contents of an existing file

[](https://www.guru99.com/images/regex1.png)

Search for content containing letter 'a'.

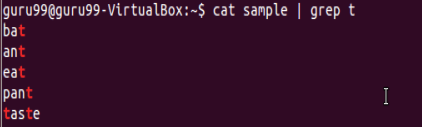
[](https://www.guru99.com/images/regex2.png)

'**^**' matches the start of a string. Let's search for content that STARTS with a

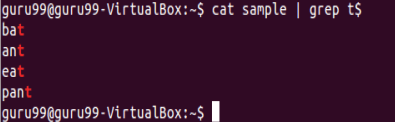
[](https://www.guru99.com/images/regex3.png)

Only lines that start with character are filtered. Lines which do not contain the character 'a' at the start are ignored.

Let's look into another example -

[](https://www.guru99.com/images/regex4.png)

Select only those lines that end with t using**$**

[](https://www.guru99.com/images/regex5.png)

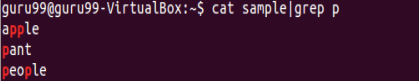
**Interval Regular expressions**

These expressions tell us about the number of occurrences of a character in a string. They are

|  |  |
| --- | --- |
| **Expression** | **Description** |
| {n} | Matches the preceding character appearing 'n' times exactly |
| {n,m} | Matches the preceding character appearing 'n' times but not more than m |
| {n, } | Matches the preceding character only when it appears 'n' times or more |

Example:

Filter out all lines that contain character 'p'

[](https://www.guru99.com/images/regex6.png)

We want to check that the character 'p' appears exactly 2 times in a string one after the other. For this the syntax would be:

cat sample | grep -E p\{2}

[Linux - Regular Expressions](https://www.guru99.com/images/regex7.png)

Note: You need to add -E with these regular expressions.

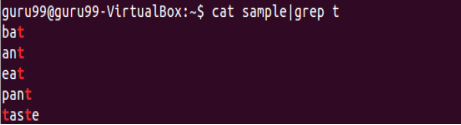
**Extended regular expressions**

These regular expressions contain combinations of more than one expression. Some of them are:

|  |  |
| --- | --- |
| **Expression** | **Description** |
| \+ | Matches one or more occurrence of the previous character |
| \? | Matches zero or one occurrence of the previous character |

Example:

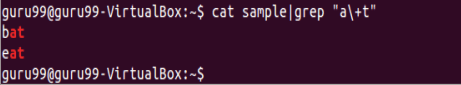
Searching for all characters 't'

[](https://www.guru99.com/images/regex8.png)

Suppose we want to filter out lines where character 'a' precedes character 't'

We can use command like

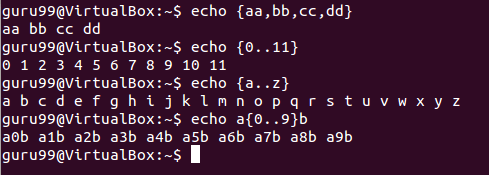
cat sample|grep "a\+t"

[](https://www.guru99.com/images/regex9.png)

**Brace expansion**

The syntax for brace expansion is either a sequence or a comma separated list of items inside curly braces "{}". The starting and ending items in a sequence are separated by two periods "..".

Some examples:

[](https://www.guru99.com/images/brace_expansion.png)

In the above examples, the echo command creates strings using the brace expansion.

**Summary:**

* Regular expressions are a set of characters used to check patterns in strings
* They are also called 'regexp' and 'regex'
* It is important to learn regular expressions for writing scripts
* Some basic regular expressions are:

|  |  |
| --- | --- |
| **Symbol** | **Descriptions** |
| . | replaces any character |
| ^ | matches start of string |
| $ | matches end of string |

|  |  |
| --- | --- |
| **Expression** | **Description** |
| \+ | Matches one or more occurrence of the previous character |
| \? | Matches zero or one occurrence of the previous character |

|  |  |
| --- | --- |
| **Expression** | **Description** |
| {n} | Matches the preceding character appearing 'n' times exactly |
| {n,m} | Matches the preceding character appearing 'n' times but not more than m |
| {n, } | Matches the preceding character only when it appears 'n' times or more |

* The brace expansion is used to generate strings. It helps in creating multiple strings out of one.

# **List of Environment Variables in Linux/Unix**

In this tutorial, you will learn-

* [What is a Computing Environment?](https://www.guru99.com/linux-environment-variables.html#1)
* [What is a Variable?](https://www.guru99.com/linux-environment-variables.html#2)
* [What are Environment variables?](https://www.guru99.com/linux-environment-variables.html#3)
* [Accessing Variable values](https://www.guru99.com/linux-environment-variables.html#4)
* [Set New Environment Variables](https://www.guru99.com/linux-environment-variables.html#5)

## What is a Computing Environment?

The Computing environment is the Platform(Platform = Operating System+ Processor) where a user can run programs.

## What is a Variable?

In computer science, a **variable is a location for storing a value** **which can be a** **filename**, **text**, **number** or any other **data**. It is usually referred to with its Symbolic name which is given to it while creation. The value thus stored can be displayed, deleted, edited and re-saved.

Variables play an important role in computer programming because they enable programmers to write flexible programs. As they are related to the Operating system that we work on, it is important to know some of them and how we can influence them.

## What are Environment variables?

Environment variables are dynamic values which affect the processes or programs on a computer. They exist in every operating system, but types may vary. Environment variables can be created, edited, saved, and deleted and give information about the system behavior.

Environment variables can change the way a software/programs behave.

E.g. $LANG environment variable stores the value of the language that the user understands. This value is read by an application such that a Chinese user is shown a Mandarin interface while an American user is shown an English interface.

Let's study some common environment variables -

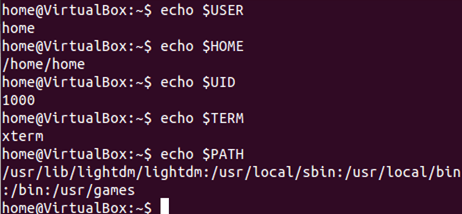
|  |  |
| --- | --- |
| **Variable** | **Description** |
| **PATH** | This variable contains a colon (:)-separated list of directories in which your system looks for executable files. [Linux - Environment Variables](https://www.guru99.com/images/echoPath.png) When you enter a command on terminal, the shell looks for the command in different directories mentioned in the $PATH variable. If the command is found, it executes. Otherwise, it returns with an error 'command not found'. |
| **USER** | The username |
| **HOME** | Default path to the user's home directory |
| **EDITOR** | Path to the program which edits the content of files |
| **UID** | User's unique ID |
| **TERM** | Default terminal emulator |
| **SHELL** | Shell being used by the user |

## Accessing Variable values

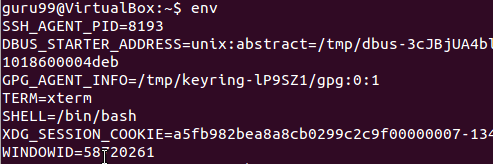
In order to determine value of a variable, use the command

echo $VARIABLE

Variables are- Case Sensitive. Make sure that you type the variable name in the right letter case otherwise you may not get the desired results.

[](https://www.guru99.com/images/echo.png)

The '**env'** command displays **all the environment variables.**

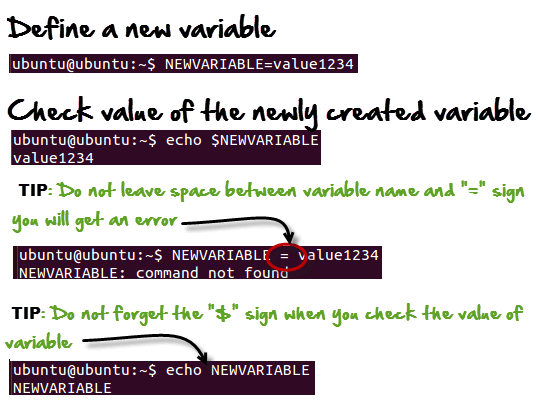
[](https://www.guru99.com/images/env.png)

## Set New Environment Variables

You can create your own user defined variable, with syntax

VARIABLE\_NAME= variable\_value

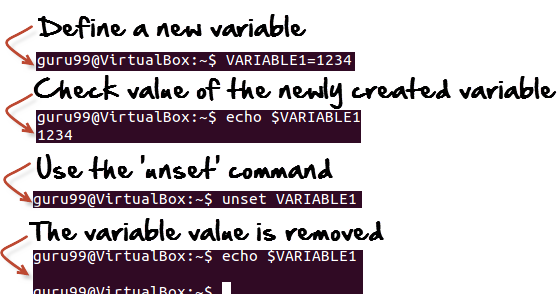
Again, bear in mind **that variables are case-sensitive** and usually they are created in upper case.

[](https://www.guru99.com/images/New_variable.png)

## Deleting Variables

The following syntax can be used to remove a Variable from the system.

unset variablename

[](https://www.guru99.com/images/unset.png)

This **would remove the Variable** and its value permanently.

### Summary:

* Environment variables govern the behavior of programs in your Operating System.

|  |  |
| --- | --- |
| **Command** | **Description** |
| echo $VARIABLE | To display value of a variable |
| env | Displays all environment variables |
| VARIABLE\_NAME= variable\_value | Create a new variable |
| unset | Remove a variable |
| export Variable=value | To set value of an environment variable |

# Linux/Unix SSH, Ping, FTP, Telnet Communication Commands

While working on a Linux operating system, you may need to **communicate with other devices**. For this, there are some basic utilities that you can make use of.

These utilities can help you communicate with:

* networks,
* other Linux systems
* and remote users

So, let us learn them one by one.

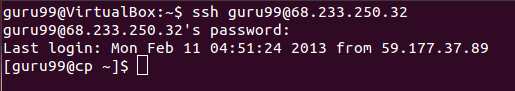
* [SSH](https://www.guru99.com/communication-in-linux.html#1)
* [Ping](https://www.guru99.com/communication-in-linux.html#2)
* [FTP](https://www.guru99.com/communication-in-linux.html#3)
* [Telnet](https://www.guru99.com/communication-in-linux.html#4)

**SSH**

SSH which stands for Secure Shell, It is used to connect to a remote computer securely. Compare to Telnet, SSH is secure wherein the client /server connection is authenticated using a digital certificate and passwords are encrypted. Hence it's widely used by system administrators to control remote Linux servers.

The syntax to log into a remote Linux machine using SSH is

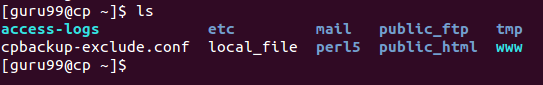
SSH username@ip-address or hostname

[](https://www.guru99.com/images/11-Feb-13_3-24-15_PM.png)

Once you are logged in, you can execute any commands that you do in your terminal

**Example:**

ls

[](https://www.guru99.com/images/11-Feb-13_3-24-50_PM.png)

**Example:**

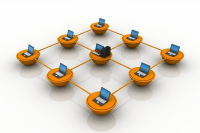
pwd

[Linux/Unix SSH, Ping, FTP, Telnet Communication Commands](https://www.guru99.com/images/11-Feb-13_3-25-12_PM.png)

**Ping**

This utility is commonly used to check whether your **connection to the server** is healthy or not.This command is also used in -

* Analyzing network and host connections
* Tracking network performance and managing it
* Testing hardware and software issues

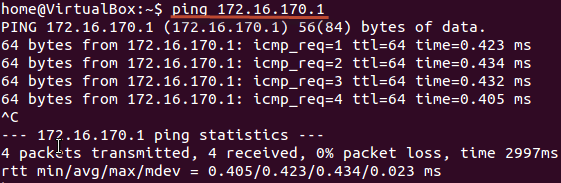
[](https://www.guru99.com/images/communication(1).png)

Command Syntax:-

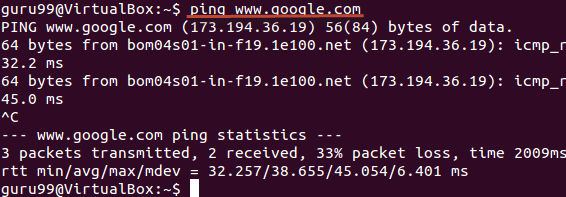
ping hostname="" or=""

Example :

ping 172.16.170.1

**[](https://www.guru99.com/images/ping.png)**

ping google.com

[](https://www.guru99.com/images/ping_hostname.png)

Here, A system has sent 64 bytes data packets to the IP Address (172.16.170.1) or the Hostname(www.google.com). If even one of data packets does not return or is lost, it would suggest an error in the connection. Usually, internet connectivity is checked using this method.

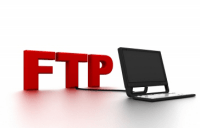
You may Press **Ctrl + c** to **exit** from the ping loop.

**FTP**

FTP **is file transfer protocol**. It's the **most preferred protocol for** **data transfer** amongst computers.

You can use FTP to -

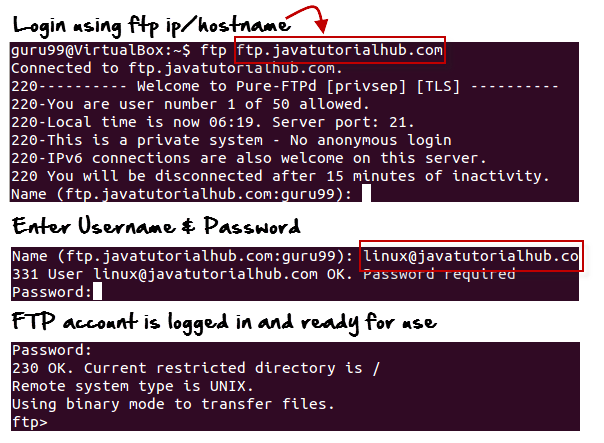
* Logging in and establishing a connection with a remote host
* Upload and download files
* Navigating through directories
* Browsing contents of the directories

[](https://www.guru99.com/images/ftp(1).png)

The syntax to establish an FTP connection to a remote host is -

ftp hostname="" or=""

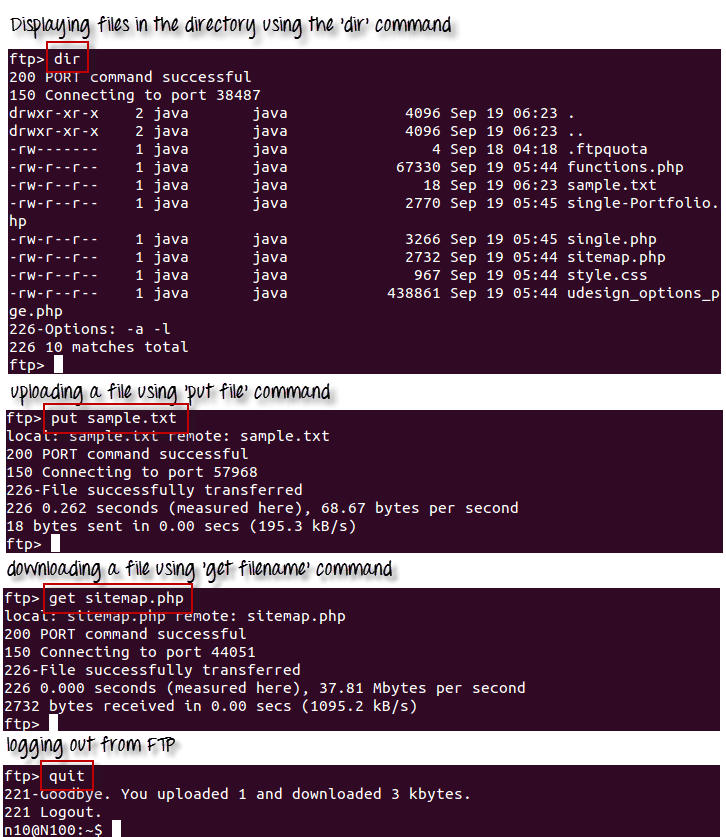
Once you enter this command, it will ask you for **authentication** via username and password.

[](https://www.guru99.com/images/FTP_login.png)

Once a connection is established, and you are logged in, you may use the following commands to perform different actions.

|  |  |
| --- | --- |
| **Command** | **Function** |
| dir | Display files in the current directory of a remote computer |
| cd "dirname" | change directory to "dirname" on a remote computer |
| put file | upload 'file' from local to remote computer |
| get file | Download 'file' from remote to local computer |
| quit | Logout |

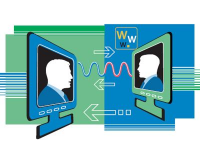
Let us run some of the important commands.

[](https://www.guru99.com/images/ftp_commands1.png)

**Telnet**

Telnet helps to -

* connect to a remote Linux computer
* run programs remotely and conduct administration

[](https://www.guru99.com/images/telnet(1).png)

This utility is similar to the Remote Desktop feature found in Windows Machine.

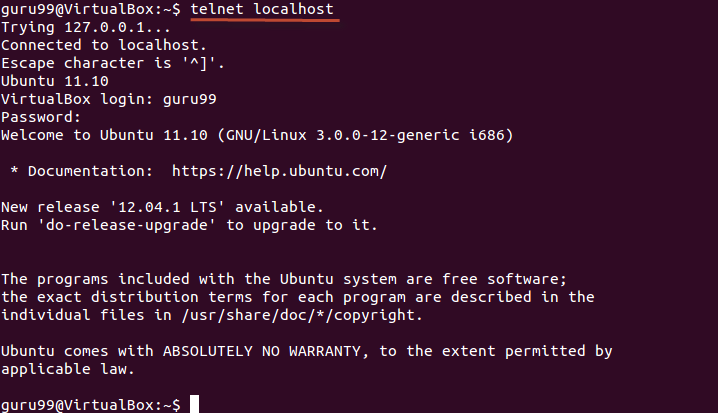
The syntax for this utility is:

telnet hostname="" or=""

Example:

telnet localhost

For demonstration purpose, we will connect to your computer (localhost). The utility will ask your username and password.

**[](https://www.guru99.com/images/telnet_localhost.png)**

Once authenticated, you can execute commands just like you have done so far, using the Terminal. The only difference is, if you are connected to a remote host, the commands will be executed on the remote machine, and not your local machine.

You may exit the telnet connection by entering the command 'logout'

**Summary:**

* Communication between Linux/UNIX and other different computers, networks and remote users is possible.
* The ping command checks whether the connection with a hostname or IP-address is working or not. Run 'ping IP address or Hostname' on the terminal
* FTP is preferred protocol for sending and receiving large files. You can establish an FTP connection to a remote host and then use commands for uploading, downloading files, checking file and browsing them
* Telnet utility helps you to connect to a remote Linux computer and work on it

# Linux/Unix Process Management: ps, kill, top, df, free, nice Commands

In this tutorial, you will learn-

* [What is a Process?](https://www.guru99.com/managing-processes-in-linux.html#1)
* [Running a Foreground Process](https://www.guru99.com/managing-processes-in-linux.html#2)
* [Running a Background process](https://www.guru99.com/managing-processes-in-linux.html#3)
* [Fg](https://www.guru99.com/managing-processes-in-linux.html#4)
* [Top](https://www.guru99.com/managing-processes-in-linux.html#5)
* [PS](https://www.guru99.com/managing-processes-in-linux.html#6)
* [Kill](https://www.guru99.com/managing-processes-in-linux.html#7)
* [NICE](https://www.guru99.com/managing-processes-in-linux.html#8)
* [DF](https://www.guru99.com/managing-processes-in-linux.html#9)
* [Free](https://www.guru99.com/managing-processes-in-linux.html#10)

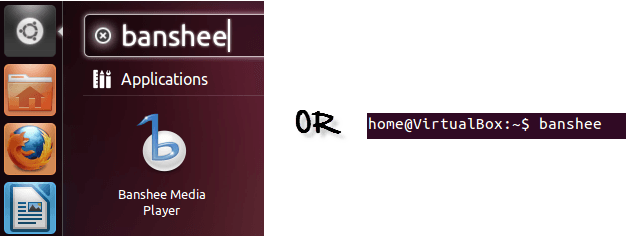
## What is a Process?

An instance of a program is called a Process. In simple terms, any command that you give to your Linux machine starts a new process.

**Running a Foreground Process**

To start a foreground process, you can either run it from the dashboard, or you can run it from the terminal.

When using the Terminal, you will have to wait, until the foreground process runs.

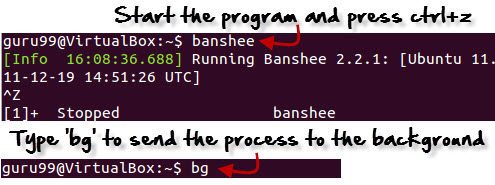
[](https://www.guru99.com/images/foreground.png)

**Running a Background process**

If you start a foreground program/process from the terminal, then you cannot work on the terminal, till the program is up and running.

Particular, data-intensive tasks take lots of processing power and may even take hours to complete. You do not want your terminal to be held up for such a long time.

To avoid such a situation, you can run the program and send it to the background so that terminal remains available to you. Let's learn how to do this -

[](https://www.guru99.com/images/bg.jpg)

**Fg**

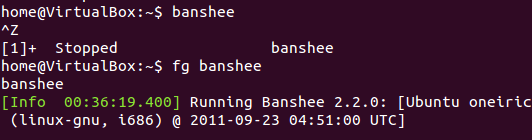
You can use the command "fg" to continue a program which was stopped and bring it to the foreground.

The simple syntax for this utility is:

fg jobname

Example

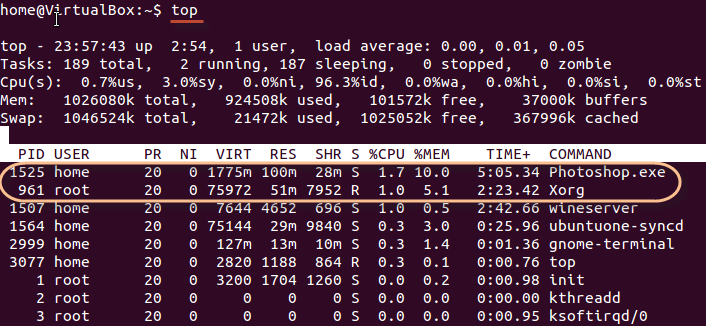
1. Launch 'banshee' music player
2. Stop it with the 'ctrl +z' command
3. Continue it with the 'fg' utility.

[](https://www.guru99.com/images/fg.png)

Let's look at other important commands to manage processes -

**Top**

This utility tells the user about all the running processes on the Linux machine.

[](https://www.guru99.com/images/top.png)

Press 'q' on the keyboard to move out of the process display.

The terminology follows:

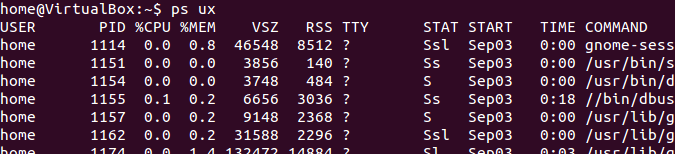
| **Field** | **Description** | **Example 1** | **Example 2** |
| --- | --- | --- | --- |
| PID | The process ID of each task | 1525 | 961 |
| User | The username of task owner | Home | Root |
| PR | Priority Can be 20(highest) or -20(lowest) | 20 | 20 |
| NI | The nice value of a task | 0 | 0 |
| VIRT | Virtual memory used (kb) | 1775 | 75972 |
| RES | Physical memory used (kb) | 100 | 51 |
| SHR | Shared memory used (kb) | 28 | 7952 |
| S | Status  There are five types:            'D' = uninterruptible sleep            'R' = running            'S' = sleeping            'T' = traced or stopped            'Z' = zombie | S | R |
| %CPU | % of CPU time | 1.7 | 1.0 |
| %MEM | Physical memory used | 10 | 5.1 |
| TIME+ | Total CPU time | 5:05.34 | 2:23.42 |
| Command | Command name | Photoshop.exe | Xorg |

**PS**

This command stands for 'Process Status'. It is similar to the "Task Manager" that pop-ups in a Windows Machine when we use Cntrl+Alt+Del. This command is similar to 'top' command but the information displayed is different.

To check all the processes running under a user, use the command -

ps ux

[](https://www.guru99.com/images/ps.png)

You can also check the process status of a single process, use the syntax -

ps PID

[Managing Processes in Linux/Unix: top, ps, kill, df, free, nice](https://www.guru99.com/images/ps_pid.jpg)

**Kill**

This command **terminates running processes** on a Linux machine.

To use these utilities you need to know the PID (process id) of the process you want to kill

Syntax -

kill PID

To find the PID of a process simply type

pidof Process name

Let us try it with an example.

[Managing Processes in Linux/Unix: top, ps, kill, df, free, nice](https://www.guru99.com/images/kill.png)

**NICE**

Linux can run a lot of processes at a time, which can slow down the speed of some high priority processes and result in poor performance.

To avoid this, you can tell your machine to prioritize processes as per your requirements.

This priority is called Niceness in Linux, and it has a value between -20 to 19. The lower the Niceness index, the higher would be a priority given to that task.

The default value of all the processes is 0.

To start a process with a niceness value other than the default value use the following syntax

nice -n 'Nice value' process name

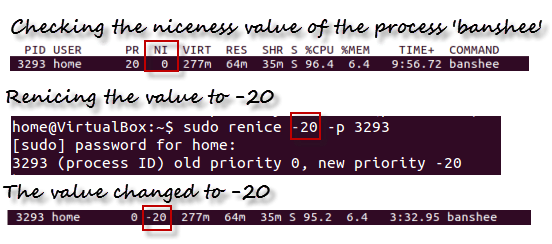
[Managing Processes in Linux/Unix: top, ps, kill, df, free, nice](https://www.guru99.com/images/changing_niceness.png)

If there is some process already running on the system, then you can 'Renice' its value using syntax.

renice 'nice value' -p 'PID'

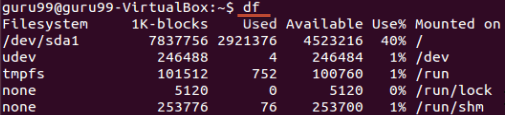
To change Niceness, you can use the 'top' command to determine the PID (process id) and its Nice value. Later use the renice command to change the value.

Let us understand this by an example.

[](https://www.guru99.com/images/renicing.png)

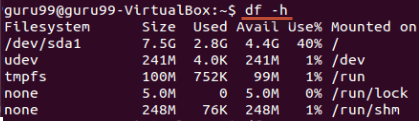
**DF**

This utility reports the free disk space(Hard Disk) on all the file systems.

[](https://www.guru99.com/images/df.png)

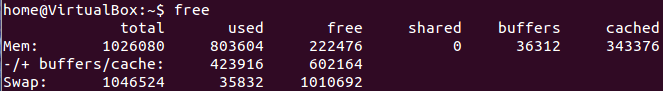
If you want the above information in a readable format, then use the command

'df -h'

[](https://www.guru99.com/images/df-h.png)

**Free**

This command shows the free and used memory (RAM) on the Linux system.

[](https://www.guru99.com/images/free.png)

You can use the arguments

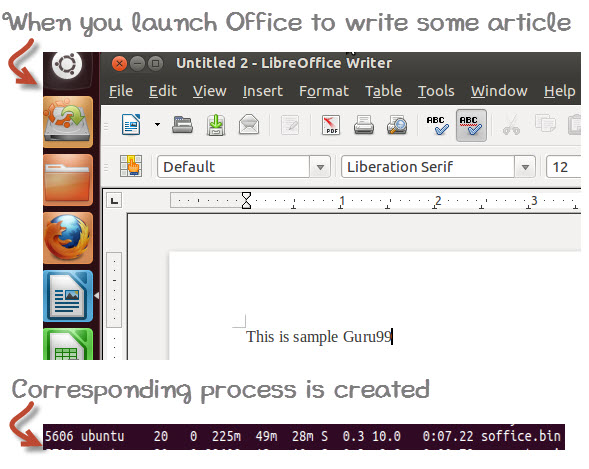
free -m to display output in MB

free -g to display output in GB

**Summary:**

* Any running program or a command given to a Linux system is called a process
* A process could run in foreground or background
* The priority index of a process is called Nice in Linux. Its default value is 0, and it can vary between 20 to -19
* The lower the Niceness index, the higher would be priority given to that task

| **Command** | **Description** |
| --- | --- |
| **bg** | To send a process to the background |
| **fg** | To run a stopped process in the foreground |
| **top** | Details on all Active Processes |
| **ps** | Give the status of processes running for a user |
| **ps PID** | Gives the status of a particular process |
| **pidof** | Gives the Process ID (PID) of a process |
| **kill PID** | Kills a process |
| **nice** | Starts a process with a given priority |
| **renice** | Changes priority of an already running process |
| **df** | Gives free hard disk space on your system |
| **free** | Gives free RAM on your system |

[](https://www.guru99.com/images/whatisprocessid.jpg)

Having multiple processes for the same program is possible.

Types of Processes:

* Foreground Processes: They run on the screen and need input from the user. For example Office Programs
* Background Processes: They run in the background and usually do not need user input. For example Antivirus.

# **VI Editor with Commands in Linux/Unix Tutorial**

## What is the VI editor?

The VI editor is the most popular and classic text editor in the Linux family. Below, are some reasons which make it a widely used editor –

1) It is available in almost all Linux Distributions

2) It works the same across different platforms and Distributions

3) It is user-friendly. Hence, millions of Linux users love it and use it for their editing needs

Nowadays, there are advanced versions of the vi editor available, and the most popular one is **VIM**which is **V**i **Im**proved. Some of the other ones are Elvis, Nvi, Nano, and Vile. It is wise to learn vi because it is feature-rich and offers endless possibilities to edit a file.

To work on VI editor, you need to understand **its operation modes**. They can be divided into two main parts.

In this tutorial, you will learn more about-

* [vi Command mode](https://www.guru99.com/the-vi-editor.html#2)
* [vi Editor Insert mode](https://www.guru99.com/the-vi-editor.html#3)
* [How to use vi editor](https://www.guru99.com/the-vi-editor.html#4)
* [vi Editing commands](https://www.guru99.com/the-vi-editor.html#5)
* [Moving within a file](https://www.guru99.com/the-vi-editor.html#6)
* [Saving and Closing the file](https://www.guru99.com/the-vi-editor.html#7)

**vi Command mode:**

[](https://www.guru99.com/images/The_vi_editor(2).jpg)

* The vi editor opens in this mode, and it only **understands commands**
* In this mode, you can, **move the cursor and cut, copy, paste the text**
* This mode also saves the changes you have made to the file
* **Commands are case sensitive.** You should use the right letter case.

**vi Editor Insert mode:**

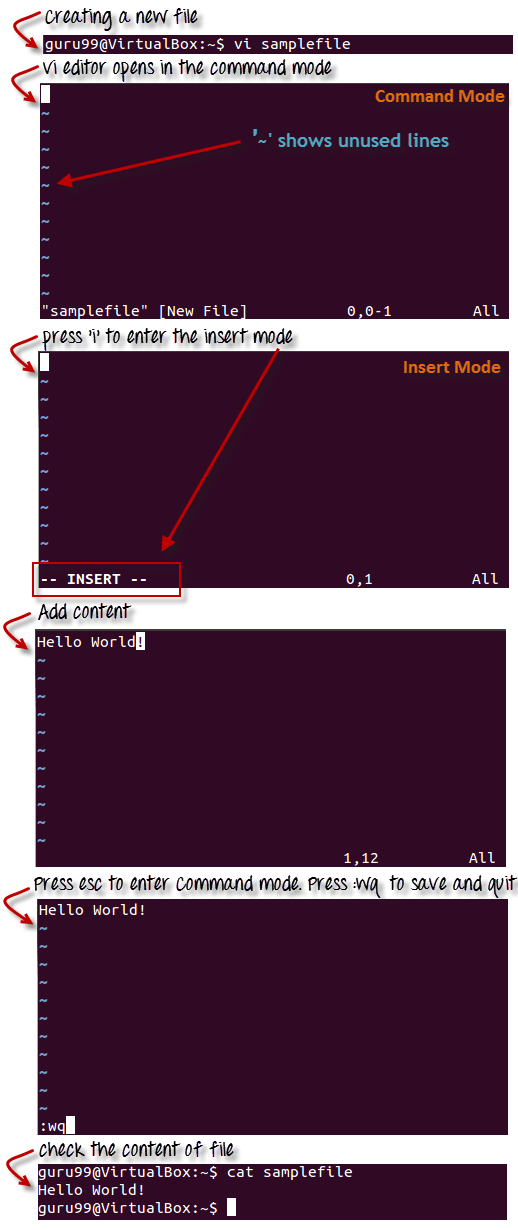
* This mode is for inserting text in the file.
* You can switch to the Insert mode from the command mode **by pressing 'i' on the keyboard**
* Once you are in Insert mode, any key would be taken as an input for the file on which you are currently working.
* To return to the command mode and save the changes you have made you need to press the Esc key

**How to use vi editor**

To launch the VI Editor -Open the Terminal (CLI) and type

vi <filename\_NEW> or <filename\_EXISTING>

And if you specify an existing file, then the editor would open it for you to edit. Else, you can create a new file.

[](https://www.guru99.com/images/Creating_a_new_file.png)

**VI Editing commands**

* i - Insert at cursor (goes into insert mode)
* a - Write after cursor (goes into insert mode)
* A - Write at the end of line (goes into insert mode)
* ESC - Terminate insert mode
* u - Undo last change
* U - Undo all changes to the entire line
* o - Open a new line (goes into insert mode)
* dd - Delete line
* 3dd - Delete 3 lines.
* D - Delete contents of line after the cursor
* C - Delete contents of a line after the cursor and insert new text. Press ESC key to end insertion.
* dw - Delete word
* 4dw - Delete 4 words
* cw - Change word
* x - Delete character at the cursor
* r - Replace character
* R - Overwrite characters from cursor onward
* s - Substitute one character under cursor continue to insert
* S - Substitute entire line and begin to insert at the beginning of the line
* ~ - Change case of individual character

**Note**: You should be in the "**command mode" to execute these commands**. VI editor is **case-sensitive** so make sure you type the commands in the right letter-case.

Make sure you press the right command otherwise you will end up making undesirable changes to the file. You can also enter the insert mode by pressing a, A, o, as required.

**Moving within a file**

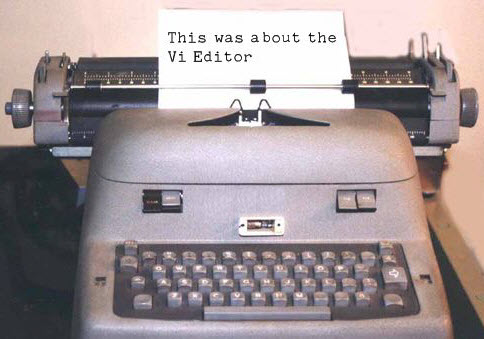
* k - Move cursor up
* j - Move cursor down
* h - Move cursor left
* l - Move cursor right

You need to be in the command mode to move within a file. The default keys for navigation are mentioned below else; You can **also use the arrow keys on the keyboard**.

**Saving and Closing the file**

* Shift+zz - Save the file and quit
* :w - Save the file but keep it open
* :q - Quit without saving
* :wq - Save the file and quit

You should be in the **command mode to exit the editor and save changes** to the file.

[](https://www.guru99.com/images/vi-editor.jpg)

**Summary:**

* The vi editor is the most popular and commonly used Unix text editor
* It is usually available in all Linux Distributions.
* It works in two modes, Command and Insert
* Command mode takes the user commands, and the Insert mode is for editing text
* You should know the commands to work on your file easily
* Learning to use this editor can benefit you in creating scripts and editing files.

# Shell Scripting Tutorial: How to Create Shell Script in Linux/Unix

## Shell Scripting

**Shell Scripting** is an open-source computer program designed to be run by the Unix/Linux shell. Shell Scripting is a program to write a series of commands for the shell to execute. It can combine lengthy and repetitive sequences of commands into a single and simple script that can be stored and executed anytime which, reduces programming efforts.

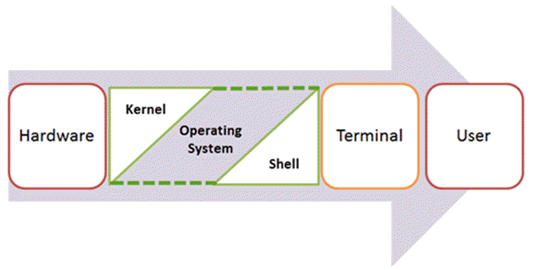
This Shell Scripting tutorial helps to learn a basic understanding of the Linux/Unix shell scripting program to advanced concepts of Shell Scripting. This Shell Script tutorial designed for beginners and professionals who want to learn What is Shell Scripting? How shell scripting works, types of shell, and more.

## What is Shell?

**Shell** is a UNIX term for an interface between a user and an operating system service. Shell provides users with an interface and accepts human-readable commands into the system and executes those commands which can run automatically and give the program’s output in a shell script.

An Operating is made of many components, but its two prime components are -

* Kernel
* Shell

[](https://www.guru99.com/images/2/ShellScripting-v1.png)

Components of Shell Program

A Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one.

A shell in a Linux operating system takes input from you in the form of commands, processes it, and then gives an output. It is the interface through which a user works on the programs, commands, and scripts. A shell is accessed by a terminal which runs it.

When you run the terminal, the Shell issues **a command prompt (usually $),** where you can type your input, which is then executed when you hit the Enter key. The output or the result is thereafter displayed on the terminal.

The Shell wraps around the delicate interior of an Operating system protecting it from accidental damage. Hence the name **Shell**.

This Unix/Linux Shell Script tutorial helps understand shell scripting basics to advanced levels.

In this Shell Script tutorial, you will learn-

* [What is Shell Scripting?](https://www.guru99.com/introduction-to-shell-scripting.html#1)
* [What is a Shell?](https://www.guru99.com/introduction-to-shell-scripting.html#2)
* [Types of Shell](https://www.guru99.com/introduction-to-shell-scripting.html#3)
* [How to Write Shell Script in Linux/Unix](https://www.guru99.com/introduction-to-shell-scripting.html#4)
* [Adding shell comments](https://www.guru99.com/introduction-to-shell-scripting.html#5)
* [What are Shell Variables?](https://www.guru99.com/introduction-to-shell-scripting.html#6)

**Types of Shell**

There are two main shells in Linux:

**1**. The **Bourne Shell**: The prompt for this shell is $ and its derivatives are listed below:

* POSIX shell also is known as sh
* Korn Shell also knew as sh
* **B**ourne **A**gain **SH**ell also knew as bash (most popular)

**2.** **The C shell**: The prompt for this shell is %, and its subcategories are:

* C shell also is known as csh
* Tops C shell also is known as tcsh

We will discuss bash shell based shell scripting in this tutorial.

**How to Write Shell Script in Linux/Unix**

**Shell Scripts** are written using text editors. On your Linux system, open a text editor program, open a new file to begin typing a shell script or shell programming, then give the shell permission to execute your shell script and put your script at the location from where the shell can find it.

Let us understand the steps in creating a Shell Script:

1. **Create a file** **using** a **vi** editor(or any other editor).  Name  script file with **extension .sh**
2. **Start** the script with **#! /bin/sh**
3. Write some code.
4. Save the script file as filename.sh
5. For **executing** the script type **bash filename.sh**

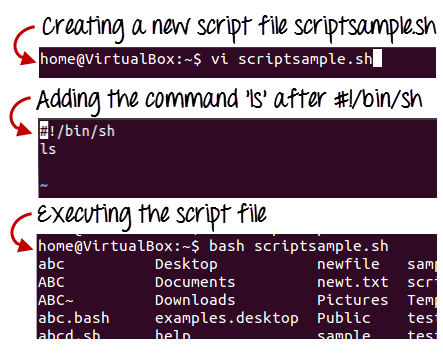
"#!" is an operator called shebang which directs the script to the interpreter location. So, if we use"#! /bin/sh" the script gets directed to the bourne-shell.

Let's create a small script -

#!/bin/sh

ls

Let's see the steps to create Shell Script Programs in Linux/Unix -

[](https://www.guru99.com/images/vi_scriptsample(2).png)Steps to Create Shell Script in Linux/Unix

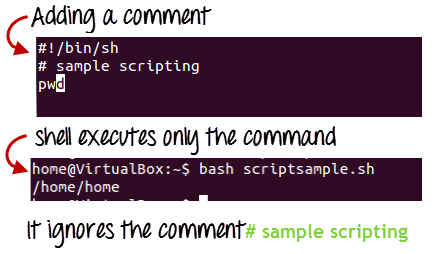
Command 'ls' is executed when we execute the scrip sample.sh file.

**Adding shell comments**

Commenting is important in any program. In Shell programming, the syntax to add a comment is

#comment

Let understand this with an example.

[](https://www.guru99.com/images/adding_comment.png)

**What are Shell Variables?**

As discussed earlier, Variables store data in the form of characters and numbers. Similarly, Shell variables are used to store information and they can by the shell only.

For example, the following creates a shell variable and then prints it:

variable ="Hello"

echo $variable

Below is a small script which will use a variable.

#!/bin/sh

echo "what is your name?"

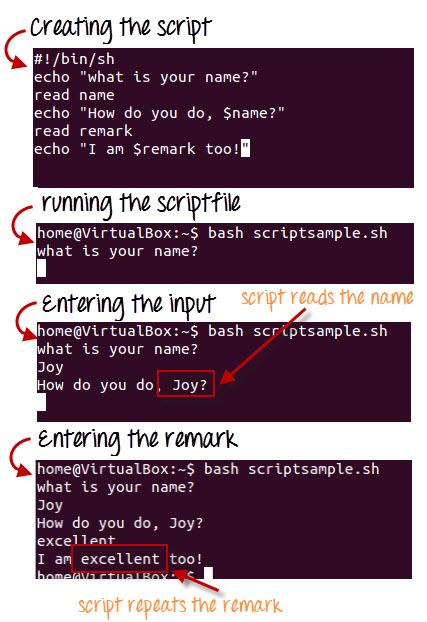
read name

echo "How do you do, $name?"

read remark

echo "I am $remark too!"

Let's understand,  the steps to create and execute the script

[](https://www.guru99.com/images/program.jpg)

As you see, the program picked the value of the variable 'name' as Joy and 'remark' as excellent.

This is a simple script. You can develop advanced scripts which contain conditional statements, loops, and functions. Shell scripting will make your life easy and Linux administration a breeze.

[](https://www.guru99.com/images/Shell(1).jpg)

**Summary:**

* Kernel is the nucleus of the operating systems, and it communicates between hardware and software
* Shell is a program which interprets user commands through CLI like Terminal
* The Bourne shell and the C shell are the most used shells in Linux
* Linux Shell scripting is writing a series of command for the shell to execute
* Shell variables store the value of a string or a number for the shell to read
* Shell scripting in Linux can help you create complex programs containing conditional statements, loops, and functions
* Basic Shell Scripting Commands in Linux: cat, more, less, head, tail, mkdir, cp, mv, rm, touch, grep, sort, wc, cut and, more.

# Linux User Commands Tutorial: Administration & Management

As Linux is a multi-user operating system, there is a high need of an administrator, who can manage user accounts, their rights, and the overall system security for User management in Linux.

You should know the basics of Linux admin so that you can handle the user accounts and usergroups for user management in Linux.

In this Linux Administration tutorial, you will learn-

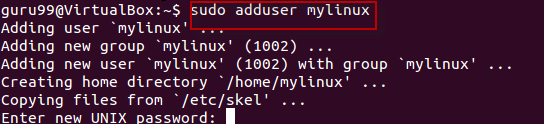
* [Creating a User](https://www.guru99.com/linux-admin.html#1)
* [Deleting, disabling account](https://www.guru99.com/linux-admin.html#2)
* [Adding users to the usergroups](https://www.guru99.com/linux-admin.html#3)
* [Finger](https://www.guru99.com/linux-admin.html#4)
* [Linux/Unix User Management Commands](https://www.guru99.com/linux-admin.html#5)

## Creating a User

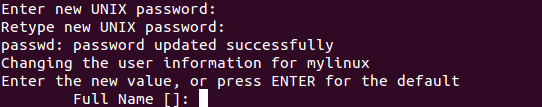
In Linux, every user is assigned an individual account which contains all the files, information, and data of the user. You can create multiple users in a Linux operating system using Linux user commands. Next in this Linux admin tutorial, we will learn how to create a user in Linux Administration. The steps for creating a user are:

### Using Terminal

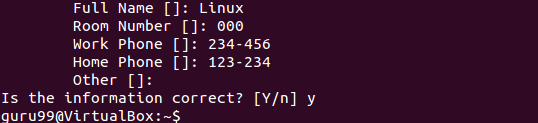
**Step 1)** Use command sudo adduser

[](https://www.guru99.com/images/sudo_adduser.png)

**Step 2)**Enter password for the new account and confirm

[](https://www.guru99.com/images/password.png)

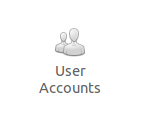
**Step 3)**Enter details of the new user and press Y

[](https://www.guru99.com/images/Name.png)

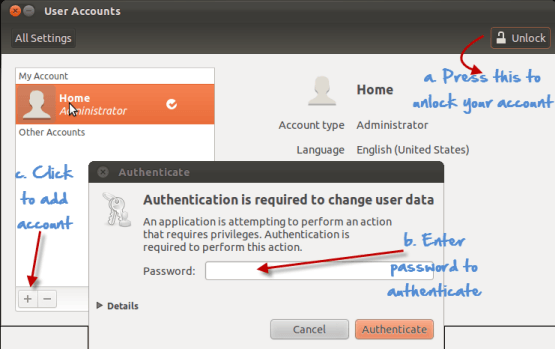
New account is created.

**Using GUI**

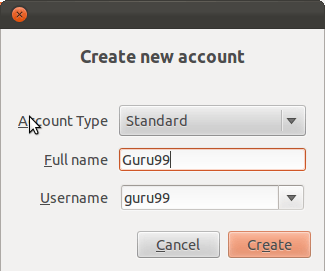
**Step 1)** Go to the system settings look for an icon which says 'User Accounts'.

[](https://www.guru99.com/images/User_Accounts.png)

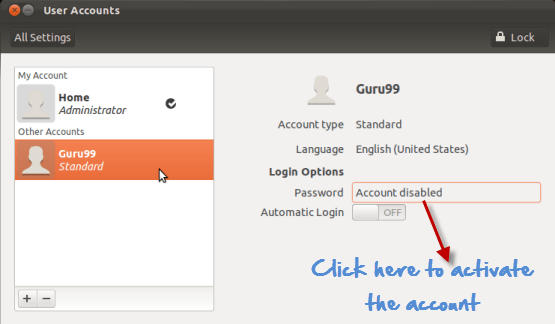
**Step 2)** Click on the unlock icon and enter a password when prompted, then click the plus sign.

[](https://www.guru99.com/images/Unlocking_account.png)

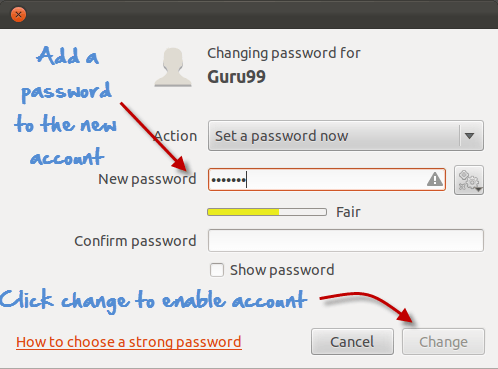
**Step 3)** A new window would pop up, asking you for adding information to the new user account.The account type offers two choices - standard and administration(Ubuntu Limitation). If you want the new user to have administrative access to the computer, select Administrator as the account type. Administrators can do things like add and delete users, install software and drivers, and change the date and time. Otherwise, choose standard.Fill in the full name, username and click on create.

[](https://www.guru99.com/images/Create_new_account.png)

**Step 4)** The new account would show, but would **be disabled by default.**

**[](https://www.guru99.com/images/Account_disabled_default.png)**

To activate it, click the password option and add a new password. Click change to enable the account.

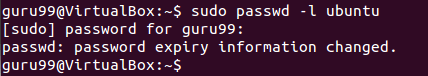
**[](https://www.guru99.com/images/enabling_account.png)**

**Deleting, disabling account**

**Terminal**

For disabling an account using Terminal, remove the password set on the account.

sudo passwd -l 'username'

**[](https://www.guru99.com/images/disabling_account_cli.png)**

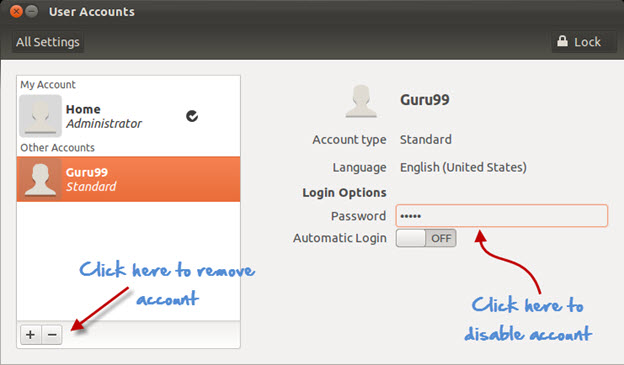
To delete an account, use the command -

sudo userdel -r 'username'

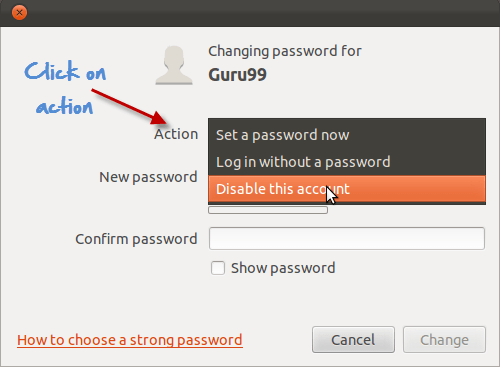
**[Unix/Linux Administration - adduser, usermod , userdel, finger](https://www.guru99.com/images/deleting_a_c.png)**

**GUI**

**Step 1)**Highlight the user account and click the minus sign to delete.

**[](https://www.guru99.com/images/disabling_account_1.jpg)**

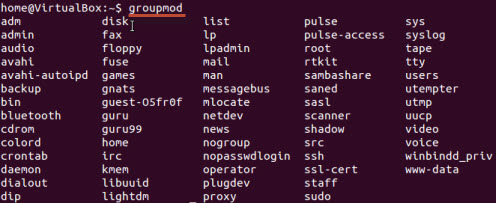
**Step 2)**For disabling click on the area where the password is stored, and you would get the following prompt. Select disable this account and click on change.

**[](https://www.guru99.com/images/disabling_account.png)**

**Adding users to the usergroups**

You can view the existing groups on your [Linux operating system](https://www.guru99.com/introduction-linux.html) by entering the following Linux user commands:

groupmod "Press Tab key twice"

**[](https://www.guru99.com/images/usergroups.jpg)**

Now to add a user to a group, use the following syntax:

sudo usermod -a -G GROUPNAME USERNAME

**[Unix/Linux Administration - adduser, usermod , userdel, finger](https://www.guru99.com/images/adding_user.png)**

The system would ask for authentication and then it would add the user to the group.

You can check whether the user is in a group by this command.

**[Unix/Linux Administration - adduser, usermod , userdel, finger](https://www.guru99.com/images/check_group.png)**

And it would show it as

**[Unix/Linux Administration - adduser, usermod , userdel, finger](https://www.guru99.com/images/show_usergropu.png)**

Removing a user from Usergroup

Use the following syntax for removing a user.

sudo deluser USER GROUPNAME

**[Unix/Linux Administration - adduser, usermod , userdel, finger](https://www.guru99.com/images/sudoDeleteuser.jpg)**

**The GUI method**

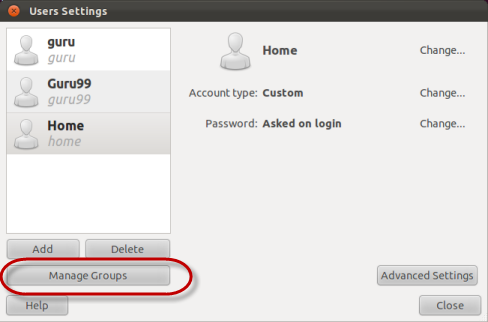
If you do not want to run the  Linux user commands in terminal to manage users and groups, then you can install a GUI add-on .

sudo apt-get install gnome-system-tools

Once done, type

users-admin

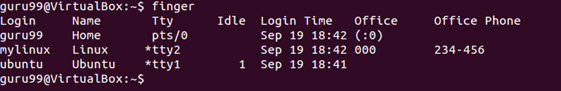
Check user settings, and a tab Manage Groups will appear-

**[](https://www.guru99.com/images/users-admin.png)**

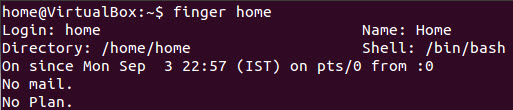
**Finger**

This command is used to **procure information of the users on a Linux machine**. You can use it on both local & remote machines

The syntax 'finger' gives data on all the logged users on the remote and local machine.

**[](https://www.guru99.com/images/finger.jpg)**

The syntax 'finger username' specifies the information of the user in User administration in Linux.

**[](https://www.guru99.com/images/finger_home.jpg)**

**Linux/Unix user management commands**

User management in Linux is done by using Linux administration commands. Here is a list of user management commands in Linux:

|  |  |
| --- | --- |
| **Command** | **Description** |
| sudo adduser username | Adds a user |
| sudo passwd -l 'username' | Disable a user |
| sudo userdel -r 'username' | Delete a user |
| sudo usermod -a -G GROUPNAME USERNAME | Add user a to a usergroup |
| sudo deluser USER GROUPNAME | Remove user from a user group |
| finger | Gives information on all logged in user |
| finger username | Gives information of a particular user |

**Summary:**

* You can use both GUI or Terminal for User Administration in Linux User management
* You can create, disable and remove user accounts using [Linux](https://www.guru99.com/introduction-linux.html) admin commands.
* You can add/delete a user to a usergroup.

# **Unix Vs. Linux: What’s the Difference Between Unix and Linux?**

## What is UNIX?

The UNIX OS was born in the late 1960s. AT&T Bell Labs released an operating system called Unix written in C, which allows quicker modification, acceptance, and portability.

It began as a one-man project under the leadership of Ken Thompson of Bell Labs. It went on to become most widely used operating systems. Unix is a proprietary operating system.

The Unix OS works on CLI (Command Line Interface), but recently, there have been developments for GUI on Unix systems. Unix is an OS which is popular in companies, universities big enterprises, etc.

## What is LINUX?

Linux is an operating system built by Linus Torvalds at the University of Helsinki in 1991. The name "Linux" comes from the Linux kernel. It is the software on a computer which enables applications and the users to access the devices on the computer to perform some specific function.

The Linux OS relays instructions from an application from the computer's processor and sends the results back to the application via the Linux OS. It can be installed on a different type of computers mobile phones, tablets video game consoles, etc.

The development of Linux is one of the most prominent examples of free and open source software collaboration. Today many companies and similar numbers of individuals have released their own version of OS based on the Linux Kernel.

## KEY DIFFERENCE

* Comparing Unix vs Linux, Linux source code is available to the general public whereas, in Unix, the source code is proprietary.
* UNIX OS was created in the late 1960s at AT&T Bell Labs whereas Linux is an operating system built by Linus Torvalds at the University of Helsinki in 1991.
* The main difference between Linux and Unix is Linux is a clone of Unix
* When we compare Unix to Linux, Linux default shell is BASH while the Unix shell is Bourne Shell.
* One key Unix and Linux difference is that Linux threat detection and solution are very fast while Unix users require longer wait times to get the proper bug fixing patch.
* Important versions of Linux are Redhat, Ubuntu, OpenSuse, Solaris, whereas important versions of Unix are HP-UX, AIS, BSD, etc.

## Features of Unix OS

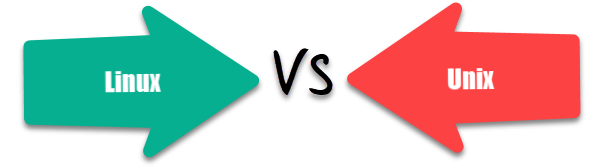
* When compare Unix to Linux, Unix is a Multi-user, multitasking operating system
* It can be used as the master control program in workstations and servers.
* Hundreds of commercial applications are available
* In its heydays, UNIX was rapidly adopted and became the standard OS in universities.

## Features of Linux Operating System

* Support multitasking
* Programs consist of one or more processes, and each process have one or more threads
* It can easily co-exists along with other Operating systems.
* Comparing features of Linux and Unix, Linux can run multiple user programs
* Individual accounts are protected because of appropriate authorization
* When comparing Unix vs Linux, Linux is a replica of UNIX but does not use its code.

## Difference between Unix and Linux

Now, we will see what is the difference between unix and linux:

[](https://www.guru99.com/images/1/053018_0703_UnixVsLinux1.png)Unix Vs Linux

| **Key Differences** | **Linux** | **Unix** |
| --- | --- | --- |
| **Cost** | Linux is freely distributed, downloaded through magazines, Books, website, etc. There are paid versions also available for Linux. | Different flavors of Unix have different pricing depending upon the type of vendor. |
| **Development** | Linux is Open Source, and thousands of programmer collaborate online and contribute to its development. | Unix systems have different versions. These versions are primarily developed by AT&T as well as other commercial vendors. |
| **User** | Everyone. From home users to developers and computer enthusiasts alike. | The UNIX can be used in internet servers, workstations, and PCs. |
| **Text made interface** | BASH is the Linux default shell. It offers support for multiple command interpreters. | Originally made to work in Bourne Shell. However, it is now compatible with many others software. |
| **GUI** | Linux provides two GUIs,viz., KDE and Gnome. Though there are many alternatives such as Mate, LXDE, Xfce, etc. | Common Desktop Environment and also has Gnome. |
| **Viruses** | Linux has had about 60-100 viruses listed to date which are currently not spreading. | There are between 80 to 120 viruses reported till date in Unix. |
| **Threat detection** | Threat detection and solution is very fast because Linux is mainly community driven. So, if any Linux user posts any kind of threat, a team of qualified developers starts working to resolve this threat. | Unix users require longer wait time, to get the proper bug fixing patch. |
| **Architectures** | Initially developed for Intel's x86 hardware processors. It is available for over twenty different types of CPU which also includes an ARM. | It is available on PA-RISC and Itanium machines. |
| **Usage** | Linux OS can be installed on various types of devices like mobile, tablet computers. | The UNIX operating system is used for internet servers, workstations & PCs. |
| **Best feature** | Kernel update without reboot | Feta ZFS - next generation filesystem DTrace - dynamic Kernel Tracing |
| **Versions** | Different Versions of Linux are Redhat, Ubuntu, OpenSuse, etc. | Different Versions of Unix are HP-UX, AIS, BSD, etc. |
| **Supported file type** | The Filesystems supported by file type like xfs, nfs, cramfsm ext 1 to 4, ufs, devpts, NTFS. | The Filesystems supported by file types are zfs, hfx, GPS, xfs, vxfs. |
| **Portability** | Linux is portable and is booted from a USB Stick | Unix is not portable |
| **Source Code** | The source is available to the general public | The source code is not available to anyone. |

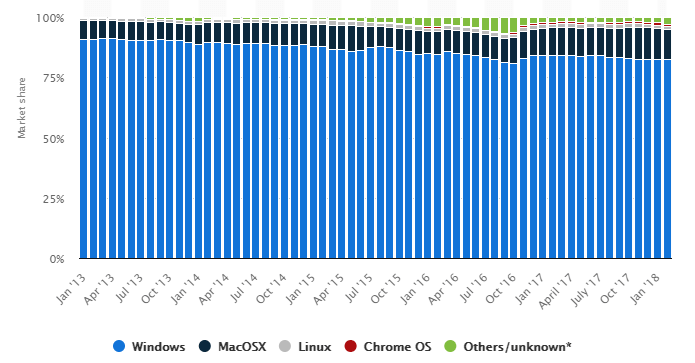
## Limitation of Linux

* For Linux vs Unix, There's no standard edition of Linux
* Linux has patchier support for drivers which may result in misfunctioning of the entire system.
* Linux is, for new users at least, not as easy to use as Windows.
* Many of the programs we are using for Windows will only run on Linux only with the help of a complicated emulator. For example. Microsoft Office.
* Linux is best suitable for a corporate user. It's much harder to introduce in a home setting.

## Limitations of Unix

* The unfriendly, terse, inconsistent, and non-mnemonic user interface
* Comparing limitation of Linux vs Unix, Unix OS is designed for a slow computer system, so you can't expect fast performance.
* Shell interface can be treacherous because typing mistake can destroy files.
* Versions on various machines are slightly different, so it lacks consistency.
* One of the Unix and Linux difference is, Unix does not provide any assured hardware interrupt response time, so it does not support real time response time systems.

**Linux Market share compared to other OS**

[](https://www.guru99.com/images/1/053018_0703_UnixVsLinux2.png)

# **Crontab in Linux: Job Scheduling EXAMPLES**

## What is crontab?

Cron is named after Greek word "Chronos" that is used for time. It is a system process that will automatically perform tasks as per the specific schedule. It is a set of commands that are used for running regular scheduling tasks. Crontab stands for "cron table". It allows to use job scheduler, which is known as cron to execute tasks.

Crontab is also the name of the program, which is used to edit that schedule. It is driven by a crontab file, a config file that indicates shell commands to run periodically for the specific schedule.

In this Operating system tutorial, you will learn:

* [What is crontab?](https://www.guru99.com/crontab-in-linux-with-examples.html#1)
* [Why use CRONJOB?](https://www.guru99.com/crontab-in-linux-with-examples.html#2)
* [How to use cron in Linux?](https://www.guru99.com/crontab-in-linux-with-examples.html#3)
* [Linux Crontab format](https://www.guru99.com/crontab-in-linux-with-examples.html#4)
* [How to Add/Modify Crontab](https://www.guru99.com/crontab-in-linux-with-examples.html#5)
* [How to List Crontab](https://www.guru99.com/crontab-in-linux-with-examples.html#6)
* [Important Crontab Examples](https://www.guru99.com/crontab-in-linux-with-examples.html#7)

## Why use Cronjobs?

Here are the reasons for using Cronjobs in Linux:

* Helps OS to take a scheduled backup of log files or database.
* Delete old log files
* Archive and purge database tables
* Send out any notification email such as Newsletters, Password expiration email
* Regular clean-up of cached data
* Crontab is an ideal option to automate Unix jobs.
* It is used to automate system maintenance

## How to use cron in Linux?

Linux system pack has a useful task scheduler named crontab. Crontab is popular because it can be scheduled to run an automated process as root. Therefore, having an automated process running as root makes system changes easier. You just need to change the task and then wait until the task is re-initiated.

## Linux Crontab format

Crontab of Linux has six fields. The first five fields define the time and date of execution, and the 6'th field is used for command execution.

Crontab syntax:

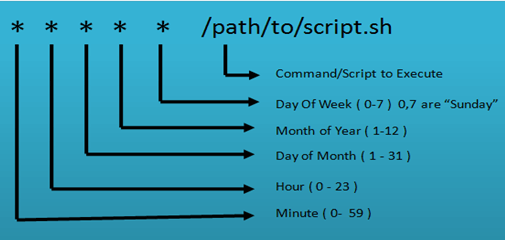
[Minute] [hour]

[Day\_of\_the\_Month]

[Month\_of\_the\_Year]

[Day\_of\_the\_Week]

[command]

[](https://www.guru99.com/images/1/011720_0741_CrontabinLi1.png)

* **Astrics (\*):** Use for matching
* **Define range:** Allows you to define a range with the help of hyphen like 1-10 or 30-40 or jan-mar, mon-wed.
* **Define multiple ranges:** Allows you to define various ranges with command separated like apr-jun,oct-dec.

## How to Add/Modify Crontab

User can edit their crontab jobs with the help of following crontab command:

$ crontab -u -e

The above command will open the personal crontab configuration of your computer system, which can be edited by using your default text editor.

There is no need to restart your crontab as it will pick up your changes automatically when you use following command.

$ crontab -l

To remove your crontab tasks, use the following command.

$ crontab -r

To add or update job in crontab, use below given command.

crontab -e

Command to edit other user's crontab

crontab -u username -e

### How to List Crontab

Command to view crontab entries of current user

crontab -l

Command to view crontab entries of a specific user:

crontab -u username -l

## Important Crontab Examples

Here, are some important examples of Crontab

|  |  |
| --- | --- |
| **Description** | **Command** |
| Cron command to do the various scheduling jobs. Below given command execute at 7 AM and 5 PM daily. | 0 7,17 \* \* \* /scripts/script.sh |
| Command to execute a cron after every 5 minutes. | \*/5\* \* \* \* \* /scripts/script.sh |
| Cron scheduler command helps you to execute the task on every Monday at 5 AM. This command is helpful for doing weekly tasks like system clean-up. | 0 5 \* \* mon /scripts/script.sh |
| Command run your script on 3 minutes interval. | \*/3 \* \* \* \* /scripts/monitor.sh |
| Command to schedule a cron to which executes for a specific month. This command to run tasks run in Feb, June and September months. Sometimes we need to schedule a task to execute a select monthly task. | \* \* \* feb,jun,sep \* /script/script.sh |
| Command to execute on selected days. This example will run each Monday and Wednesday at 5 PM. | 0 17 \* \* mon,wed /script/script.sh |
| This command allows cron to execute on first Saturday of every month. | 0 2 \* \* sat [ $(date +%d) -le 06 ] && /script/script.sh |
| Command to run a script for 6 hours interval so it can be configured like below. | 0 \*/6 \* \* \* /scripts/script.sh |
| This command schedule a task to execute twice on Monday and Tuesday. Use the following settings to do it. | 0 4,17 \* \* mon,tue /scripts/script.sh |
| Command schedule a cron to execute after every 15 Seconds. | \* \* \* \* \* /scripts/script.sh  \* \* \* \* \* sleep 15; /scripts/script.sh |
| Command to schedule tasks on a yearly basis. @yearly timestamp is= to "0 0 5 1 \*". This executes the task in the fifth minute of every year. You can use it to send for new year greetings. | @yearly /scripts/script.sh |
| Command tasks to execute on a monthly basis. @monthly timestamp is similar to "0 0 1 \* \*". This command expression allows the execution of a task in the first minute of the month. | @monthly /scripts/script.sh |
| Command to execute multiple tasks using a single cron. | \* \* \* \* \* /scripts/script.sh; /scripts/scrit2.sh |
| Command to schedule tasks to execute on a weekly basis. @weekly timestamp is similar to "0 0 4 \* sun". This is used to perform the weekly tasks like the system cleanup etc. | @weekly /bin/script.sh |
| Task will be scheduled to execute on a daily basis. @daily timestamp is similar to "0 2 \* \* \*". It executes the task in the second minute of every day. | @daily /scripts/script.sh |
| Allows tasks to execute on an hourly. @hourly timestamp is similar to "0 \* \* \* \*". This command executes a task in the first minute of every hour. | @hourly /scripts/script.sh |
| Allows tasks to execute on system reboot. @reboot expression is useful for those tasks that the system wants to run on your system startup. This is helpful to begin tasks background automatically. | @reboot /scripts/script.sh |

### Summary:

* Cron is named after Greek word "Chronos" that is used for time.
* Cronjobs help OS to take a scheduled backup of log files or database.
* Linux system pack has a useful task scheduler named crontab that can be scheduled to run an automated process as root.
* Crontab of Linux has six fields. The first five fields define the time and date of execution, and the 6'th field is used for command execution.

https://www.guru99.com/best-linux-certifications.html