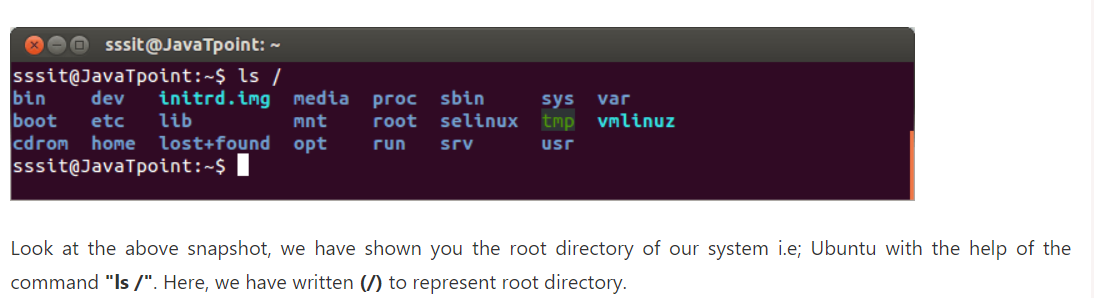
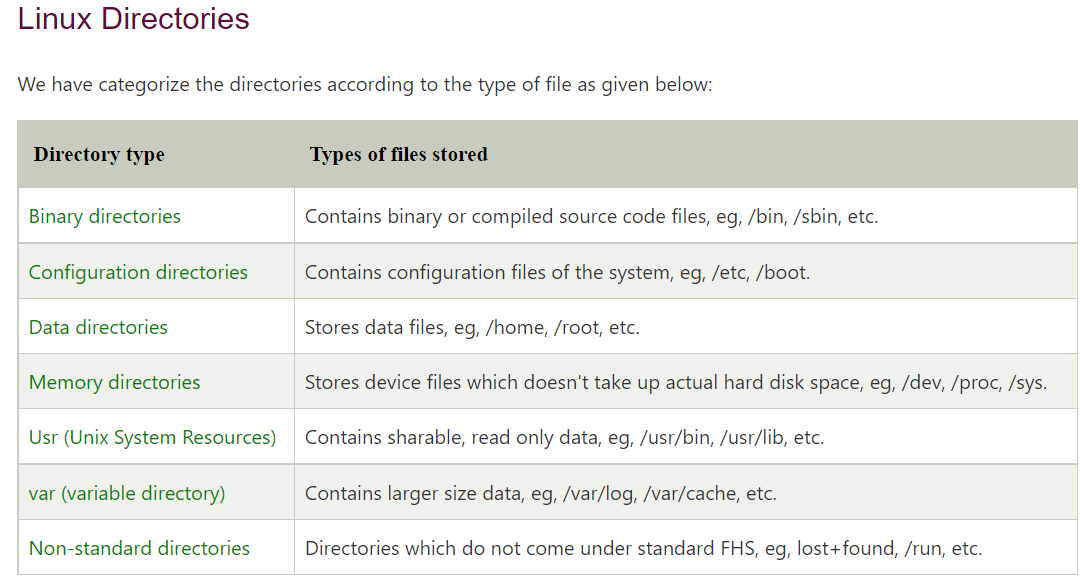
**Linux File Hierarchy Structure**

The Linux File Hierarchy Structure or the Filesystem Hierarchy Standard (FHS) defines the directory structure and directory contents in Unix-like operating systems. It is maintained by the Linux Foundation.

## **The Root Directory**





# Linux Binary Directory

Binary files are the files which contain compiled source code (or machine code). They are also called executable files because they can be executed on the computer.

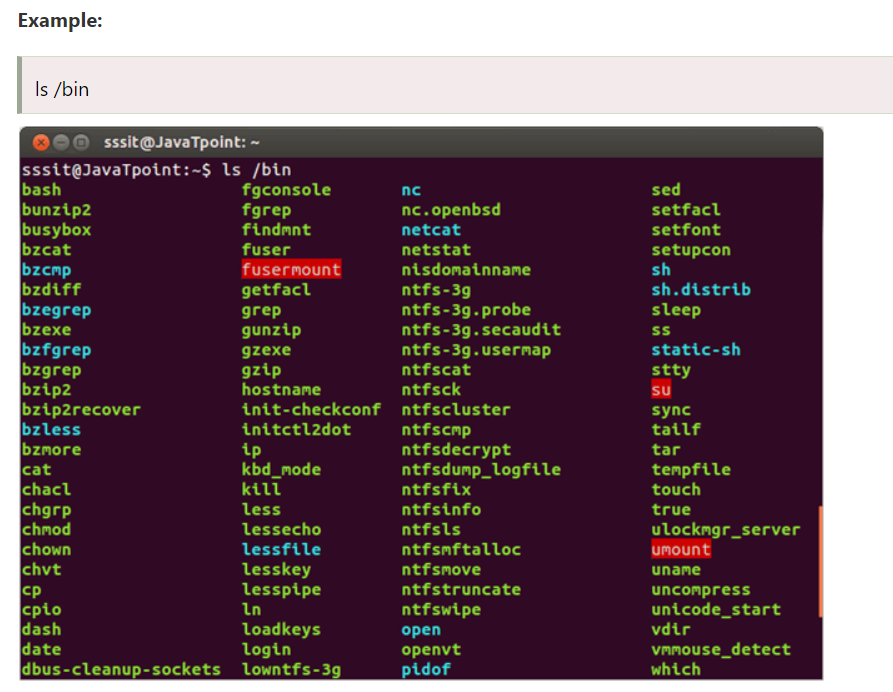
Binary directory contains following directories:

* /bin
* /sbin
* /lib
* /opt

## **/bin**

The '/bin' directory contains user binaries, executable files, Linux commands that are used in single user mode, and common commands that are used by all the users, like cat, cp, cd, ls, etc.

The '/bin' directory doesn't contain directories.



## **/sbin**

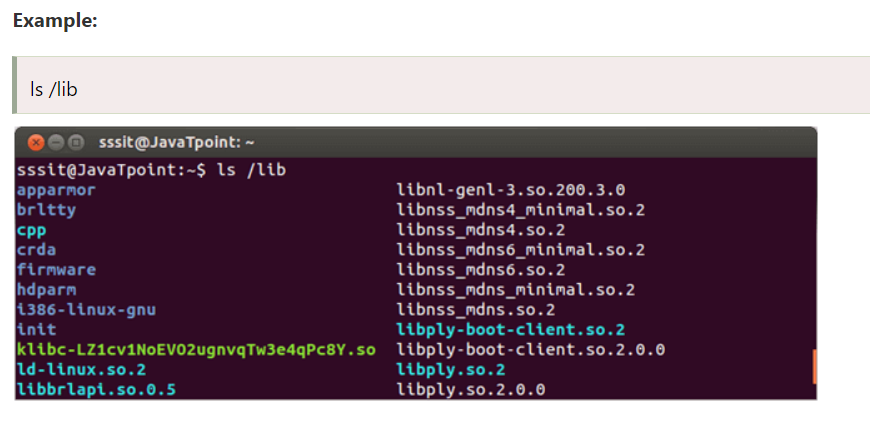
The '/sbin' directory also contains executable files, but unlike '/bin' it only contains system binaries which require root privilege to perform certain tasks and are helpful for system maintenance purpose. e.g. fsck, root, init, ifconfig, etc.

**Example:**



## **/lib**

The '/lib' directory contains **shared libraries** which are often used by the '/bin' and '/sbin' directories. It also contains kernel module. These filenames are identable as ld\* or lib\*.so.\*. For example, ld-linux.so.2 and libfuse.so.2.8.6



Look at the above snapshot, command **"ls /lib"** displays the list of '/lib' directory.

## **/opt**

The term **'opt'** is short for optional. Its main purpose is to store optional application software packages. Add-on applications from individual vendors should be installed in '/opt'.

# Linux Configuration Directory

Configuration directory have following sub-diectories:

* **/boot**
* **/etc**

## **/boot**

The '/boot' directory contains boot loader files which are essential to boot the system. In other words, they only contain files which are needed for a basic Linux system to get up and going.

## **/etc**

All the machine related configurtion files are kept in '/etc'. Almost everything related to the configuration of your system is placed here. It also contain startup and shutdown shell script which is used to start and stop a program. All the files are static and text based and no binary files can be placed in this directory

## Configuration files will have an extension of **.conf**. **Some common directories of /etc are:**

* **/etc/init.d/**: The term 'init' is short for **initialization**. This directory contains script to control the system or to start and stop the daemons (background process). The 'init' is a daemon process that continues running until the system is shut down.
* **/etc/X11/**: The X Window system configuration files are stored in this directory. The configuration file of graphical display (xorg.conf) is also stored here.
* **/etc/skel/**: The term 'skel' is short for **skeleton**. Everything in the system has a skeleton which is called hidden file and is stored in this directory. It is not an important part in the system and can be deleted but still it serves a specific purpose. Its purpose is to serve the basic set of files, a basic framework which can be used in the creation of a new user.

Linux Data directory

Data directory is used to store data of the system.

Data directory contains following directories.

* /home
* /root
* /srv
* /media
* /mnt
* /tmp

## **/home**

The '/home' directory stores users personnel files. After the '/home' there is a directory which is generally named at the user's name like we have **'/home/sssit'**. Inside this directory we have our sub-directories like Desktop, Downloads, Documents, pictures, etc.

**Example:**

## **/root**

The '/root' directory is the home directory of the root user.

Please note that '/root' directory is different from (/) root.

## **/srv**

The term 'srv' is short for **service**. The '/srv' directory contains server specific data for services provided by the system like www, cvs, rysync, ftp, etc.

## **/media**

The '/media' directory acts as a mount point for removable media devices such as CD-Rom, floppy, USB devices, etc.

This is newly introduced directory and hence a system can run without this directory also.

## **Example:** **/mnt**

The term 'mnt' stands for **mount**. The '/mnt' directory should be empty and sysadmins can only mount temporary filesystems.

## **/tmp**

The term 'tmp' stands for **temporary**. Data stored in '/tmp' is temporary and may use either disk space or RAM. When system is rebooted, files under this directory is automatically deleted. So it is advisable that never use '/tmp' to store important data.

Linux Memory Directory

Memory directory contains files of the whole system. All the device information, process running indata or system related information are stored in this directory.

Memory directory contains the following directories.

* /dev
* /proc
* /sys

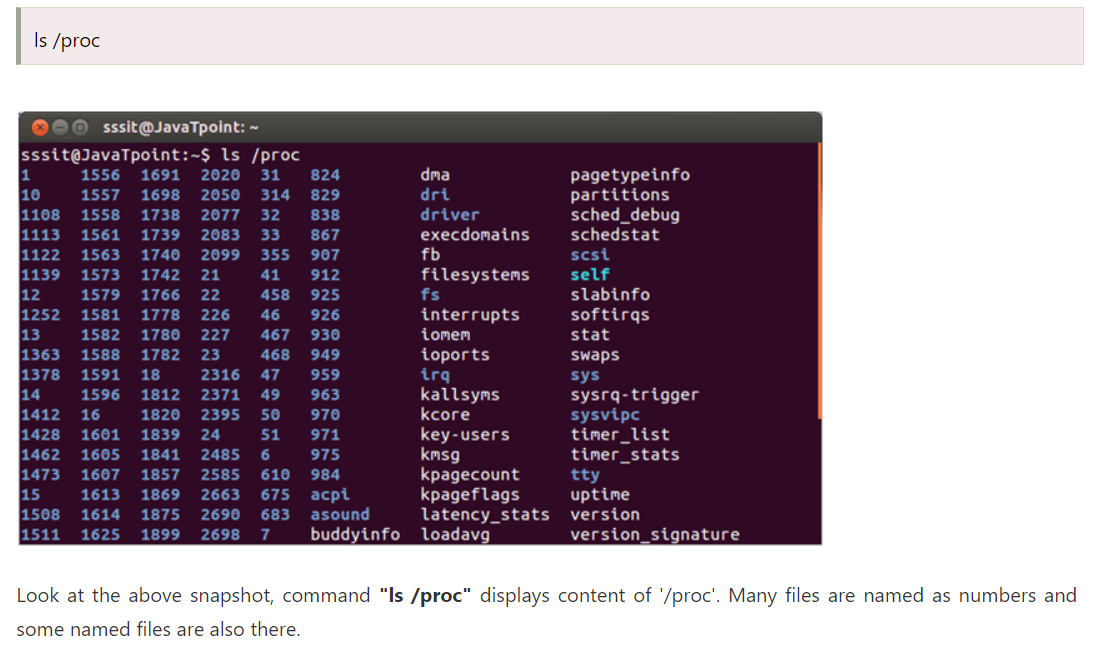
## **/dev**

The term 'dev' is short for **device**. As you know in Linux operating system everything is a file. It appears to be an ordinary file but doesn't take up disk space. Files which are used to represent and access devices are stored here including terminal devices like usb. All the files stored in '/dev' are not related to real devices, some are related to virtual devices also.

* **/dev/tty and /dev/pts**: The '/dev/tty' file represents the command line interface that is a terminl or console attached to the system. Typing commands in a terminal is a part of the graphical interface like Gnome or KDE, then terminal will be represented as '/dev/pts/1' (here 1 is replacable by any another number).
* **/dev/null**: The '/dev/null' file is considered as black hole, it has unlimited storage but nothing can be retrieved from it. You can discard your unwanted output from the terminal but can't retrieve it back.

## **/proc**

The term 'proc' is short for process. Same as '/dev', '/proc' also doesn't take up disk space. It contains process information. It is a pseudo filesystem that contains information about running processes. It also works as virtual filesystem containing text information about system resources.

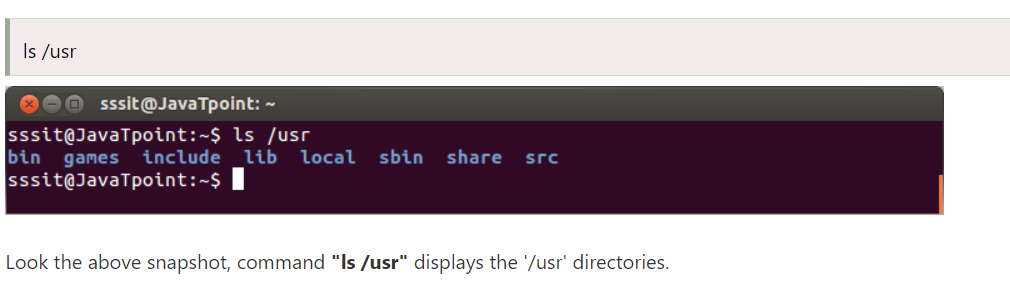
* **/proc conversation with the kernel**: The '/proc' displays view of the kernel, what the kernel manages and it is a means to directly communicate with the kernel.
* **Example:** **/proc/interrupts**: The '/proc/interrupts' displays the interrupt.

## **/sys**

The term 'sys' is short for **system**. Basically it contains kernel information about hardware. It was created for Linux 2.6 kernel. It is a kind of '/proc' and is used for plug and play configuration.

# Unix System Resources (/usr)

Although it is pronounced as user but in actual it stands for **Unix System Resources**. It is also called secondary hierarchy as it contains binaries, libraries, documentation for all the user applications. It only contains shareable read-only data.

**Example:** Look the above snapshot, command **"ls /usr"** displays the '/usr' directories.

We'll explain some of the /usr sub-directories:

* /usr/bin
* /usr/include
* /usr/lib
* /usr/share
* /usr/local
* /usr/src

## **/usr/bin**

The '/usr/bin' directory contains non-essential binary commands for all users. If you can't find a command in '/bin', search it in '/usr/bin'. It contains a lot of commands.

## **/usr/include**

The '/usr/include' directory contains standard include files for C.

## **/usr/lib**

The '/usr/lib' directory contains libraries that are not directly executed by the users. In other words, it contains binaries for the '/usr/bin' and '/usr/sbin'.

## **/usr/share**

The '/usr/share' directory contains architecture independent (shared) data.

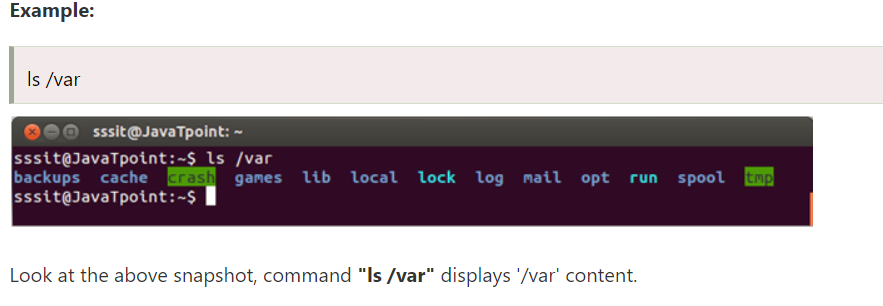
## **/usr/local**

The '/usr/local' directory is used to install software locally. It means all the user programs that you'll install from source will be installed here.

## **/usr/src**

The term 'src' is short for **source**. It is used to store source code like kernel source code with its header files.

# Variable Directory (/var)

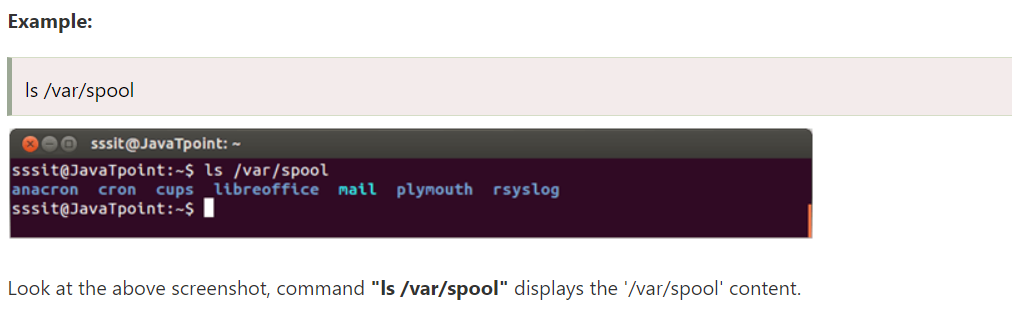
* The term 'var' is short for **variable**. Files that have an unexpected size and whose content is expected to change continuously (that's why it is named as variable) during normal operation of the system are stored here. For example, log files, spool files and cache files.  /var sub-directories/var/log
* /var/cache
* /var/spool
* /var/lib

## **/var/log**

The '/var/log' directory contains all log files.

## **/var/spool**

The '/var/spool' directory is used to spool the files waiting to be processed. For example, printing queues and mail queues.



## **/var/lib**

The '/var/lib' directory stores the files that contains state information like databases. File's data modifies as their respective programs run.

Non-Standard Directories

Directories which do not come under the standard FHS are called non-standard directories.

Non-standard directories are as follows:

* /cdrom
* /run
* /lost + found

## **/cdrom**

The '/cdrom' directory is not in the standard FHS but cdrom can be mounted on this directory. Ideally according to standard FHS cdrom should be mounted under '/media'.

## **/run**

The '/run' directory stores run-time variable data. Run-time vriable data means, data about the running system since last boot. For eg, running daemons.

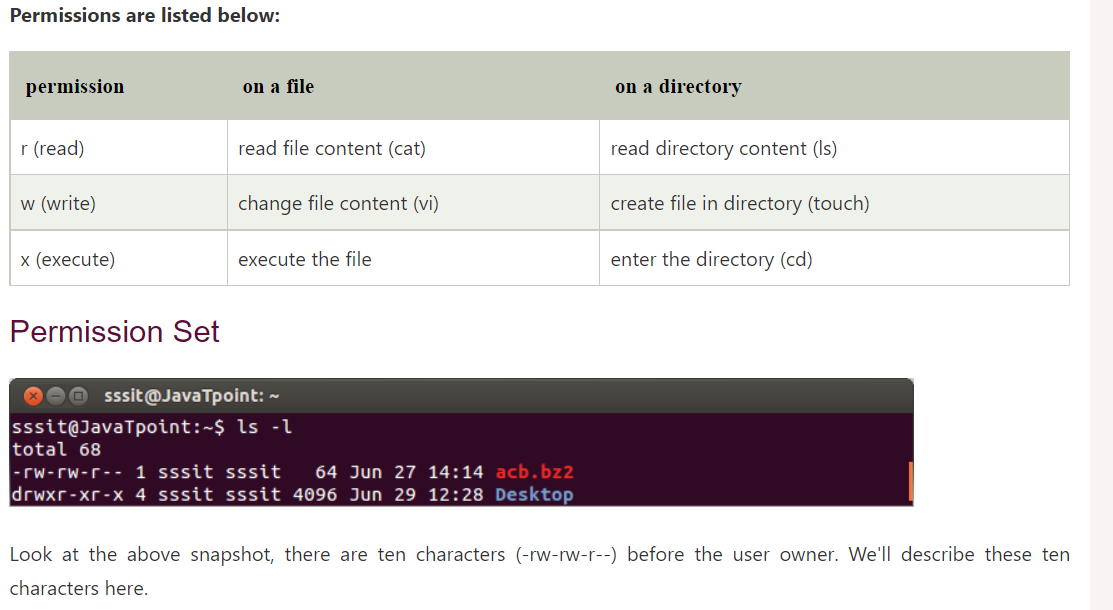
## **/lost + found**

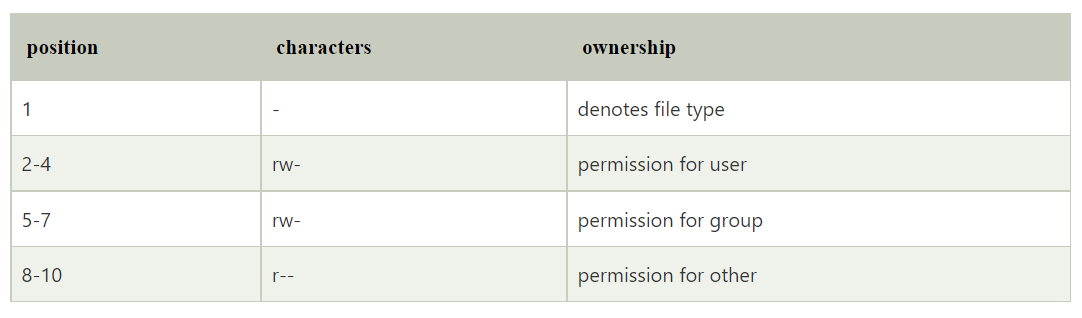
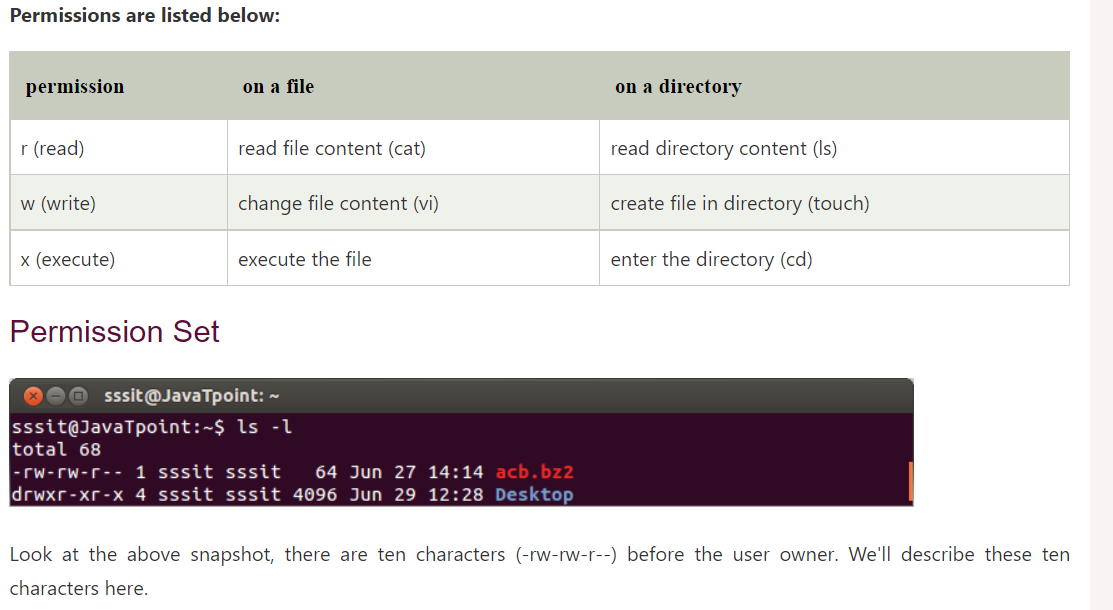
During system crash or in any other situation when Linux file system checker (fsck) recovers lost data, that data is stored in this directory. Data may or may not be in a good condition.

# File Permissions

All the three owners (user owner, group, others) in the Linux system have three types of permissions defined. Nine characters denotes the three types of permissions.

1. **Read (r) :** The read permission allows you to open and read the content of a file. But you can't do any editing or modification in the file.
2. **Write (w) :** The write permission allows you to edit, remove or rename a file. For instance, if a file is present in a directory, and write permission is set on the file but not on the directory, then you can edit the content of the file but can't remove, or rename it.
3. **Execute (x):** In Unix type system, you can't run or execute a program unless execute permission is set.But in Windows, there is no such permission available.

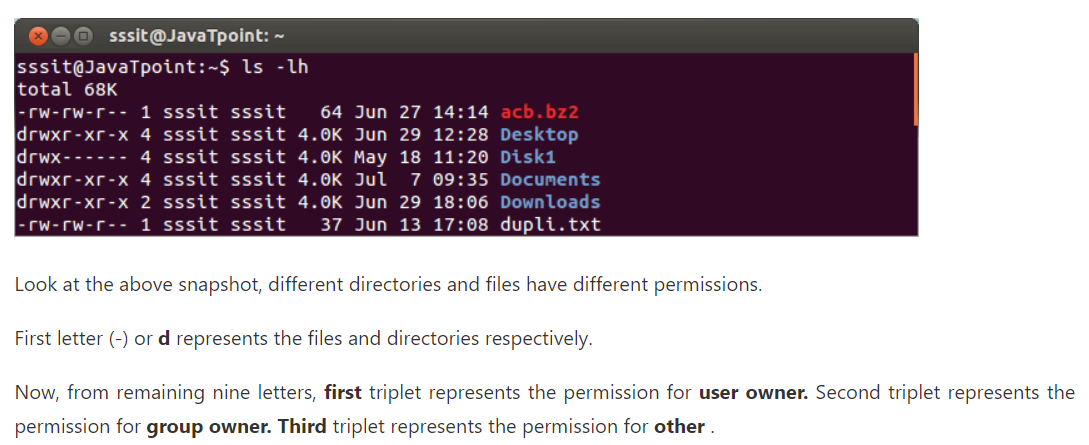


When you are the **User owner**, then the user owner permission applies to you. Other permissions are not relevant to you.

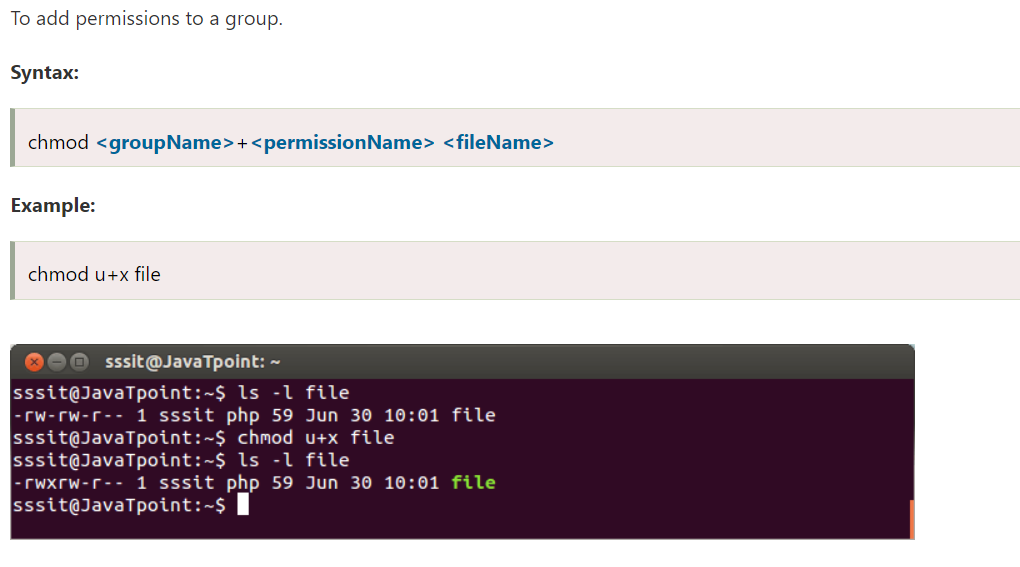
When you are the **Group** then the group permission applies to you. Other permissions are not relevant to you.

When you are the**Other,** then the other permission applies to you. User and group permissions are not relevant to you.

**Permission Example**



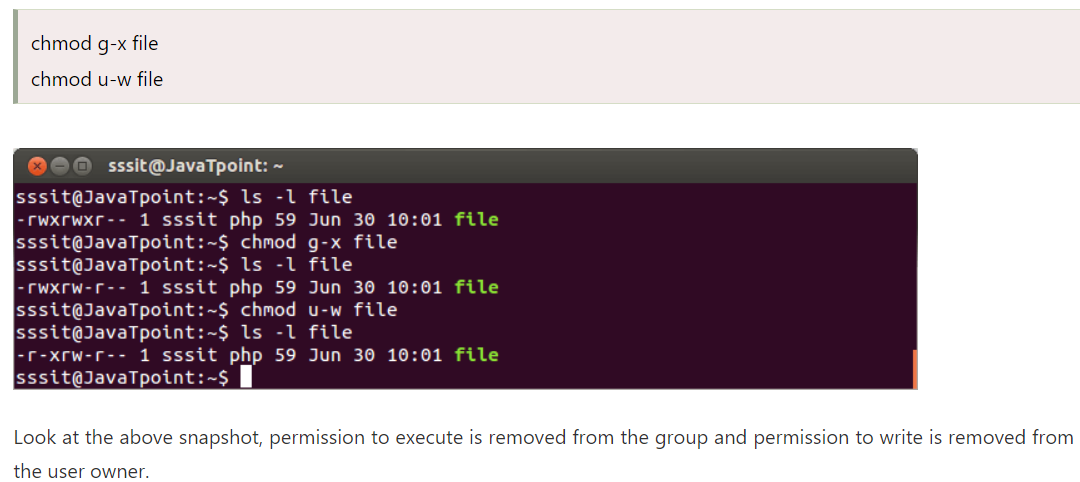
## **Setting Permissions With chmod**

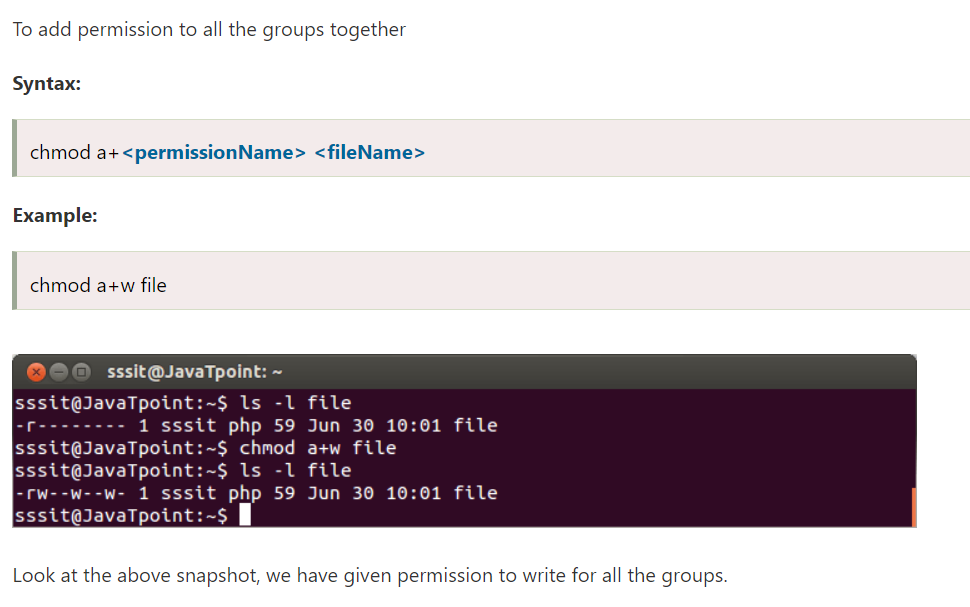


Look at the above snapshot, permission to execute is added to the user owner group.

To remove permissions from a group

**Syntax:**





To add permission to all the groups without typing a

