What is Linux?

Linux is an open source operating system (OS). It was originally conceived of and created as a hobby by Linus Torvalds in 1991. Linus, while at university, wanted to create an alternative, free, open source version of the MINIX operating system, which was itself based on the principles and design of Unix. That hobby has since become the OS with the largest user base, the most-used OS on publicly available internet servers, and the only OS used on the top 500 fastest supercomputers.

Perhaps the best thing about Linux is that it's open source. Linux is released under the GNU General Public License (GPL). That means that anyone can run, study, share, and modify the software. The modified code can also be redistributed, and even sold, but must be done so under the same license. This differs greatly from traditional operating systems—Unix and Windows, for example—which are proprietary, locked-down, and delivered as-is and unmodifiable.

Open source

Open source is a term that originally referred to open source software (OSS). Open source software is code that is designed to be publicly accessible—anyone can see, modify, and distribute the code as they see fit.

Open source software is developed in a decentralized and collaborative way, relying on peer review and community production. Open source software is often cheaper, more flexible, and has more longevity than its proprietary peers because it is developed by communities rather than a single author or company.

Open source has become a movement and a way of working that reaches beyond software production. The open source movement uses the values and decentralized production model of open source software to find new ways to solve problems in their communities and industries.

Operating System

Linux is an open source operating system (OS). An operating system is the software that directly manages a system's hardware and resources, like CPU, memory, and storage. The OS sits between applications and hardware and makes the connections between all of your software and the physical resources that do the work.

Think about an OS like a car engine. An engine can run on its own, but it becomes a functional car when it's connected with a transmission, axles, and wheels. Without the engine running properly, the rest of the car won't work.

GNU General Public License (GPL)

Published software should be free software. To make it free software, you need to release it under a free software license. We normally use the GNU General Public License (GNU GPL), specifying version 3 or any later version, but occasionally we use other free software licenses. We use only licenses that are compatible with the GNU GPL for GNU software.

Linux’s root in Unix

The Unix operating system is a set of programs that act as a link between the computer and the user.

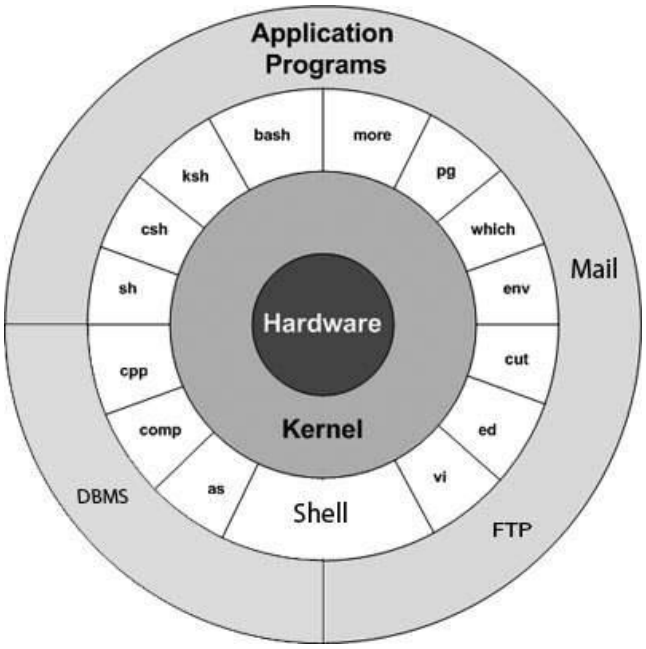
The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the **operating system** or the **kernel**.

Users communicate with the kernel through a program known as the **shell**. The shell is a command line interpreter; it translates commands entered by the user and converts them into a language that is understood by the kernel.

- Unix was originally developed in 1969 by a group of AT&T employees Ken Thompson, Dennis Ritchie, Douglas McIlroy, and Joe Ossanna at Bell Labs.
- There are various Unix variants available in the market. Solaris Unix, AIX, HP Unix and BSD are a few examples. Linux is also a flavor of Unix which is freely available.
- Several people can use a Unix computer at the same time; hence Unix is called a multiuser system.
- A user can also run multiple programs at the same time; hence Unix is a multitasking environment.

Unix Architecture

Here is a basic block diagram of a Unix system −



The main concept that unites all the versions of Unix is the following four basics −

- **Kernel** − The kernel is the heart of the operating system. It interacts with the hardware and most of the tasks like memory management, task scheduling and file management.
- **Shell** − The shell is the utility that processes your requests. When you type in a command at your terminal, the shell interprets the command and calls the program that you want. The shell uses standard syntax for all commands. C Shell, Bourne Shell and Korn Shell are the most famous shells which are available with most of the Unix variants.
- **Commands and Utilities** − There are various commands and utilities which you can make use of in your day to day activities. **cp**, **mv**, **cat** and **grep**, etc. are few examples of commands and utilities. There are over 250 standard commands plus numerous others provided through 3rd party software. All the commands come along with various options.

- **Files and Directories** − All the data of Unix is organized into files. All files are then organized into directories. These directories are further organized into a tree-like structure called the **filesystem**.

Important Features of Linux Operating System

Basic Features

Following are some of the important features of Linux Operating System.

- **Portable** – Portability means software can works on different types of hardware in same way. Linux kernel and application programs supports their installation on any kind of hardware platform.
- **Open Source** – Linux source code is freely available and it is community based development project. Multiple team works in collaboration to enhance the capability of Linux operating system and it is continuously evolving.
- **Multi-User** – Linux is a multiuser system means multiple users can access system resources like memory/ ram/ application programs at same time.
- **Multiprogramming** – Linux is a multiprogramming system means multiple applications can run at same time.
- **Hierarchical File System** – Linux provides a standard file structure in which system files/ user files are arranged.
- **Shell** – Linux provides a special interpreter program which can be used to execute commands of the operating system. It can be used to do various types of operations, call application programs etc.
- **Security** – Linux provides user security using authentication features like password protection/ controlled access to specific files/ encryption of data.

Linux is fast, free and easy to use, power laptops and servers around the world. Linux has many more features to amaze its users such as:

- **Live CD/USB:** Almost all Linux distributions have Live CD/USB feature by which user can run/try the OS even without installing it on the system.
- **Graphical user interface (X Window System):** People think that Linux is a command line OS, somewhere its true also but not necessarily, Linux have packages which can be installed to make the whole OS graphics based as Windows.

- **Support's** most national **or customized keyboards:** Linux is used worldwide and hence available in multiple languages, and supports most of their custom national keyboards.
- **Application Support:** Linux has its own software repository from where users can download and install thousands of applications just by issuing a command in Linux Terminal or Shell. Linux can also run Windows applications if needed.

Advantages of Linux

Linux is an open-source operating system like Windows and MacOS. It is not just limited to the operating system, but nowadays, it is also used as a platform to run desktops, servers, and embedded systems. It provides various distributions and variations as it is open source and has a modular design. The kernel *is a core part of* the Linux system.

Linux system is used to manage various services such as process scheduling, application scheduling, basic peripheral devices, file system, and more. Linux provides various advantages over other operating systems such as Windows and macOS. So, it is used in almost every field, from cars to home appliances and smartphones to servers (supercomputers).

We will see the advantages of Linux over other operating systems and will determine why it is better than other operating systems.

There are many features of the Linux operating system that demonstrate that it is better than other operating systems. However, in some prospective other operating systems can be more useful than Linux. Let's see the top 20 advantages of Linux OS.

Top 20 Advantages of Linux

Following are top 20 advantages of the Linux operating system:

1. **Open Source:** As it is open-source, its source code is easily available. Anyone having programming knowledge can customize the operating system. One can contribute, modify, distribute, and enhance the code for any purpose.

2. **Security:** The Linux security feature is the main reason that it is the most favorable option for developers. It is not completely safe, but it is less vulnerable than others. Each application needs to authorize by the admin user. The virus is not executed until the administrator provides

the access password. Linux systems do not require any antivirus program.

3. **Free:** Certainly, the biggest advantage of the Linux system is that it is free to use. We can easily download it, and there is no need to buy the license for it. It is distributed under GNU GPL (General Public License). Comparatively, we have to pay a huge amount for the license of the other operating systems.

4. **Lightweight**: Linux is lightweight. The requirements for running Linux are much less than other operating systems. In Linux, the memory footprint and disk space are also lower. Generally, most of the Linux distributions required as little as 128MB of RAM around the same amount for disk space.

5. **Stability**: Linux is more stable than other operating systems. Linux does not require to reboot the system to maintain performance levels. It rarely hangs up or slow down. It has big up-times.

6. **Performance**: Linux system provides high performance over different networks. It is capable of handling a large number of users simultaneously.

7. **Flexibility**: Linux operating system is very flexible. It can be used for desktop applications, embedded systems, and server applications too. It also provides various restriction options for specific computers. We can install only necessary components for a system.

8. **Software Updates**: In Linux, the software updates are in user control. We can select the required updates. There a large number of system updates are available. These updates are much faster than other operating systems. So, the system updates can be installed easily without facing any issue.

9. **Distributions/ Distros**: There are many Linux distributions available in the market. It provides various options and flavors of Linux to the users. We can choose any distros according to our needs. Some popular distros are Ubuntu, Fedora, Debian, Linux Mint, Arch Linux, and many more. For the beginners, Ubuntu and Linux Mint would be useful and, Debian and Fedora would be good choices for proficient programmers.

10. **Live CD/USB**: Almost all Linux distributions have a Live CD/USB option. It allows us to try or run the Linux operating system without installing it.

11. **Graphical User Interface**: Linux is a command-line based OS but, it provides an interactive user interface like Windows.

12. **Suitable for programmers:** It supports almost all of the most used programming languages such as C/C++, Java, Python, Ruby, and more. Further, it offers a vast range of useful applications for development.

    The programmers prefer the Linux terminal over the Windows command line. The package manager on Linux system helps programmers to understand how things are done. Bash scripting is also a functional feature for the programmers. It also provides support for SSH, which helps in managing the servers quickly.

13. **Community Support**: Linux provides large community support. We can find support from various sources. There are many forums available on the web to assist users. Further, developers from the various open source communities are ready to help us.

14. **Privacy**: Linux always takes care of user privacy as it never takes much private data from the user. Comparatively, other operating systems ask for the user's private data.

15. **Networking**: Linux facilitates with powerful support for networking. The client-server systems can be easily set to a Linux system. It provides various command-line tools such as ssh, ip, mail, telnet, and more for connectivity with the other systems and servers. Tasks such as network backup are much faster than others.

16. **Compatibility**: Linux is compatible with a large number of file formats as it supports almost all file formats.

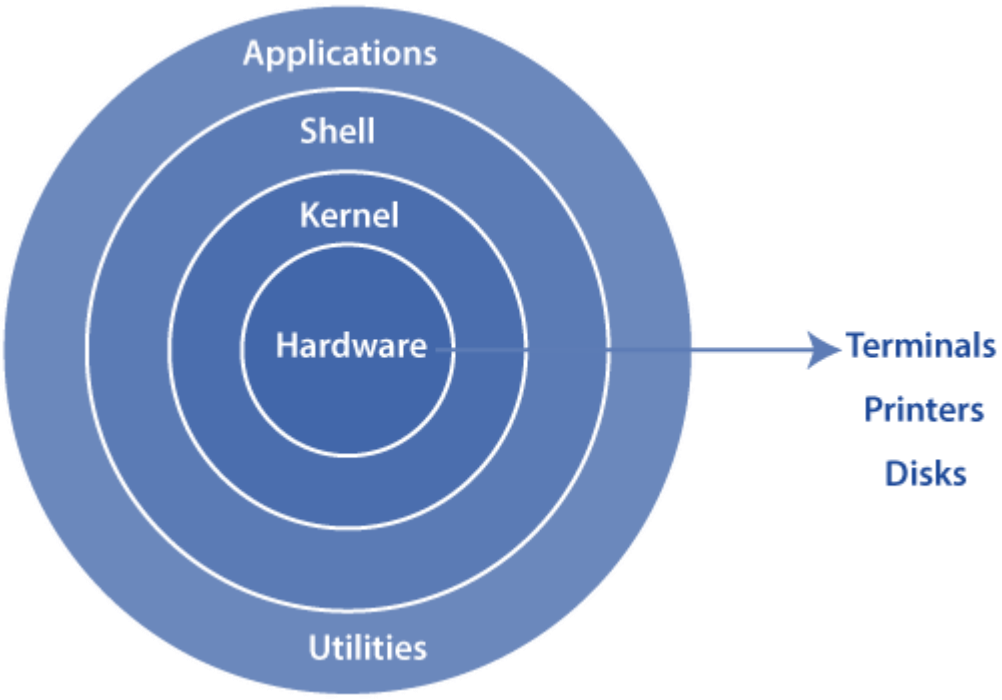
17. **Installation**: Linux installation process takes less time than other operating systems such as Windows. Further, its installation process is much easy as it requires less user input. It does not require much more system configuration even it can be easily installed on old machines having less configuration.

18. **Multiple Desktop Support**: Linux system provides multiple desktop environment support for its enhanced use. The desktop environment option can be selected during installation. We can select any desktop environment such as GNOME (GNU Network Object Model Environment) or KDE (K Desktop Environment) as both have their specific environment.

19. **Multitasking**: It is a multitasking operating system as it can run multiple tasks simultaneously without affecting the system speed.

20. **Heavily documented for beginners**: There are many command-line options that provide documentation on commands, libraries, standards such as manual pages and info pages. Also, there are plenty of documents available on the internet in different formats, such as Linux tutorials, Linux documentation project, Server fault, and more. To help the beginners, several communities are available such as Ask Ubuntu, Reddit, and Stack Overflow.

Architecture of Linux system



The Linux operating system's architecture mainly contains some of the components: **the Kernel, System Library, Hardware layer, System,** and **Shell utility**.

1. Kernel: - The kernel is one of the core section of an operating system. It is responsible for each of the major actions of the Linux OS. This operating system contains distinct types of modules and cooperates with underlying hardware directly. The kernel facilitates required abstraction for hiding details of low-level hardware or application programs to the system. There are some of the important kernel types which are mentioned below:

- Monolithic Kernel
- Micro kernels
- Exo kernels
- Hybrid kernels

2. System Libraries: - These libraries can be specified as some special functions. These are applied for implementing the operating system's functionality and don't need code access rights of the modules of kernel.

3. System Utility Programs: - It is responsible for doing specialized level and individual activities.

4. Hardware layer: - Linux operating system contains a hardware layer that consists of several peripheral devices like CPU, HDD, and RAM.

5. Shell: - It is an interface among the kernel and user. It can afford the services of kernel. It can take commands through the user and runs the functions of the kernel. The shell is available in distinct types of OSes. These operating systems are categorized into two different types, which are the **graphical shells** and **command-line shells**.

The graphical line shells facilitate the graphical user interface, while the command line shells facilitate the command line interface. Thus, both of these shells implement operations. However, the graphical user interface shells work slower as compared to the command-line interface shells.

There are a few types of these shells which are categorized as follows:

- Korn shell
- Bourne shell
- C shell
- POSIX shell

Linux File System

A Linux file system is a structured collection of files on a disk drive or a partition. A partition is a segment of memory and contains some specific data. In our machine, there can be various partitions of the memory. Generally, every partition contains a file system. The general-purpose computer system needs to store data systematically so that we can easily access the files in less time. It stores the data on hard disks (HDD) or some equivalent storage type. There may be below reasons for maintaining the file system:

- Primarily the computer saves data to the RAM storage; it may lose the data if it gets turned off. However, there is non-volatile RAM (Flash RAM and SSD) that is available to maintain the data after the power interruption.
- Data storage is preferred on hard drives as compared to standard RAM as RAM costs more than disk space. The hard disks costs are dropping gradually comparatively the RAM.

The Linux file system contains the following sections:

- The root directory (/)
- A specific data storage format (EXT3, EXT4, BTRFS, XFS and so on)
- A partition or logical volume having a particular file system.

Linux file system is generally a built-in layer of a Linux operating system used to handle the data management of the storage. It helps to arrange the file on the disk storage. It manages the file name, file size, creation date, and much more information about a file.
If we have an unsupported file format in our file system, we can download software to deal with it.

Linux File System Structure

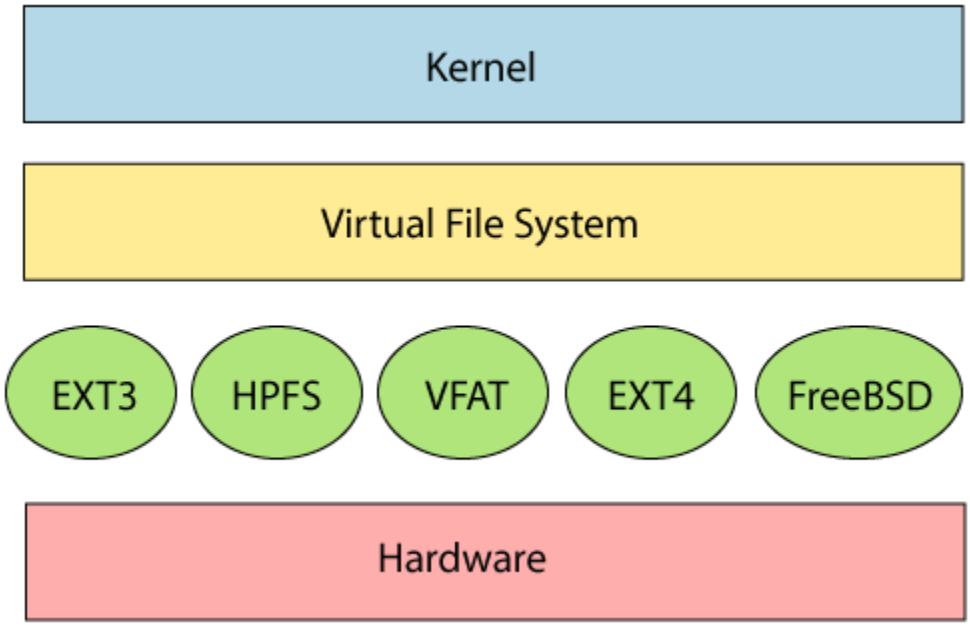
Linux file system has a hierarchal file structure as it contains a root directory and its subdirectories. All other directories can be accessed from the root directory. A partition usually has only one file system, but it may have more than one file system.
A file system is designed in a way so that it can manage and provide space for non-volatile storage data. All file systems required a namespace that is a naming and organizational methodology. The namespace defines the naming process, length of the file name, or a subset of characters that can be used for the file name. It also defines the logical structure of files on a memory segment, such as the use of directories for organizing the specific files. Once a namespace is described, a Metadata description must be defined for that particular file.

The data structure needs to support a hierarchical directory structure; this structure is used to describe the available and used disk space for a particular block. It also has the

other details about the files such as file size, date & time of creation, update, and last modified.

Also, it stores advanced information about the section of the disk, such as partitions and volumes.

The advanced data and the structures that it represents contain the information about the file system stored on the drive; it is distinct and independent of the file system metadata. Linux file system contains two-part file system software implementation architecture. Consider the below image:



The file system requires an API (Application programming interface) to access the function calls to interact with file system components like files and directories. API facilitates tasks such as creating, deleting, and copying the files. It facilitates an algorithm that defines the arrangement of files on a file system.

The first two parts of the given file system together called a **Linux virtual file system**. It provides a single set of commands for the kernel and developers to access the file system. This virtual file system requires the specific system driver to give an interface to the file system.

Hardware Requirements for Linux
Most people will want to install a desktop system such as **Ubuntu**, **Kubuntu**, or **Xubuntu**. A desktop system is typically used for personal computing tasks and has a graphical user interface (GUI), while a server system typically has a command-line interface (CLI).

Recommended Minimum System Requirements
The **Recommended** Minimum System Requirements, here, should allow even someone fairly new to installing Ubuntu or Gnu&Linux to easily install a usable system with enough room to be comfortable. A good "rule of thumb" is that machines that could run XP, Vista, Windows 7 or x86 OS X will almost always be a lot faster with Ubuntu even if they are lower-spec than described below. Simply try Ubuntu CD as a LiveCD first to check the hardware works.

Ubuntu Desktop Edition
1. 2 GHz dual core processor
2. 4 GiB RAM (system memory)
3. 25 GB of hard-drive space (or USB stick, memory card or external drive but see LiveCD for an alternative approach)
4. VGA capable of 1024x768 screen resolution
5. Either a CD/DVD drive or a USB port for the installer media
6. Internet access is helpful

Screen resolution will be set at the highest your graphics card can handle but when you boot-up you should be given a "Low graphics mode" option which allows you to set it to something better for your monitor.

On one hand, hardware produced in the last few years or with an efficient architecture or machines built for a specific purpose can often work well with less. For example, a netbook with an 8 GB SSD will work well although there wont be much room for saving stuff directly onto the drive so cloud storage services could help a lot. A machine with a crumbling, 15 year-old, slow, 8 GB, IDE hard-drive probably won't work and doesn't really compare with the netbook anyway. It might be worth trying Ubuntu but really start looking at other distros. On the other hand, some GNU/Linux distributions may require more powerful hardware as minimum system requirements, like the Ubuntu GNOME case.

All 64-bit (x86-64) CPUs should be fast enough to run Ubuntu and can run the 32-bit (x86) version as well. For an optimized installation (and especially for those wishing to run more than ~3 GiB of RAM) however, a 64-bit installation CD is available. The 32-bit version tends to be easier to use and runs into less problems. 32-bit ISO images are no longer being produced (as of 17.10).

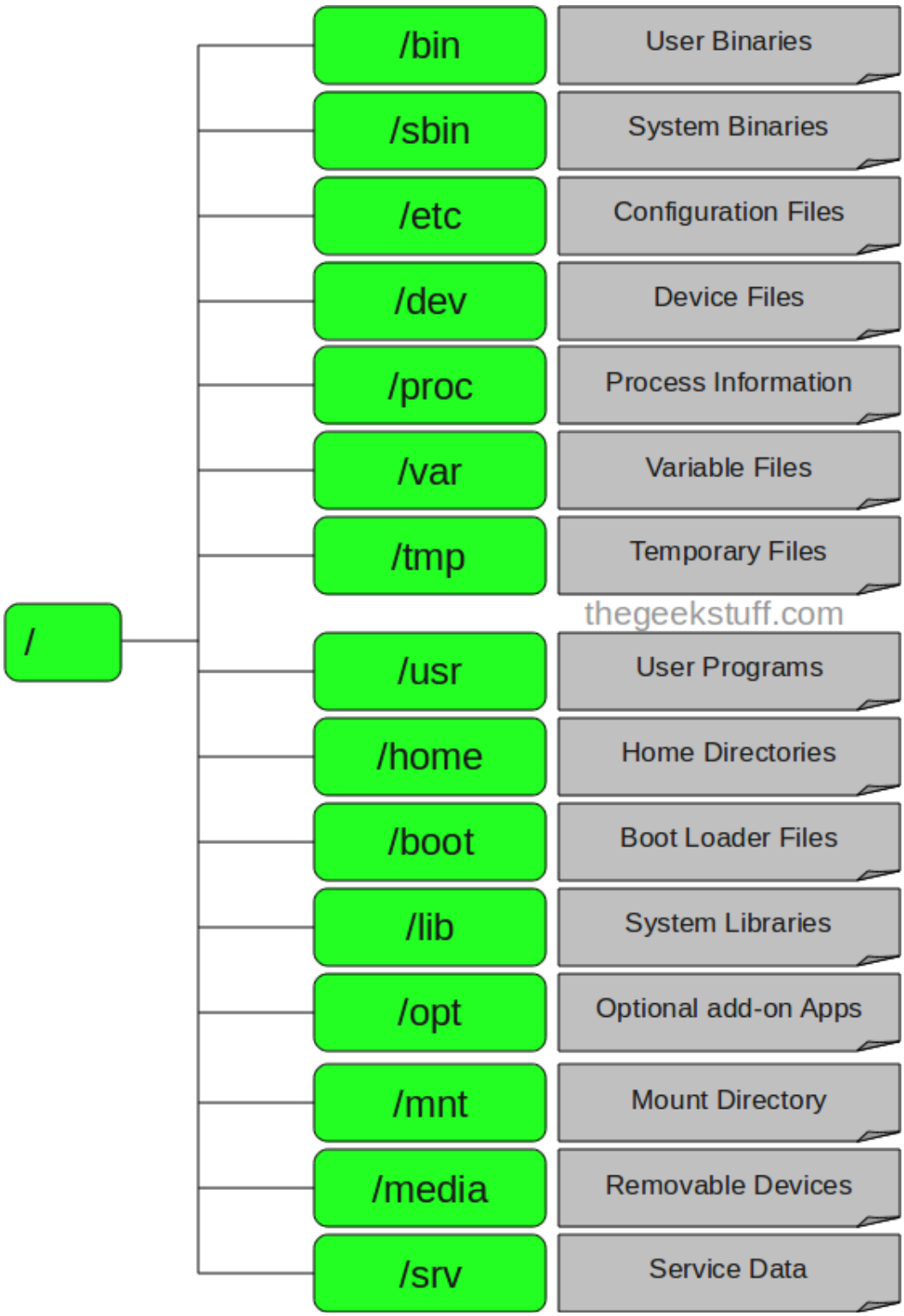
Ubuntu Desktop 11.04 through 17.04 uses Unity as the default GUI while the previous releases used GNOME Panel by default. From 17.10 onwards the desktop uses GNOME Shell. In order to run these environments the system needs a more capable graphics adapter – see more here or below:

1. 4096 MiB RAM (system memory) for physical installs.
2. 2048 MiB RAM (system memory) for virtualised installs.
3. 3D Acceleration Capable Videocard with at least 256 MB

Machines that are 10 or more years old (originally preloaded with "Windows ME" or "Windows 2000") that don't meet these guideline will probably require some work to revive (the RAM usually needs to be upgraded to the level described above). You could try a lighter-weight distro or a minimal install of Ubuntu.

Linux Standard Directories

Let us review the Linux filesystem structures and understand the meaning of individual high-level directories.



1. / – Root

- Every single file and directory starts from the root directory.
- Only root user has write privilege under this directory.
- Please note that /root is root user's home directory, which is not same as /.

2. /bin – User Binaries

- Contains binary executables.
- Common Linux commands you need to use in single-user modes are located under this directory.
- Commands used by all the users of the system are located here.
- For example: ps, ls, ping, grep, cp.

3. /sbin – System Binaries

- Just like /bin, /sbin also contains binary executables.
- But, the Linux commands located under this directory are used typically by system administrator or super user, for system maintenance purpose.
- For example: iptables, reboot, fdisk, ifconfig, swapon

4. /etc – Configuration Files

- Contains configuration files required by all programs.
- This also contains startup and shutdown shell scripts used to start/stop individual programs.
- For example: /etc/resolv.conf, /etc/logrotate.conf

5. /dev – Device Files

- Contains device files.
- These include terminal devices, usb, or any device attached to the system.
- For example: /dev/tty1, /dev/usbmon0

6. /proc – Process Information

- Contains information about system process.
- This is a pseudo filesystem contains information about running process. For example: /proc/{pid} directory contains information about the process with that particular pid.
- This is a virtual filesystem with text information about system resources. For example: /proc/uptime

7. /var – Variable Files

- var stands for variable files.
- Content of the files that are expected to grow can be found under this directory.

- This includes — system log files (/var/log); packages and database files (/var/lib); emails (/var/mail); print queues (/var/spool); lock files (/var/lock); temp files needed across reboots (/var/tmp);

8. /tmp – Temporary Files

- Directory that contains temporary files created by system and users.
- Files under this directory are deleted when system is rebooted.

9. /usr – User Programs

- Contains binaries, libraries, documentation, and source-code for second level programs.
- /usr/bin contains binary files for user programs. If you can't find a user binary under /bin, look under /usr/bin. For example: at, awk, cc, less, scp
- /usr/sbin contains binary files for system administrators. If you can't find a system binary under /sbin, look under /usr/sbin. For example: atd, cron, sshd, useradd, userdel
- /usr/lib contains libraries for /usr/bin and /usr/sbin
- /usr/local contains users programs that you install from source. For example, when you install apache from source, it goes under /usr/local/apache2

10. /home – Home Directories

- Home directories for all users to store their personal files.
- For example: /home/john, /home/nikita

11. /boot – Boot Loader Files

- Contains boot loader related files.
- Kernel initrd, vmlinux, grub files are located under /boot
- For example: initrd.img-2.6.32-24-generic, vmlinuz-2.6.32-24-generic

12. /lib – System Libraries

- Contains library files that supports the binaries located under /bin and /sbin
- Library filenames are either ld* or lib*.so.*
- For example: ld-2.11.1.so, libncurses.so.5.7

13. /opt – Optional add-on Applications

- opt stands for optional.
- Contains add-on applications from individual vendors.
- add-on applications should be installed under either /opt/ or /opt/ sub-directory.

14. /mnt – Mount Directory

- Temporary mount directory where sysadmins can mount filesystems.

15. /media – Removable Media Devices

- Temporary mount directory for removable devices.
- For examples, /media/cdrom for CD-ROM; /media/floppy for floppy drives; /media/cdrecorder for CD writer

16. /srv – Service Data

- srv stands for service.
- Contains server specific services related data.
- For example, /srv/cvs contains CVS related data.

Commands for Files and Directories

The Linux command is a utility of the Linux operating system. All basic and advanced tasks can be done by executing commands. The commands are executed on the **Linux terminal**. The terminal is a command-line interface to interact with the system, which is similar to the command prompt in the Windows OS. *Commands in Linux are **case-sensitive***.

Linux provides a powerful command-line interface compared to other operating systems such as Windows and MacOS. We can do basic work and advanced work through its terminal. We can do some basic tasks such as creating a file, deleting a file, moving a file, and more. In addition, we can also perform advanced tasks such as administrative tasks (including package installation, user management), networking tasks (ssh connection), security tasks, and many more.

Linux terminal is a user-friendly terminal as it provides various support options. To open the Linux terminal, press "**CTRL + ALT + T**" keys together, and execute a command by pressing the '**ENTER**' key.

In this topic, we will discuss the top 50 most frequently used Linux commands with their examples. These commands are very useful for a beginner and professional both. We have divided these commands into following sections so that you can easily identify their usage:

- Linux Directory Commands
- Linux File Commands
- Linux File Content Commands
- Linux User Commands
- Linux Filter Commands
- Linux Utility Commands
- Linux Networking Command

Linux Top 50 Commands

The following are the top 50 Linux commands:

Linux Directory Commands

1. pwd Command

The pwd command is used to display the location of the current working directory.

Syntax:

pwd

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ pwd
/home/javatpoint
```

2. mkdir Command

The mkdir command is used to create a new directory under any directory.

Syntax:

mkdir **<directory** name**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ mkdir new_directory
javatpoint@javatpoint-Inspiron-3542:~$
```

3. rmdir Command

The rmdir command is used to delete a directory.

Syntax:

rmdir **<directory** name**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ rmdir new_directory
javatpoint@javatpoint-Inspiron-3542:~$
```

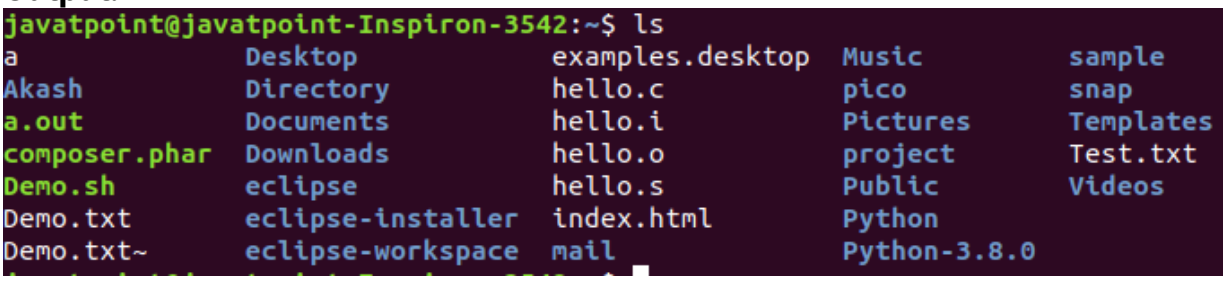
4. ls Command

The ls command is used to display a list of content of a directory.

Syntax:

ls

Output:



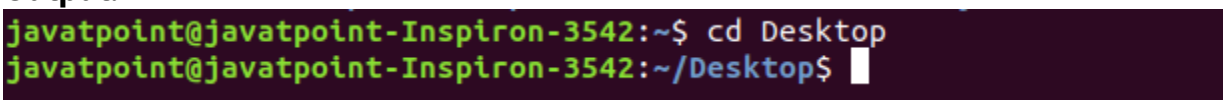
5. cd Command

The cd command is used to change the current directory.

Syntax:

cd **<directory** name**>**

Output:



Linux File commands

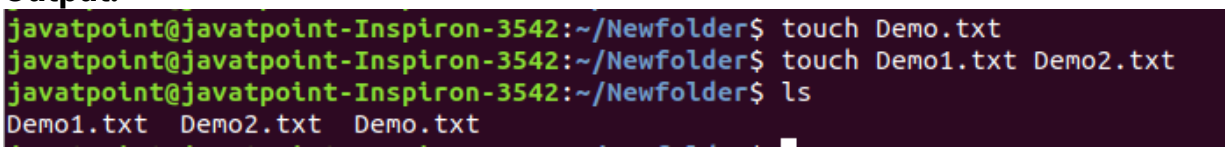
6. touch Command

The touch command is used to create empty files. We can create multiple empty files by executing it once.

Syntax:

touch **<file** name**>**
touch **<file1> <file2>** ....

Output:



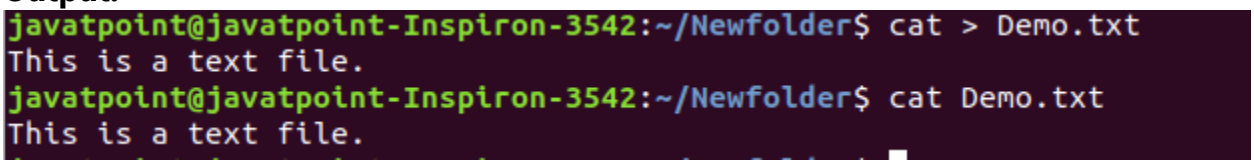
7. cat Command

The cat command is a multi-purpose utility in the Linux system. It can be used to create a file, display content of the file, copy the content of one file to another file, and more.

Syntax:

cat [OPTION]... [FILE]..
To create a file, execute it as follows:
cat **> <file** name**>**
// Enter file content
Press "**CTRL+ D**" keys to save the file. To display the content of the file, execute it as follows:
cat **<file** name**>**

Output:



8. rm Command

The rm command is used to remove a file.

Syntax:

rm <file name>

Output:

```
javatpoint@javatpoint-Inspiron-3542:~/Newfolder$ rm Demo.txt
javatpoint@javatpoint-Inspiron-3542:~/Newfolder$ rm Demo1.txt Demo2.txt
```

9. cp Command

The cp command is used to copy a file or directory.

Syntax:

To copy in the same directory:

cp **<existing** file name**>** **<new** file name**>**

To copy in a different directory:

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ cp demo.txt demo1.txt
javatpoint@javatpoint-Inspiron-3542:~$ cp demo.txt Documents
```

10. mv Command

The mv command is used to move a file or a directory form one location to another location.

Syntax:

mv **<file** name**>** **<directory** path**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ mv demo.txt Directory
```

11. rename Command

The rename command is used to rename files. It is useful for renaming a large group of files.

Syntax:

rename 's/old-name/new-name/' files

For example, to convert all the text files into pdf files, execute the below command:

rename 's/\.txt$/\.pdf/' *.txt

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ rename 's/\.txt$/\.pdf/' *.txt
javatpoint@javatpoint-Inspiron-3542:~$ ls
a                Desktop             examples.desktop  Music       Python-3.8.0
Akash            Directory           hello.c           Newfolder   sample
a.out            Documents           hello.i           pico        snap
composer.phar    Downloads           hello.o           Pictures    Templates
demo1.pdf        eclipse             hello.s           project     Test.pdf
Demo.sh          eclipse-installer   index.html        Public      Videos
Demo.txt~        eclipse-workspace   mail              Python
```

Linux File Content Commands

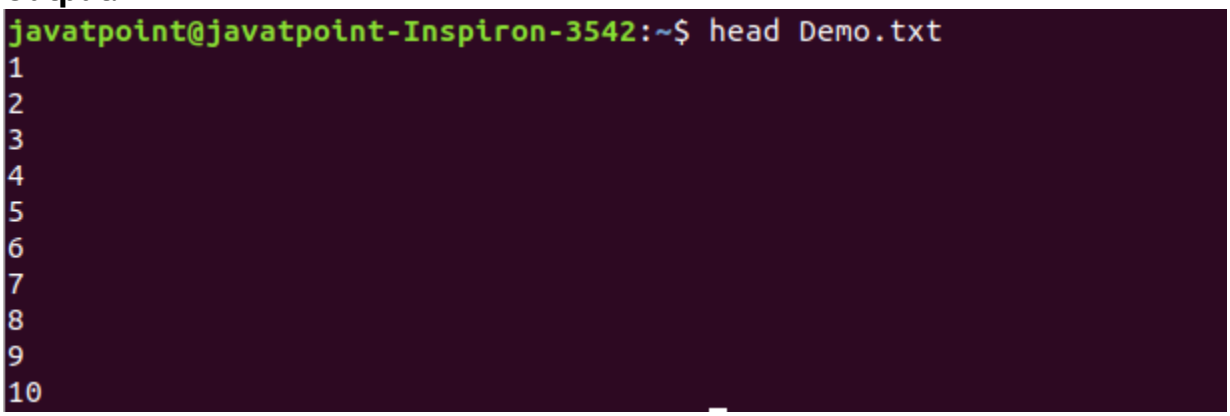
12. head Command

The head command is used to display the content of a file. It displays the first 10 lines of a file.

Syntax:

head **<file** name**>**

Output:



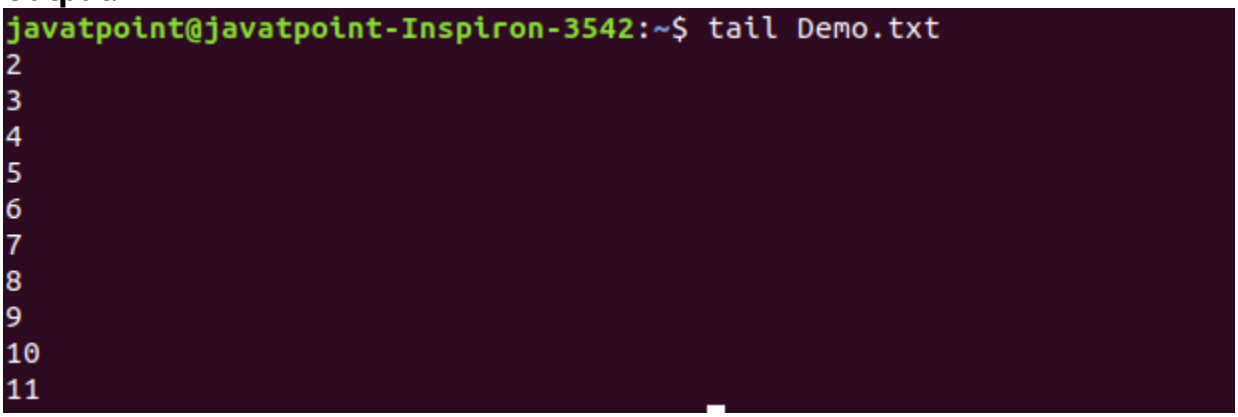
13. tail Command

The tail command is similar to the head command. The difference between both commands is that it displays the last ten lines of the file content. It is useful for reading the error message.

Syntax:

tail **<file** name**>**

Output:



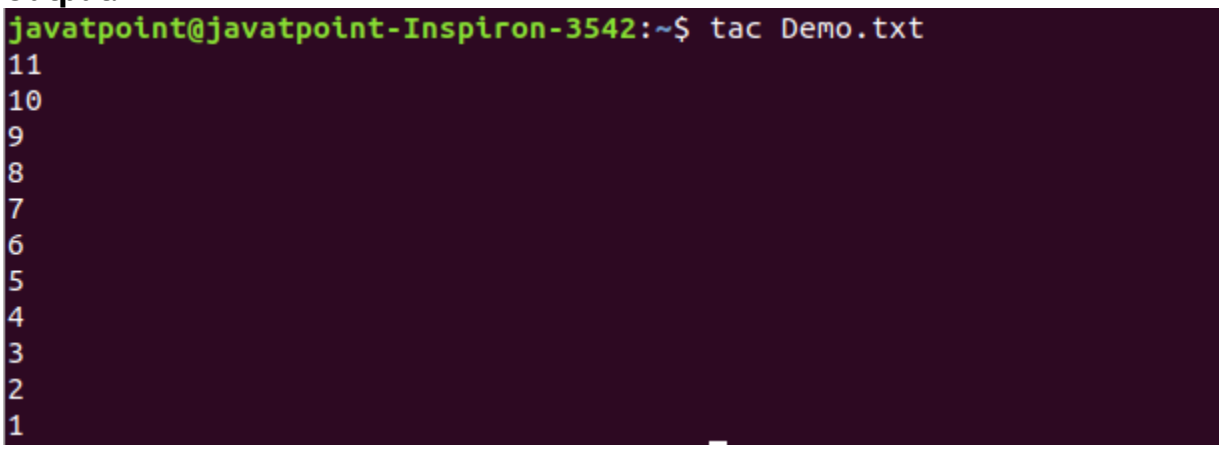
14. tac Command

The tac command is the reverse of cat command, as its name specified. It displays the file content in reverse order (from the last line).

Syntax:

tac **<file** name**>**

Output:



15. more command

The more command is quite similar to the cat command, as it is used to display the file content in the same way that the cat command does. The only difference between both commands is that, in case of larger files, the more command displays screenful output at a time.

In more command, the following keys are used to scroll the page:

ENTER key: To scroll down page by line.

Space bar: To move to the next page.

b key: To move to the previous page.

/ key: To search the string.

Syntax:

more **<file** name**>**

Output:

```
;;; gyp.el - font-lock-mode support for gyp files.

;; Copyright (c) 2012 Google Inc. All rights reserved.
;; Use of this source code is governed by a BSD-style license that can be
;; found in the LICENSE file.

;; Put this somewhere in your load-path and
;; (require 'gyp)

(require 'python)
(require 'cl)

(when (string-match "python-mode.el" (symbol-file 'python-mode 'defun))
  (error (concat "python-mode must be loaded from python.el (bundled with "
                 "recent emacsen), not from the older and less maintained "
                 "python-mode.el")))

(defadvice python-indent-calculate-levels (after gyp-outdent-closing-parens
                                                 activate)
  "De-indent closing parens, braces, and brackets in gyp-mode."
  (when (and (eq major-mode 'gyp-mode)
             (string-match "^ *[])}][],)}]* *$"
                           (buffer-substring-no-properties
--More--(7%)
```

16. less Command

The less command is similar to the more command. It also includes some extra features such as 'adjustment in width and height of the terminal.' Comparatively, the more command cuts the output in the width of the terminal.

Syntax:

less **<file** name**>**

Output:

```
;;; gyp.el - font-lock-mode support for gyp files.

;; Copyright (c) 2012 Google Inc. All rights reserved.
;; Use of this source code is governed by a BSD-style license that can be
;; found in the LICENSE file.

;; Put this somewhere in your load-path and
;; (require 'gyp)

(require 'python)
(require 'cl)

(when (string-match "python-mode.el" (symbol-file 'python-mode 'defun))
  (error (concat "python-mode must be loaded from python.el (bundled with "
                 "recent emacsen), not from the older and less maintained "
                 "python-mode.el")))

(defadvice python-indent-calculate-levels (after gyp-outdent-closing-parens
                                                 activate)
```

Linux User Commands

17. su Command

The su command provides administrative access to another user. In other words, it allows access of the Linux shell to another user.

Syntax:

su **<user** name**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ su javatpoint
Password:
javatpoint@javatpoint-Inspiron-3542:~$
```

18. id Command

The id command is used to display the user ID (UID) and group ID (GID).

Syntax:

id

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ id
uid=1000(javatpoint) gid=1000(javatpoint) groups=1000(javatpoint),4(adm),24(cdro
m),27(sudo),30(dip),46(plugdev),116(lpadmin),126(sambashare)
javatpoint@javatpoint-Inspiron-3542:~$
```

19. useradd Command

The useradd command is used to add or remove a user on a Linux server.

Syntax:

useradd username

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ sudo useradd JTP
[sudo] password for javatpoint:
javatpoint@javatpoint-Inspiron-3542:~$
```

20. passwd Command

The passwd command is used to create and change the password for a user.

Syntax:
passwd **<username>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ sudo passwd JTP
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

21. groupadd Command

The groupadd command is used to create a user group.

Syntax:
groupadd **<group** name**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ sudo groupadd Developer
javatpoint@javatpoint-Inspiron-3542:~$
```

Linux Filter Commands

22. cat Command

The cat command is also used as a filter. To filter a file, it is used inside pipes.

Syntax:

cat **<fileName>** | cat or tac | cat or tac |. . .

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ cat Demo.txt | tac | cat | cat | tac
1
2
3
4
5
6
7
8
9
10
11
```

23. cut Command

The cut command is used to select a specific column of a file. The '-d' option is used as a delimiter, and it can be a space (' '), a slash (/), a hyphen (-), or anything else. And, the '-f' option is used to specify a column number.

Syntax:

cut -d(delimiter) -f(columnNumber) **<fileName>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ cat >marks.txt
alex-50
alen-70
jon-75
carry-85
celena-90
justin-80
javatpoint@javatpoint-Inspiron-3542:~$ cut -d- -f2 marks.txt
50
70
75
85
90
80
javatpoint@javatpoint-Inspiron-3542:~$
```

24. grep Command

The grep is the most powerful and used filter in a Linux system. The 'grep' stands for "**global regular expression print**." It is useful for searching the content from a file. Generally, it is used with the pipe.

Syntax:

command | grep **<searchWord>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ cat marks.txt | grep 9
celena-90
```

25. comm Command

The 'comm' command is used to compare two files or streams. By default, it displays three columns, first displays non-matching items of the first file, second indicates the non-matching item of the second file, and the third column displays the matching items of both files.

Syntax:

comm **<file1> <file2>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ comm Demo.txt Demo1.txt
                1
2
                3
comm: file 2 is not in sorted order
        11
                4
                5
        22
        33
6
7
8
9
comm: file 1 is not in sorted order
10
11
```

26. sed command

The sed command is also known as **stream editor**. It is used to edit files using a regular expression. It does not permanently edit files; instead, the edited content remains only on display. It does not affect the actual file.

Syntax:
command | sed 's/**<oldWord>**/**<newWord>**/'

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ echo class7 | sed 's/class/jtp/'
jtp7
javatpoint@javatpoint-Inspiron-3542:~$ echo class7 | sed 's/7/10/'
class10
```

27. tee command

The tee command is quite similar to the cat command. The only difference between both filters is that it puts standard input on standard output and also write them into a file.

Syntax:
cat **<fileName>** | tee **<newFile>** | cat or tac |.....

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ cat marks.txt | tee new.txt | cat
alex-50
alen-70
jon-75
carry-85
celena-90
justin-80
javatpoint@javatpoint-Inspiron-3542:~$ cat new.txt
alex-50
alen-70
jon-75
carry-85
celena-90
justin-80
```

28. tr Command

The tr command is used to translate the file content like from lower case to upper case.

Syntax:
command | tr <'old'> <'new'>

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ cat marks.txt | tr 'prcu' 'PRCU'
alex-50
alen-70
jon-75
CaRRy-85
Celena-90
jUstin-80
```

29. uniq Command

The uniq command is used to form a sorted list in which every word will occur only once.

Syntax:

command **<fileName>** | uniq

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ sort marks.txt |uniq
alen-70
alex-50
carry-85
celena-90
jon-75
justin-80
```

30. wc Command

The wc command is used to count the lines, words, and characters in a file.

Syntax:

wc **<file** name**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ wc marks.txt
 6  6 52 marks.txt
```

31. od Command

The od command is used to display the content of a file in different s, such as hexadecimal, octal, and ASCII characters.

Syntax:

od -b **<fileName>** // Octal format
od -t x1 **<fileName>** // Hexa decimal format
od -c **<fileName>** // ASCII character format

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ od -b marks.txt
0000000 141 154 145 170 055 065 060 012 141 154 145 156 055 067 060 012
0000020 152 157 156 055 067 065 012 143 141 162 162 171 055 070 065 012
0000040 143 145 154 145 156 141 055 071 060 012 152 165 163 164 151 156
0000060 055 070 060 012
0000064
javatpoint@javatpoint-Inspiron-3542:~$ od -t x1 marks.txt
0000000 61 6c 65 78 2d 35 30 0a 61 6c 65 6e 2d 37 30 0a
0000020 6a 6f 6e 2d 37 35 0a 63 61 72 72 79 2d 38 35 0a
0000040 63 65 6c 65 6e 61 2d 39 30 0a 6a 75 73 74 69 6e
0000060 2d 38 30 0a
0000064
javatpoint@javatpoint-Inspiron-3542:~$ od -c marks.txt
0000000   a   l   e   x   -   5   0  \n   a   l   e   n   -   7   0  \n
0000020   j   o   n   -   7   5  \n   c   a   r   r   y   -   8   5  \n
0000040   c   e   l   e   n   a   -   9   0  \n   j   u   s   t   i   n
0000060   -   8   0  \n
0000064
```

32. sort Command

The sort command is used to sort files in alphabetical order.

Syntax:

sort **<file** name**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ sort marks.txt
alen-70
alex-50
carry-85
celena-90
jon-75
justin-80
```

33. gzip Command

The gzip command is used to truncate the file size. It is a compressing tool. It replaces the original file by the compressed file having '.gz' extension.

Syntax:

gzip **<file1> <file2> <file3>**...

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ gzip Demo.txt Demo1.txt
javatpoint@javatpoint-Inspiron-3542:~$ ls
a               Demo.txt.gz         examples.desktop  Music       Python-3.8.0
Akash           Desktop             hello.c           Newfolder   sample
a.out           Directory           hello.i           new.txt     snap
composer.phar   Documents           hello.o           pico        Templates
demo1.pdf       Downloads           hello.s           Pictures    Test.pdf
Demo1.txt.gz    eclipse             index.html        project     Videos
Demo.sh         eclipse-installer   mail              Public
Demo.txt~       eclipse-workspace   marks.txt         Python
```

34. gunzip Command

The gunzip command is used to decompress a file. It is a reverse operation of gzip command.

Syntax:

gunzip **<file1> <file2> <file3>**. .

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ gunzip Demo.txt Demo1.txt
javatpoint@javatpoint-Inspiron-3542:~$ ls
a               Demo.txt~           examples.desktop  Music       Python-3.8.0
Akash           Desktop             hello.c           Newfolder   sample
a.out           Directory           hello.i           new.txt     snap
composer.phar   Documents           hello.o           pico        Templates
demo1.pdf       Downloads           hello.s           Pictures    Test.pdf
Demo1.txt       eclipse             index.html        project     Videos
Demo.sh         eclipse-installer   mail              Public
Demo.txt        eclipse-workspace   marks.txt         Python
```

Linux Utility Commands

35. find Command

The find command is used to find a particular file within a directory. It also supports various options to find a file such as byname, by type, by date, and more.

The following symbols are used after the find command:

(.) : For current directory name

(/) : For root

Syntax:

find . -name "*.pdf"

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ find . -name "*.pdf"
./Test.pdf
./Python-3.8.0/Doc/library/turtle-star.pdf
./Akash/Joomla/Origional Copy/Brochure-Joomla-2019.pdf
./Akash/Joomla/Origional Copy/Joomla-Guide-Final.pdf
./.local/share/Trash/files/2400966-250544e72f817db3bcef-1587140240830.pdf
./.local/share/Trash/files/2400966-3ad982eaa58c5d43fb53-1585763620407.pdf
find: ‘./.anydesk/incoming’: Permission denied
./Downloads/ConfirmationPage_20030070774.pdf
./demo1.pdf
find: ‘./.dbus’: Permission denied
find: ‘./.cache/dconf’: Permission denied
./Directory/demo.pdf
./Directory/demo2.pdf
./Directory/demo1.pdf
```

36. locate Command

The locate command is used to search a file by file name. It is quite similar to find command; the difference is that it is a background process. It searches the file in the database, whereas the find command searches in the file system. It is faster than the find command. To find the file with the locates command, keep your database updated.

Syntax:

locate **<file** name**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ locate sysctl.conf
/etc/sysctl.conf
/etc/sysctl.d/99-sysctl.conf
/etc/ufw/sysctl.conf
/snap/core/8935/etc/sysctl.conf
/snap/core/8935/etc/sysctl.d/99-sysctl.conf
/snap/core/9066/etc/sysctl.conf
/snap/core/9066/etc/sysctl.d/99-sysctl.conf
/snap/core18/1705/etc/sysctl.d/99-sysctl.conf
/snap/core18/1754/etc/sysctl.d/99-sysctl.conf
/usr/share/doc/procps/examples/sysctl.conf
/usr/share/man/man5/sysctl.conf.5.gz
```

37. date Command

The date command is used to display date, time, time zone, and more.

Syntax:

date

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ date
Fri May 22 21:51:05 IST 2020
```

38. cal Command

The cal command is used to display the current month's calendar with the current date highlighted.

Syntax:

cal<

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ cal
      May 2020
Su Mo Tu We Th Fr Sa
                1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31
```

39. sleep Command

The sleep command is used to hold the terminal by the specified amount of time. By default, it takes time in seconds.

Syntax:

sleep **<time>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ sleep 4
```

40. time Command

The time command is used to display the time to execute a command.

Syntax:
time

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ time

real    0m0.000s
user    0m0.000s
sys     0m0.000s
```

41. zcat Command

The zcat command is used to display the compressed files.

Syntax:
zcat **<file** name**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ ls
a               Demo.txt.gz         examples.desktop  Music       Python-3.8.0
Akash           Desktop             hello.c           Newfolder   sample
a.out           Directory           hello.i           new.txt     snap
composer.phar   Documents           hello.o           pico        Templates
demo1.pdf       Downloads           hello.s           Pictures    Test.pdf
Demo1.txt       eclipse             index.html        project     Videos
Demo.sh         eclipse-installer   mail              Public
Demo.txt~       eclipse-workspace   marks.txt         Python
javatpoint@javatpoint-Inspiron-3542:~$ zcat Demo.txt
1
2
3
4
5
6
```

42. df Command

The df command is used to display the disk space used in the file system. It displays the output as in the number of used blocks, available blocks, and the mounted directory.

Syntax:
df

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ df
Filesystem     1K-blocks     Used Available Use% Mounted on
udev             1931652        0   1931652   0% /dev
tmpfs             393260     1756    391504   1% /run
/dev/sda1      479668904 26471148 428762148   6% /
tmpfs            1966284   243536   1722748  13% /dev/shm
tmpfs               5120        4      5116   1% /run/lock
tmpfs            1966284        0   1966284   0% /sys/fs/cgroup
/dev/loop1        231936   231936         0 100% /snap/wine-platform-runtime/136
/dev/loop2        144128   144128         0 100% /snap/gnome-3-26-1604/98
/dev/loop4           384      384         0 100% /snap/gnome-characters/539
/dev/loop6        220160   220160         0 100% /snap/wine-platform-5-stable/4
/dev/loop5        164096   164096         0 100% /snap/gnome-3-28-1804/116
```

43. mount Command

The mount command is used to connect an external device file system to the system's file system.

Syntax:

mount -t type **<device> <directory>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,nosuid,relatime,size=1931652k,nr_inodes=482913,mo
de=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmod
e=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=393260k,mode=755)
/dev/sda1 on / type ext4 (rw,relatime,errors=remount-ro)
securityfs on /sys/kernel/security type securityfs (rw,nosuid,nodev,noexec,relat
ime)
tmpfs on /dev/shm type tmpfs (rw,nosuid,nodev)
```

44. exit Command

Linux exit command is used to exit from the current shell. It takes a parameter as a number and exits the shell with a return of status number.

Syntax:

exit

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ exit
```

After pressing the ENTER key, it will exit the terminal.

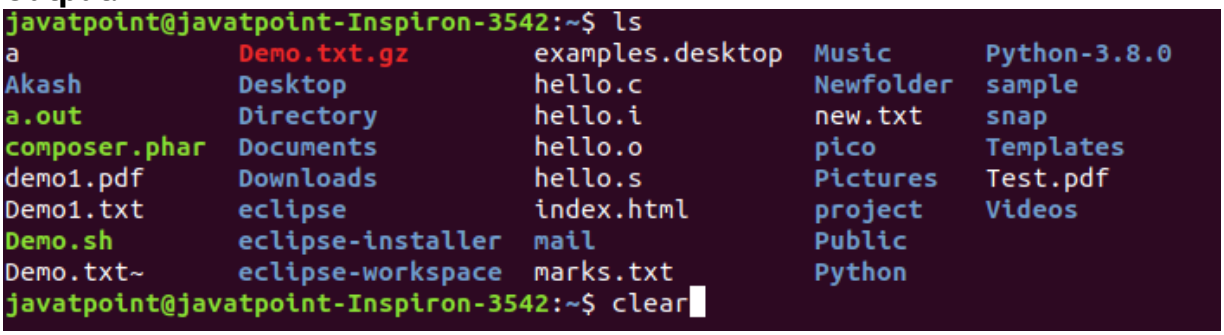
45. clear Command

Linux **clear** command is used to clear the terminal screen.

Syntax:

clear

Output:



After pressing the ENTER key, it will clear the terminal screen.

Linux Networking Commands

46. ip Command

Linux ip command is an updated version of the ipconfig command. It is used to assign an IP address, initialize an interface, disable an interface.

Syntax:

ip a or ip addr

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp7s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOW
N group default qlen 1000
    link/ether 74:e6:e2:02:93:b8 brd ff:ff:ff:ff:ff:ff
3: wlp6s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP gro
up default qlen 1000
    link/ether 00:71:cc:00:e2:89 brd ff:ff:ff:ff:ff:ff
    inet 192.168.43.240/24 brd 192.168.43.255 scope global dynamic noprefixroute
 wlp6s0
       valid_lft 2296sec preferred_lft 2296sec
    inet6 fe80::8c59:e84e:1670:27cc/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

47. ssh Command

Linux ssh command is used to create a remote connection through the ssh protocol.

Syntax:

ssh user_name@host(IP/Domain_name)**</p>**

48. mail Command

The mail command is used to send emails from the command line.

Syntax:

mail -s "Subject" **<recipient** address**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ mail -s "Hello World" Himanshudubey481@gmail.com
Cc:
Hello There
Hope you are doing well.
```

49. ping Command

The ping command is used to check the connectivity between two nodes, that is whether the server is connected. It is a short form of "Packet Internet Groper."

Syntax:

ping **<destination>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ ping javatpoint.com
PING javatpoint.com (194.169.80.121) 56(84) bytes of data.
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=1 ttl=48 time=3889 m
s
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=2 ttl=48 time=3043 m
s
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=3 ttl=48 time=2136 m
s
64 bytes from www.javatpoint.com (194.169.80.121): icmp_seq=4 ttl=48 time=1122 m
s
```

50. host Command

The host command is used to display the IP address for a given domain name and vice versa. It performs the DNS lookups for the DNS Query.

Syntax:

host **<domain** name**>** or **<ip** address**>**

Output:

```
javatpoint@javatpoint-Inspiron-3542:~$ host javatpoint.com
javatpoint.com has address 194.169.80.121
```

cd command

To navigate through the Linux files and directories, use the **cd** command. It requires either the full path or the name of the directory, depending on the current working directory that you're in.

Let's say you're in **/home/username/Documents** and you want to go to **Photos**, a subdirectory of **Documents**. To do so, simply type the following command:

cd Photos.

Another scenario is if you want to switch to a completely new directory, for example,**/home/username/Movies**. In this case, you have to type **cd** followed by the directory's absolute path: **cd /home/username/Movies**.

There are some shortcuts to help you navigate quickly:

- **cd ..** (with two dots) to move one directory up
- **cd** to go straight to the home folder
- **cd-** (with a hyphen) to move to your previous directory

On a side note, Linux's shell is case sensitive. So, you have to type the name's directory exactly as it is.

ls command

The **ls** command is used to view the contents of a directory.
By default, this command will display the contents of your current working directory.

If you want to see the content of other directories, type **ls** and then the directory's path. For example, enter **ls /home/username/Documents** to view the content of **Documents**.

There are variations you can use with the **ls** command:

- **ls -R** will list all the files in the sub-directories as well
- **ls -a** will show the hidden files
- **ls -al** will list the files and directories with detailed information like the permissions, size, owner, etc.

cp command
Use the **cp** command to copy files from the current directory to a different directory.

For instance, the command **cp scenery.jpg /home/username/Pictures** would create a copy of **scenery.jpg** (from your current directory) into the **Pictures** directory.

mv command
The primary use of the **mv** command is to move files, although it can also be used to rename files.

The arguments in mv are similar to the cp command. You need to type **mv**, the file's name, and the destination's directory.

For example: **mv file.txt /home/username/Documents**.
To rename files, the Linux command is **mv oldname.ext newname.ext**

mkdir command
Use **mkdir** command to make a new directory — if you type **mkdir Music** it will create a directory called **Music**.

There are extra **mkdir** commands as well:

- To generate a new directory inside another directory, use this Linux basic command **mkdir Music/Newfile**
- use the **p** (parents) option to create a directory in between two existing directories.

  For example, **mkdir -p Music/2020/Newfile** will create the new "2020" file.

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

rm command
The **rm** command is used to delete directories and the contents within them. If you only want to delete the directory — as an alternative to rmdir — use **rm -r**.

Note: Be very careful with this command and double-check which directory you are in. This will delete everything and there is no undo.

pwd command

Use the **pwd** command to find out the path of the current (present) working directory (folder) you're in. The command will return an absolute (full) path, which is basically a path of all the directories that starts with a forward slash **(/)**.

An example of an absolute path is **/home/username**.

file command

file command is used to determine the type of a file. **.file** type may be of human-readable(e.g. 'ASCII text') or MIME type(e.g. 'text/plain; charset=us-ascii'). This command tests each argument in an attempt to categorize it.

It has three sets of tests as follows:

- **Filesystem test:** This test is based on the result which returns from a stat system call. The program verifies that if the file is empty, or if it's some sort of special file. This test causes the file type to be printed.
- **Magic test:** These tests are used to check for files with data in particular fixed formats.
- **Language test:** This test search for particular strings which can appear anywhere in the first few blocks of a file.

Syntax:

file [option] [filename]

Example: Command displays the file type

file email.py
file name.jpeg
file Invoice.pdf
file exam.ods
file videosong.mp4

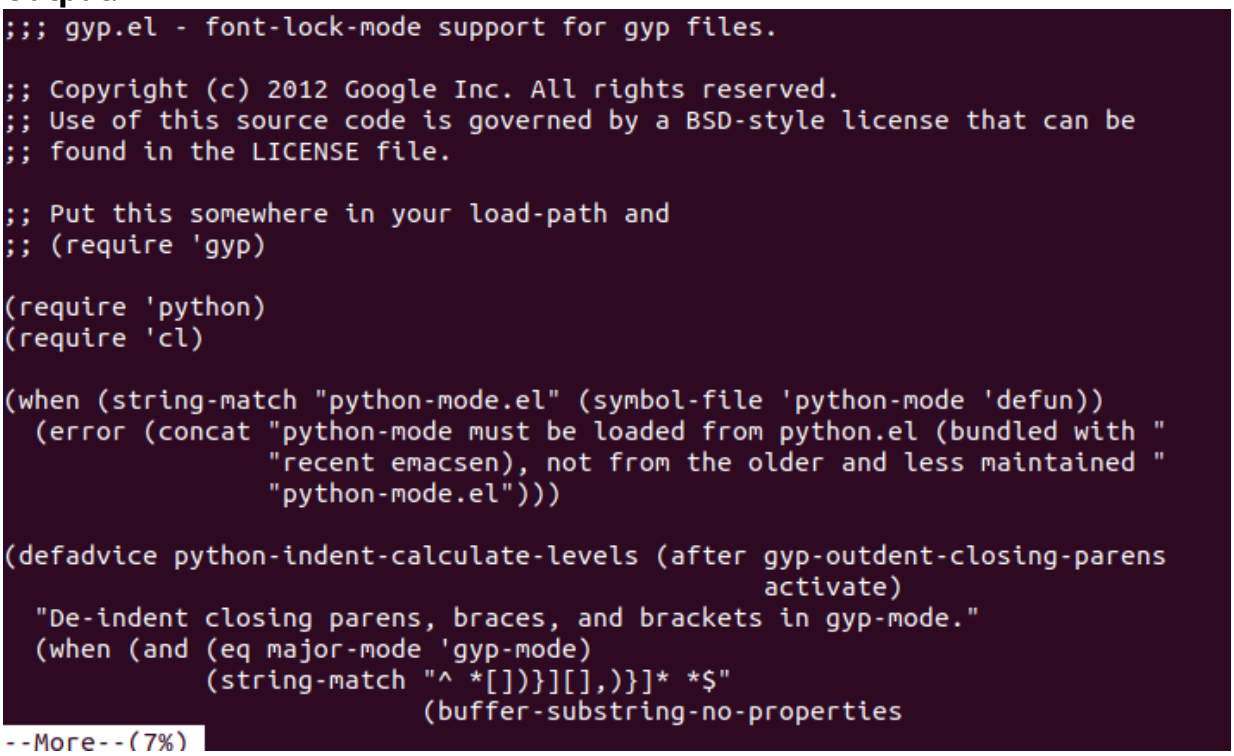
more command

The more command is quite similar to the cat command, as it is used to display the file content in the same way that the cat command does. The only difference between both commands is that, in case of larger files, the more command displays screenful output at a time.

In more command, the following keys are used to scroll the page:
ENTER key: To scroll down page by line.
Space bar: To move to the next page.
b key: To move to the previous page.
/ key: To search the string.

Syntax:
more **<file** name**>**

Output:



less Command

The less command is similar to the more command. It also includes some extra features such as 'adjustment in width and height of the terminal.' Comparatively, the more command cuts the output in the width of the terminal.

Syntax:

less **<file** name**>**

Output:

```
;;; gyp.el - font-lock-mode support for gyp files.

;; Copyright (c) 2012 Google Inc. All rights reserved.
;; Use of this source code is governed by a BSD-style license that can be
;; found in the LICENSE file.

;; Put this somewhere in your load-path and
;; (require 'gyp)

(require 'python)
(require 'cl)

(when (string-match "python-mode.el" (symbol-file 'python-mode 'defun))
  (error (concat "python-mode must be loaded from python.el (bundled with "
                 "recent emacsen), not from the older and less maintained "
                 "python-mode.el")))

(defadvice python-indent-calculate-levels (after gyp-outdent-closing-parens
                                                 activate)
```

Creating and Viewing files using cat

The cat (short for "concatenate") command is one of the most frequently used command in Linux/Unix like operating systems. cat command allows us to create single or multiple files, view contain of file, concatenate files and redirect output in terminal or files. Here, we are going to find out handy use of cat commands with their examples in Linux.

General Syntax

cat [OPTION] [FILE]...

1. Display Contents of File

In the below example, it will show contents of /etc/passwd file.

```
# cat /etc/passwd
```

Output

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
narad:x:500:500::/home/narad:/bin/bash
```

2. View Contents of Multiple Files in terminal

In below example, it will display contents of test and test1 file in terminal.

```
# cat test test1
```

Output

```
Hello everybody
Hi world,
```

3. Create a File with Cat Command

We will create a file called test2 file with below command.

```
# cat >test2
```

Awaits input from user, type desired text and press CTRL+D (hold down Ctrl Key and type 'd') to exit. The text will be written in test2 file. You can see content of file with following cat command.

```
# cat test2
```

Output
hello everyone, how do you do?

4. Use Cat Command with more & less Options

If file having large number of content that won't fit in output terminal and screen scrolls up very fast, we can use parameters more and less with cat command as show above.

cat song.txt | more

cat song.txt | less

5. Display Line Numbers in File
With -n option you could see the line numbers of a file song.txt in the output terminal.

cat -n song.txt

Output
1 "Heal The World"
2 There's A Place In
3 Your Heart
4 And I Know That It Is Love
5 And This Place Could
6 Be Much
7 Brighter Than Tomorrow
8 And If You Really Try
9 You'll Find There's No Need
10 To Cry
11 In This Place You'll Feel
12 There's No Hurt Or Sorrow

6. Display $ at the End of File
In the below, you can see with -e option that '$' is shows at the end of line and also in space showing '$' if there is any gap between paragraphs. This options is useful to squeeze multiple lines in a single line.
cat -e test

Output
hello everyone, how do you do?
$

$
Hey, am fine.$
How's your training going on?$
$

7. Display Tab separated Lines in File

In the below output, we could see TAB space is filled up with '^I' character.

cat -T test

Output

hello ^Ieveryone, how do you do?

Hey, ^Iam fine.
^I^IHow's your training ^Igoing on?

8. Display Multiple Files at Once

In the below example we have three files test, test1 and test2 and able to view the contents of those file as shown above. We need to separate each file with ; (semi colon).

cat test; cat test1; cat test2

Output

This is test file
This is test1 file.
This is test2 file.

9. Use Standard Output with Redirection Operator

We can redirect standard output of a file into a new file else existing file with '>' (greater than) symbol. Careful, existing contents of test1 will be overwritten by contents of test file.

cat test > test1

10. Appending Standard Output with Redirection Operator

Appends in existing file with '>>' (double greater than) symbol. Here, contents of test file will be appended at the end of test1 file.

cat test >> test1

11. Redirecting Standard Input with Redirection Operator

When you use the redirect with standard input '<' (less than symbol), it use file name test2 as a input for a command and output will be shown in a terminal.

```
# cat < test2
```

Output

This is test2 file.

12. Redirecting Multiple Files Contain in a Single File

This will create a file called test3 and all output will be redirected in a newly created file.

```
# cat test test1 test2 > test3
```

13. Sorting Contents of Multiple Files in a Single File

This will create a file test4 and output of cat command is piped to sort and result will be redirected in a newly created file.

```
# cat test test1 test2 test3 | sort > test4
```

File Comparisons

cmp Command

cmp command in Linux/UNIX is used to compare the two files byte by byte and helps you to find out whether the two files are identical or not.

- When cmp is used for comparison between two files, it reports the location of the first mismatch to the screen if difference is found and if no difference is found i.e the files compared are identical.
- cmp displays no message and simply returns the prompt if the the files compared are identical.

Syntax:
cmp [OPTION]... FILE1 [FILE2 [SKIP1 [SKIP2]]]

where SKIP1, SKIP2 & OPTION are optional and FILE1 & FILE2 refer to the filenames .

The syntax of cmp command is quite simple to understand. If we are comparing two files then clearly we will need their names as arguments (i.e. as FILE1 & FILE2 in syntax).

In addition to this, the optional SKIP1 and SKIP2 specify the number of bytes to skip at the beginning of each file which is zero by default and OPTION refers to the options compatible with this command about which we will discuss later on.

cmp Example : As explained that the cmp command reports the byte and line number if a difference is found. Now let's find out the same with the help of an example. Suppose there are two files which you want to compare one is file1.txt and other is file2.txt :

$cmp file1.txt file2.txt

1. **If the files are not identical :** the output of the above command will be :
   $cmp file1.txt file2.txt
   file1.txt file2.txt differ: byte 9, line 2

   /*indicating that the first mismatch found in two files at byte 20 in second line*/

2. **If the files are identical :** you will see something like this on your screen:
   $cmp file1.txt file2.txt
   $ _
   /*indicating that the files are identical*/