LINUX

CHAPTER 1

- Linux is an open-source operating system like other operating systems such as Microsoft Windows, Apple Mac OS, iOS, Google android, etc.
- An operating system is software that enables the communication between computer hardware and software.
- It conveys input to get processed by the processor and brings output to the hardware to display it. This is the basic function of an operating system.
- Like other operating systems, Linux has a graphical interface, and the types of software you are accustomed to using on other operating systems, such as word processing applications, have Linux equivalents.
- But Linux also is different from other operating systems in many important ways. First, and perhaps most importantly, Linux is open-source software. The code used to create Linux is free and available to the public to view, edit, and—for users with the appropriate skills contribute to.
- Linux is also different in that, although the core pieces of the Linux operating system are generally common, there are many distributions of Linux, which include different software options. This means that Linux is incredibly customizable, because not just applications, such as word processors and web browsers, can be swapped out. Linux users also can choose core components, such as which system displays graphics, and other user-interface components.
- Companies and individuals choose Linux for their servers because it is secure, and you can receive excellent support from a large community of users, many

of the devices you own probably, such as Android phones, digital storage devices, personal video recorders, cameras, wearables, and more, also run Linux.

- It was originally conceived of and created as a hobby by Linus Torvalds in 1991. Linus, while at university, sought to create an alternative, free, open-source version of the MINIX operating system, which was itself based on the principles and design of Unix.

Features of the Linux Operating System

1. Multiprogramming – It has the capability of running various programs by several users at a time. This feature of Linux is called multiprogramming.
2. Time-Sharing – All the programs are queued and given CPU time. It compels Operating System to finish its work in a time specified. In case it is not completed any way the very task is automatically put in the queue again.
3. Portable Environment -Linux software operates flawlessly on a variety of hardware platforms. Without the worry of incompatibility, individuals

can use Linux operating system on any device. It runs the same way on both high-end and low-end hardware.

4. Free and Open-Source -Its source code is available for anybody to use and alter. Many developers collaborate in organizations to improve and strengthen Linux, and lots of developers constantly work on updating the Linux system.
5. Shell/ Command-line Interface - The Linux system includes essential programs that users can utilize in order to issue commands to the operating system for executing the design flawlessly. You may also direct it to carry out various forms of Linux commands for effectively carrying out the applications.
6. End-to-end encryption - Authentication can help you keep your data protected. Before you may access some critical files, the Linux Operating System requires you to enter a password. Furthermore, the Linux environment allows users to encrypt their data.
7. Graphical User Interface (GUI) - Linux Operating System comes with Graphical User Interface (GUI) abilities in the same way you can with Windows. Similarly, users can install the programs, and the computer graphics will begin to work in the same way that Windows does.
8. Configure Keyboards in Different Languages - Because Linux is available in various languages, it is simple to use it worldwide. As a result, you can change the language on your keyboard as per your preference.
9. Frequent New Updates - Software updates are controlled by the users in Linux. Individuals have the option to pick and choose which updates are required, and there are plenty of system updates accessible. These

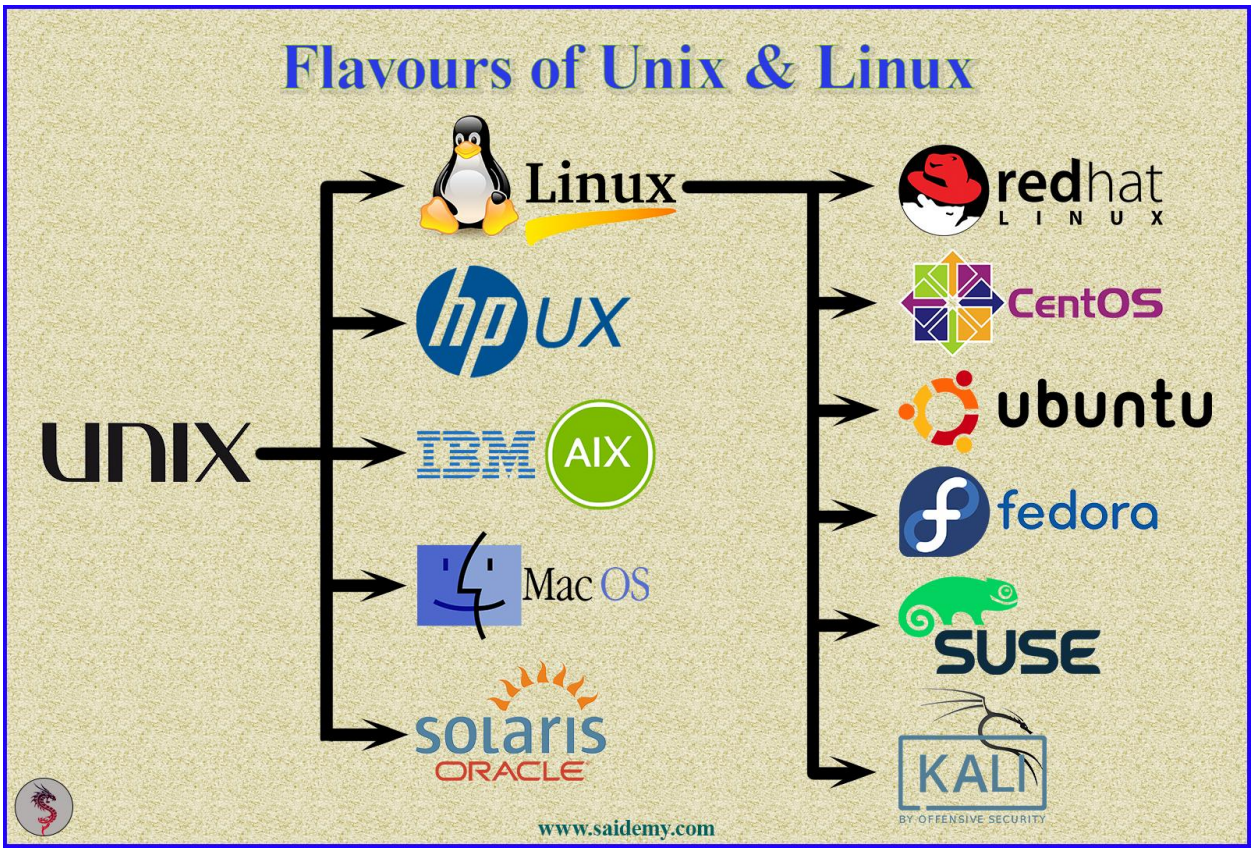
upgrades happen considerably more quickly than on other operating systems. Therefore, system upgrades can be deployed without difficulty.

10. Lightweight Infrastructure - Linux is a highly lightweight operating system. Linux has far fewer prerequisites than any other operating system, has a smaller memory footprint, and uses less storage space. Typically, you'll find a Linux Distro with only 128MB of RAM and around the same amount of disc space.
11. Extremely Flexible - Linux is highly flexible, and a variety of desktop applications, embedded systems, and server applications can benefit from the same. It also offers a number of computer-specific limitation settings for admins to allow only essential components to get installed.
12. Best For Developers - Linux supports nearly every popular programming language, including C/C++, Java, Python, Ruby, etc. Furthermore, it provides a wide range of development-related applications.

    A majority of developers all around the world prefer the Linux terminal over the Windows command line. On a Linux system, the package manager aids programmers in understanding how things are performed. It also comes with features like bash scripting and supports SSH that aids in the speedy management of servers.

Linux Distribution

Generally, the Linux distribution comprises a Linux kernel, which is the heart of Linux: libraries, GNU tools, documentation, X Window, desktop environment, and free and open-source software. Mostly, the distributions available to download are .deb or .rpm files. The .deb files are meant for Debian distributions of Linux (like Ubuntu, Linux Mint, etc.), while the .rpm files are used primarily by the Redhat-based distros (like CentOS, Fedora, RHEL) and the OpenSUSE distribution.



LINUX vs UNIX

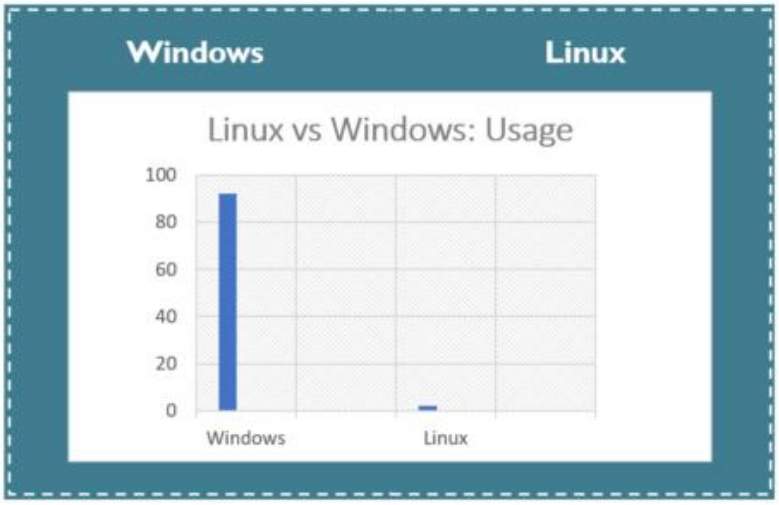
PARAMETERS	LINUX	UNIX
Inception Year	1991	1969
Standard	Open source operating system which is freely available	Operating system can only be used by its copywriters
System type	Just the kernel	Complete Operating system
Target use	Can be used by anyone including home user and developer.	Developed mainly for servers, workstations and mainframes.
Cost	LINUX is freely available and distributed with no associated cost.	UNIX variants come as customized cost.
Security	60-100 viruses listed till date	85-120 viruses listed till date
Interface type	Primarily uses GUI with option of CLI	Primarily uses CLI
Portability	Portable	Not portable
Variants	Ubuntu, RedHat, Solaris, OpenSuse, etc.	AIS, HP-UX, BSD, etc
Source Code	The source code of Linux is available in general public.	The source code not available in general public.

Linux vs Windows

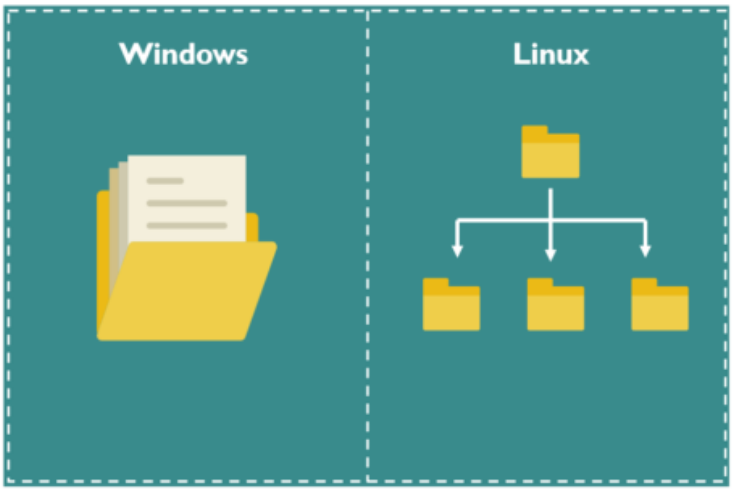
- Users

There are 3 types of users in Linux (Regular, Administrative (root)) and Service users) whereas, in Windows, there are 4 types of user accounts (Administrator, Standard and Guest).

- Usage

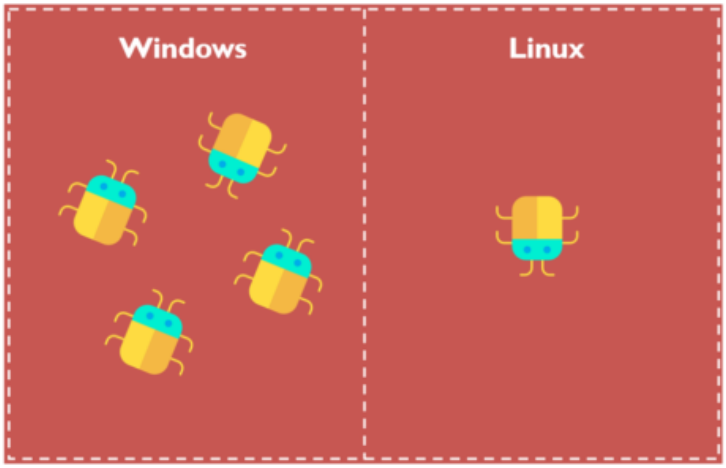


- File Systems



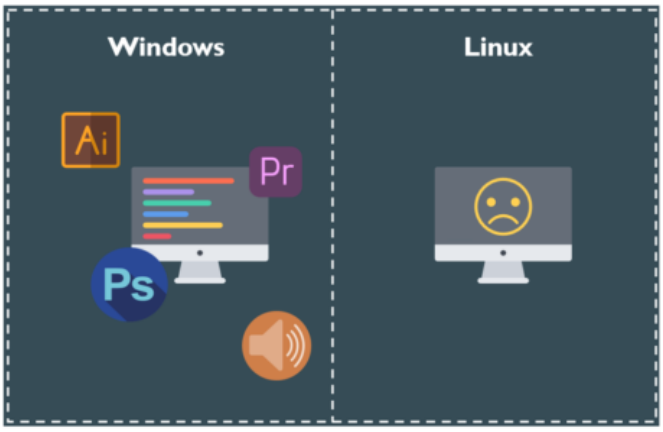
In Microsoft Windows, files are stored in directories/folders on different data drives like C: D: E: but, in Linux, files are ordered in a tree structure starting with the root directory, further branched out to various other sub-directories.

- Security



Every Windows user has faced security and stability issues at some point in time. Since Windows is an extensively used OS, hackers, and spammers target Windows frequently. Windows (consumer versions) were originally designed for ease of use on a single-user PC without a network connection and did not have security features built-in. Microsoft often releases security patches through its Windows Update service. These go out once a month, although critical updates are made available at shorter intervals or when necessary.

- Compatibility

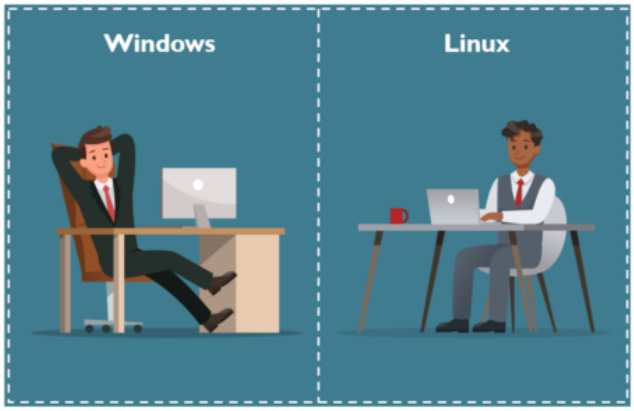


Windows shoots. Windows scores. This is where the Redmond offering wipes the floor with Linux. Despite recent improvements in software being ported or developed to Linux, Windows is still the king of compatibility.

Users of Windows can be certain that almost any software (even the most obscure, outdated software) will work, even when it is abandoned by developers. Windows has great legacy support. Plain and simple.

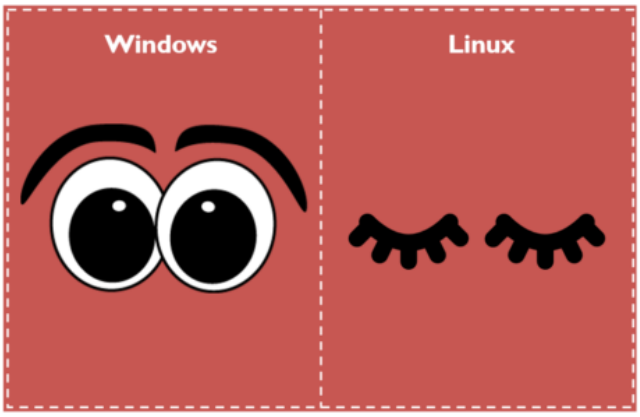
Linux, on the other hand, has been struggling with basics that Windows users take for granted.

- Ease of Use



Linux, over recent years, has gone leaps and bounds when it comes to usability. Distributions like Linux Mint and Ubuntu, have even gone as far as making their *installation and setup* simpler for non-technical users to carry on with day-to-day activities with the utmost ease.

- Privacy



If you are a Linux user, you have an operating system that doesn't *spy* on you. Having Linux means the system is *yours and yours alone*. You can also add to the mix that most Linux systems come with an option of *built-in military-grade encryption.* As a user, you can be sure that device theft poses no real problem to your data.

On the contrary, Windows has gotten more advert-driven over the last few years. Users are definitely given the choice to opt-out but then again, who can help the clever registry hacks which are clearly a part of Redmond's plan. Windows can watch what users do, offering to sync to the Microsoft One-Drive service or to learn behavior to make Cortana (the Microsoft personal assistant) better. To be honest, I do not favor these tools, as they're pretty intrusive. Though, some users like these features. Subjective opinion.

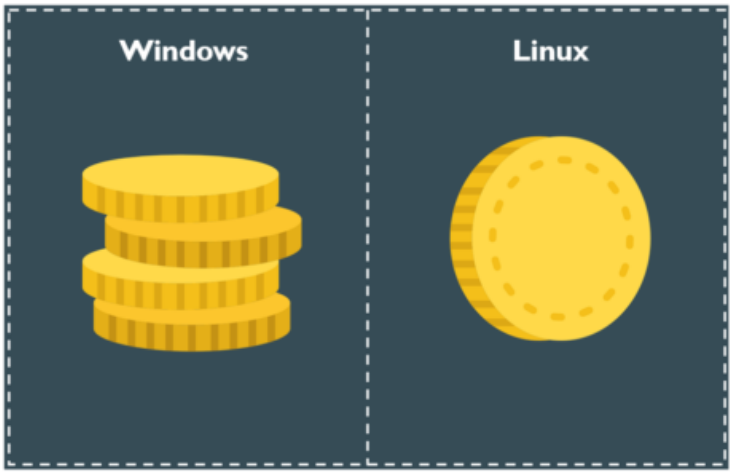
- Source Code

Linux is an open-source operating system whereas Windows OS is commercial. Linux has access to the source code and alters the code as per user need whereas Windows does not have access to the source code.

In Linux, the user has access to the source code of the kernel and alters the code according to his need. It has its own advantages. Bugs in the OS will get fixed at a rapid pace but developers may take advantage of any weakness in the OS if found.

In windows only selected members to have access to the source code.

- License



- Reliability

Windows, as we all have come to know, becomes sluggish by the day. You have to re-install Windows after a while when you encounter crashes or slowdowns on your system.

If you are a Linux user, you will not have to re-install it just to experience a faster and smoother system. Linux helps your system run smoothly for a longer period.

- Programming

Linux supports almost all of the major programming languages (*Python, C/C++, Java, Ruby, Perl, etc*.). Moreover, it portrays a vast range of applications useful for programming purposes.

The Linux terminal is far superior to use over Window's command line for developers. You would find many libraries natively developed for Linux. Also, a lot of programmers point out that they can do things easily using the package manager on Linux. The ability to script in *different shells* is also one of the most compelling reasons why programmers prefer using Linux OS.

Advantages of Linux

1. Open-source software. The Linux kernel is released under the GNU GPL open-source software license. Most distros include hundreds of applications, with many options in almost every category. Many distributions also include proprietary software, such as device drivers provided by manufacturers, to support their hardware.
2. Licensing costs. Unlike Microsoft Windows or Apple macOS, Linux has no explicit licensing fees. While system support is available for a fee from many Linux vendors, the OS itself is free to copy and use. Some IT organizations have increased their savings by switching their server software from a commercial OS to Linux.
3. Reliability. Linux is considered a reliable OS and is well-supported with security patches. Linux is also considered to be stable, meaning it can run in most circumstances. Linux also copes with errors when running software and unexpected input.
4. Backward compatibility. Linux and other open-source software tend to be updated frequently for security and functional patches while retaining core functionality. Configurations and shell scripts are likely to work unchanged even when software updates are applied. Unlike commercial software vendors that roll out new versions of their OSes along with new ways to work, Linux and open source applications generally don't change their modes of operation with new releases.
5. Many choices. Between the hundreds of available distributions, thousands of applications, and almost infinite options for configuring,

compiling, and running Linux on almost any hardware platform, it is possible to optimize Linux for almost any application.

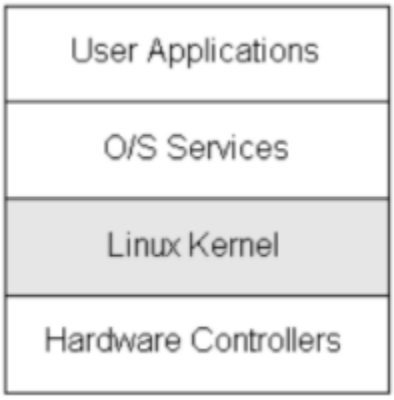
Disadvantages of Linux

1. Lack of established standard. There is no *standard* version of Linux, which may be good for optimizing Linux for particular applications, but less so for deploying standardized server or desktop images. The wide range of options can complicate support as a result.
2. Support costs. While an organization can acquire Linux freely without licensing fees, support is not free. Most enterprise Linux distributors like SUSE and Red Hat offer support contracts. Depending on the circumstances, these license fees can reduce savings significantly.
3. Proprietary software. Desktop productivity software like Microsoft Office cannot be used on Linux desktops, and other proprietary software may be unavailable for Linux platforms.
4. Unsupported hardware. While many hardware manufacturers make Linux device drivers available for their products, many do not.
5. Steep learning curve. Many users struggle to learn to use the Linux desktop or Linux-based applications.

Linux Operating System

The Linux operating system is composed of four major subsystems. These are:

1. User Applications — The set of applications in use on a particular Linux. The system will be different depending on what the computer system is used for, but typical examples include a word-processing application and a web browser.



2. O/S Services — These are services that are typically considered part of the operating system (a windowing system, command shell, etc.); also, the programming interface to the kernel (compiler tool and library) is included in this subsystem.

3. Linux Kernel — The kernel abstracts and mediates access to the hardware resources, including the CPU.

4. Hardware Controllers — This subsystem is comprised of all the possible physical devices in a Linux installation; for example, the CPU, memory hardware, hard disks, and network hardware are all members of this sub-system.

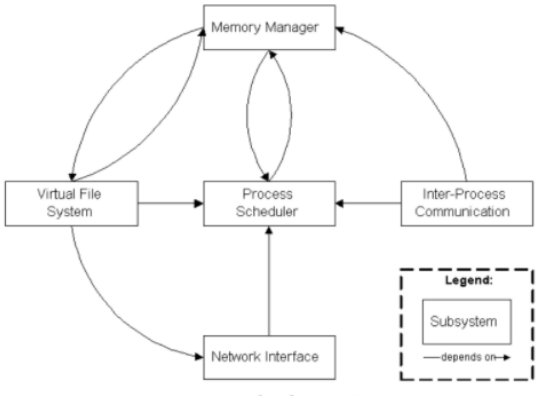
1. KERNEL

- The kernel is the heart of the Linux operating system. Like any kernel, it acts as the brain of the system and controls how the hardware and software interact, plus decides which activity a computer should carry out at any particular instant.
- The kernel also makes sure that processes and daemons (server processes) are started and stopped at the exact right times.

Definition: "The kernel is the center of the Linux operating system. It functions to control processes, handles memory management, and manages software and hardware communications."

- The Linux kernel is composed of five main subsystems:
  1. The Process Scheduler (SCHED): It is responsible for controlling process access to the CPU. The scheduler enforces a policy that ensures that processes will have fair access to the CPU while ensuring that necessary hardware actions are performed by the kernel on time.
  2. The Memory Manager (MM): It permits multiple processes to securely share the machine's main memory system. In addition, the memory manager supports virtual memory that allows Linux to support processes that use more memory than is available in the system. Unused memory is swapped out to persistent storage using the file system and then swapped back in when it is needed.
  3. The Virtual File System (VFS): VFS abstracts the details of the variety of hardware devices by presenting a common file interface to all devices. In addition, the VFS supports several file system formats that are compatible with other operating systems.

4. The Network Interface (NET): It provides access to several networking standards and a variety of network hardware.
5. The Inter-Process Communication (IPC): The IPC subsystem supports several mechanisms for process-to-process communication on a single Linux system.



2. SHELLS AND GUIS

- A shell is a program that acts as an intermediary between user and the backbone (Kernel) of the operating system.
- Linux shells have more interesting names (like bash, etc.), but they do pretty much the same thing. In addition to translating your commands into something the kernel can understand and act upon, the shell adds some important functions that the base operating system doesn't supply.
- Using a Linux shell means working with a command line, which is much like working from a DOS prompt. Modern versions of Linux come with a graphical user interface (GUI) but some Linux tasks can only be done from the command line. The shell is an advanced way of communicating

with the system because it allows for two-way conversation and taking initiative. Both partners in the communication are equal, so new ideas can be tested. The shell allows the user to handle a system in a very flexible way.

- An additional asset is that the shell allows for task automation.
- The Linux OS is controlled by the kernel, which is the heart of the entire system. However, the kernel can only understand machine code. This is why a shell must be used. The shell interprets commands given by the user and translates them into machine code that the kernel can understand.
- The basic features of all Linux shells are following:
    - Prompts: A prompt is a character or string of characters (such as $ or #) that the shell displays when it is ready to receive a new command.
    - Command resolution: When you enter a command, the shell must determine which program to run in order to perform that command.
    - Job control: Linux lets you multitask (run more than one command at a time).
    - Command history and completion: When you're entering lots of commands, sometimes you want to repeat the previous command or issue a similar one.
    - Wildcards and aliases: Wildcards let you process a whole bunch of files at once, instead of having to repeat the same command for each file.

- Piping and I/O redirection: Sending the output of one program directly to another program or to a file can save you time and keystrokes.

3. SYSTEM UTILITIES and APPLICATION PROGRAMS

- System libraries are special functions, that are used to implement the functionality of the operating system and do not require code access rights of kernel modules.
- System Utility programs are liable to do individual and specialized-level tasks.
- A daemon is also called background processes. It is a UNIX or Linux program that executes inside the background.
- Almost every daemon contains names that finish with the "d" letter. For example, sshd, manages connections of SSH remote access, or the httpd daemon manages the Apache server.

Few Application Programs Are:

1. LibreOffice

The Linux replacement for Microsoft Office, this can do everything Microsoft Office can do, and you don't have to pay hundreds of dollars for it. Most distros include LibreOffice by default.

2. gscan2pdf

A simple app for scanning documents to the PDF format, which students may find useful, but is also helpful when you need to upload official documents you may have.

3. Notepadqq

If you're a fan of Notepad++ on Windows, you'll be glad to know that the developer has a Linux port called notepadqq. It may not be in your distro's official repositories, but you can always add a PPA or install the package manually.

4. WINE

Stands for WINE Is Not an Emulator, allows you to run Windows applications on Linux. It's not perfect, many applications and games work flawlessly, but some just won't no matter how hard you try. Your mileage may vary.

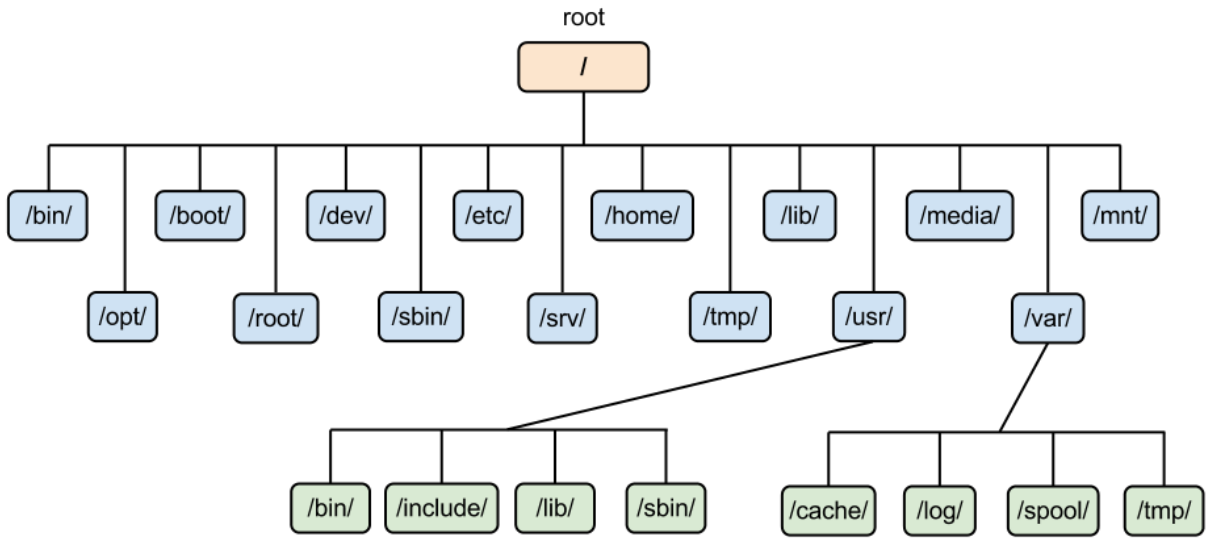
5. PlayOnLinux

A "helper" application for WINE, which can make the installation of certain games much easier.

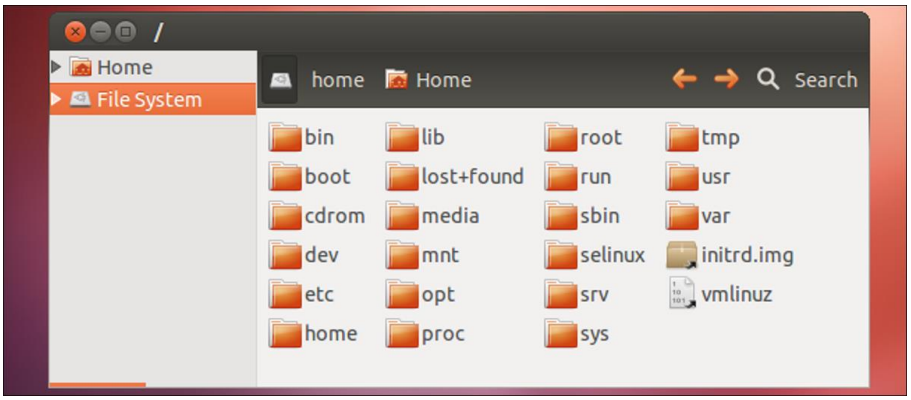
6. Gimp

Linux's answer to Photoshop. Probably the most popular image editor on Linux.

Linux Directory Structure



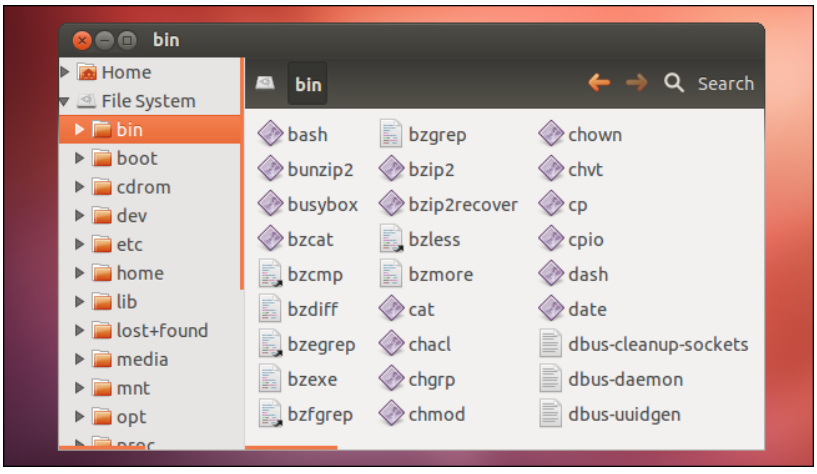
The Filesystem Hierarchy Standard (FHS) defines the structure of file systems on Linux and other UNIX-like operating systems. The file system resembles an upside-down tree and is very similar to the treelike directory structure in DOS. The top level is denoted by a slash (/) and is called the root directory. This (/) is the base, or root, of the filesystem. Everything in the Linux system is housed in this directory. Several system-related directories such as bin, dev, and etc. appear below the root directory.



Following is a summary of their purpose in life:

(1) /bin: Pronounced “slash bin.” **Essential User Binaries.**

- Unlike /sbin, the bin directory contains several useful Linux commands that are of use to both the system administrator as well as non-privileged users.
- It usually contains the shells like bash, csh, etc.... and commonly used commands like cp, mv, rm, cat, ls etc.
- For this reason and in contrast to /usr/bin, the binaries in this directory are considered to be essential. The reason for this is that it contains essential system programs that must be available even if only the partition containing/is mounted.
- This situation may arise should you need to repair other partitions but have no access to shared directories (ie. you are in single user mode and hence have no network access).
- Applications such as Firefox are stored in /usr/bin, while important system programs and utilities such as the bash shell are located in /bin.

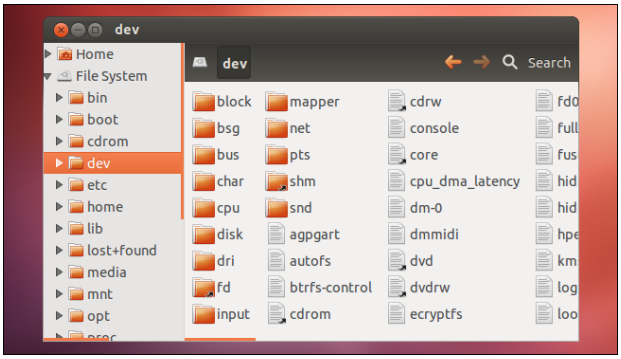


(2) /boot: **Static Boot Files**

- The /boot directory contains everything required for the boot process except for configuration files not needed at boot time (the most notable of those being those that belong to the GRUB boot-loader) and the map installer.
- Thus, the /boot directory stores data that is used before the kernel begins executing user-mode programs.
- This may include redundant (back-up) master boot records, sector/system map files, the kernel and other important boot files and data that is not directly edited by hand. Programs necessary to arrange for the boot loader to be able to boot a file are placed in /sbin.
- Configuration files for boot loaders are placed in /etc. The system kernel is located in either / or /boot.

(3) /dev: **Device Files.**

- The /dev directory contains special device files that correspond to hardware components. It is a very interesting directory that highlights one important aspect of the Linux filesystem - everything is a file or a directory.
- Look through this directory and you should hopefully see hda1, hda2 etc.... which represent the various partitions on the first master drive of the system. /dev/cdrom and /dev/fd0 represent your CD-ROM drive and your floppy drive.



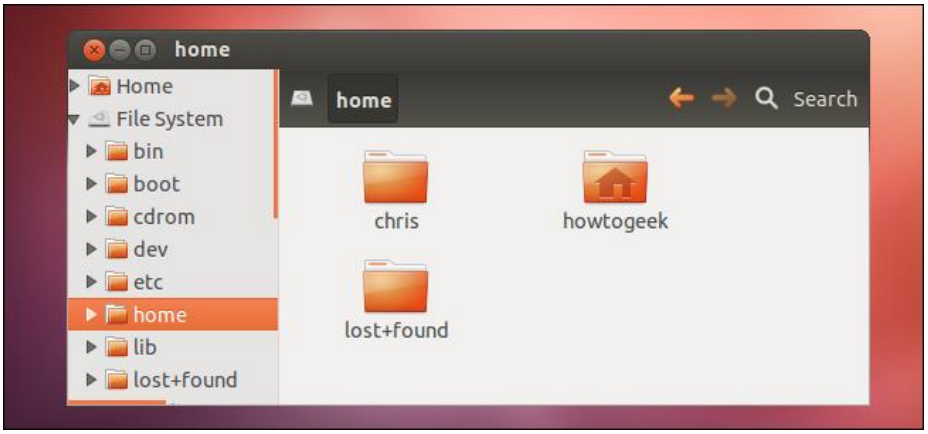
(4) /etc: **Configuration Files.**

- The /etc directory contains configuration files for Linux and other installed software.
- This is the nerve center of your system, it contains all system related configuration files in here or in its sub-directories.
- A "configuration file" is defined as a local file used to control the operation of a program; it must be static and cannot be an executable binary. For this reason, it's a good idea to backup this directory regularly. It will definitely save you a lot of re-configuration later if you re-install or lose your current installation.
- Normally, no binaries should be or are located here.

(5) /home: **Home Folders.**

- The /home directory contains the home directories (personal storage) for each user on the system.
- Linux is a multi-user environment so each user is also assigned a specific directory that is accessible only to them and the system administrator.

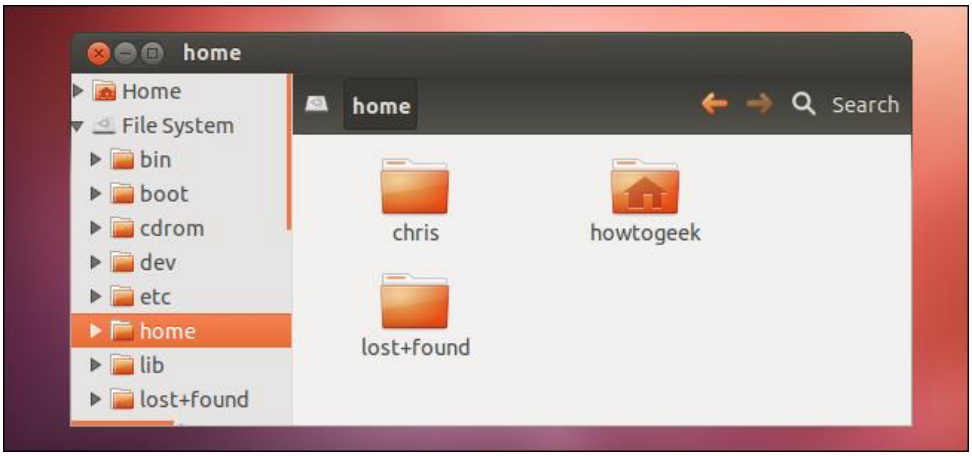
- These are the user home directories, which can be found under '/home/$USER' (~/).
- It is your playground: everything is at your command, you can write files, delete them, install programs, etc.. . Your home directory contains your personal configuration files, the so-called dot files (their name is preceded by a dot).
- Personal configuration files are usually 'hidden', if you want to see them, you either have to turn on the appropriate option in your file manager or run ls with the -a switch. If there is a conflict between personal and system-wide configuration files, the settings in the personal file will prevail.
- The /home can get quite large and can be used for storing downloads, compiling, installing, and running programs, your mail, your collection of image or sound files, etc.



(6) /root: **Root Home Directory.**

- The /root is a home directory for the root user; not to be confused with /.

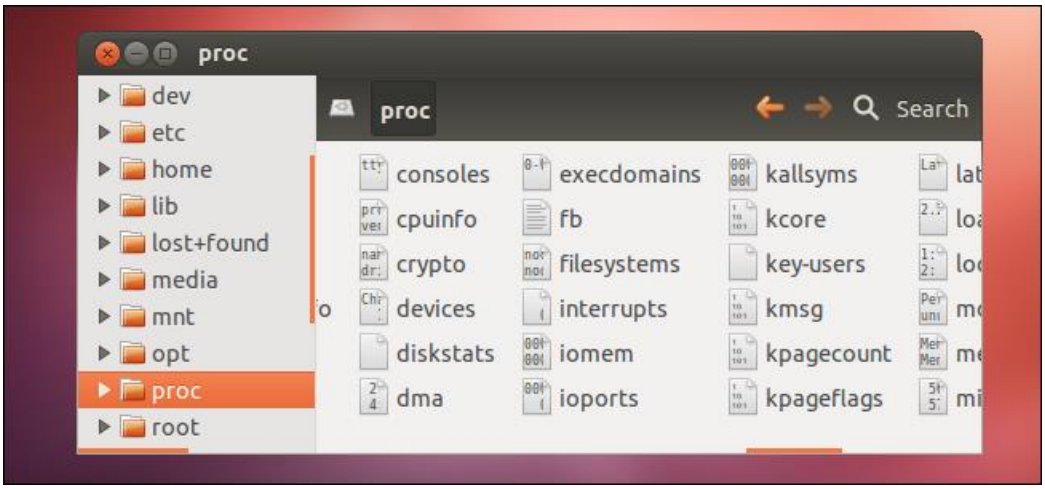
- This is the home directory of the System Administrator, 'root'. This may be somewhat confusing ('root on root') but in former days, '/' was root's home directory (hence the name of the Administrator account).
- To keep things tidier, 'root' got his own home directory. Why not in '/home'? Because '/home' is often located on a different partition or even on another system and would thus be inaccessible to 'root' when - for some reason - only '/' is mounted.



(7) /proc: **Kernel & Process Files.**

- This directory provides a mechanism for the kernel to send information to processes.
- /proc is very special in that it is also a virtual filesystem. It's sometimes referred to as a process information pseudo-file system.
- It doesn't contain 'real' files but runtime system information (e.g. system memory, devices mounted, hardware configuration, etc).

- For this reason it can be regarded as a control and information centre for the kernel. In fact, quite a lot of system utilities are simply calls to files in this directory.



(8) /lib**: Essential Shared Libraries.**

- The /lib directory contains kernel modules and the shared library images (the C programming code library) needed to boot the system and run the commands in the root filesystem, ie. By binaries in /bin and /sbin.
- Libraries are readily identifiable through their filename extension of *.so. Windows equivalent to a shared library would be a DLL (dynamically linked library) file.
- They are essential for basic system functionality. Kernel modules (drivers) are in the subdirectory /lib/modules/'kernel-version'.

(9) /media: **Removable Media**

- The /media directory provides a location for mounting devices, such as remote file systems and removable media (with directory names of cdrom, floppy, and so on).
- In Fedora and RHEL, many removable media are mounted automatically in this directory when the media is inserted (CD or DVD) or connected (USB pen drives or cameras).

(10) /mnt: **Temporary Mount Points.**

- This is a generic mount point under which you mount your filesystems or devices.
- Mounting is the process by which you make a filesystem available to the system. After mounting your files will be accessible under the mount-point. This directory usually contains mount points or sub-directories where you mount your floppy and your CD.
- Standard mount points would include /mnt/cdrom and /mnt/floppy. There is no limitation to creating a mount-point anywhere on your system but by convention and for sheer practicality do not litter your file system with mount-points.
- It should be noted that some distributions like Debian allocate /floppy and /cdrom as mount points while Redhat put them in /mnt/floppy and /mnt/cdrom respectively.

(11) /opt: **Optional Packages.**

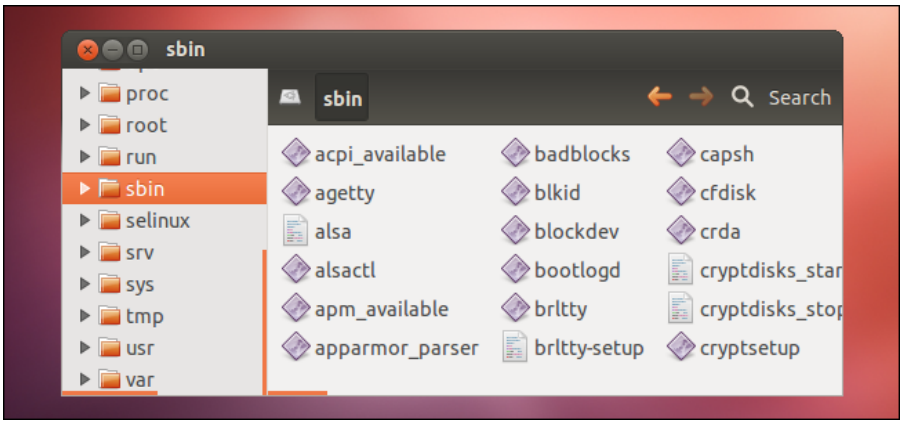
- This directory is reserved for all the software and add-on packages that are not part of the default installation.

- For example, StarOffice, Netscape Communicator, and WordPerfect packages are normally found here.
- To comply with the File System, all third-party applications should be installed in this directory. Any package to be installed here must locate its static files (ie. Extra fonts, clipart, database files) must locate its static files in a separate /opt/'package' or /opt/'provider' directory tree (similar to the way in which Windows will install new software to its own directory tree C:\Windows\ProgamFiles\" Program Name"), where 'package' is a name that describes the software package and 'provider' is the provider's LANANA registered name.

(12) /sbin: **System Administration Binaries.**

- The /sbin directory contains more Linux binaries (special utilities not for general users). Linux discriminates between 'normal' executables and those used for system maintenance and/or administrative tasks.
- The latter reside either here or - the less important ones - in /usr/sbin.
- Locally installed system administration programs should be placed into /usr/local/sbin.
- Programs executed after /usr is known to be mounted (when there are no problems) are generally placed into /usr/sbin. This directory contains binaries that are essential to the working of the system.

- These include system administration as well as maintenance and hardware configuration programs. You may find lilo, fdisk, init, ifconfig, etc.... here.



(13) /srv: **Service Data.**

- The /srv directory contains site-specific data which is served by this system.
- The main purpose of specifying this is so that users may find the location of the data files for a particular service, and so that services that require a single tree for read-only data, writable data, and scripts (such as CGI scripts) can be reasonably placed.
- Data that is only of interest to a specific user should go in that user's home directory.

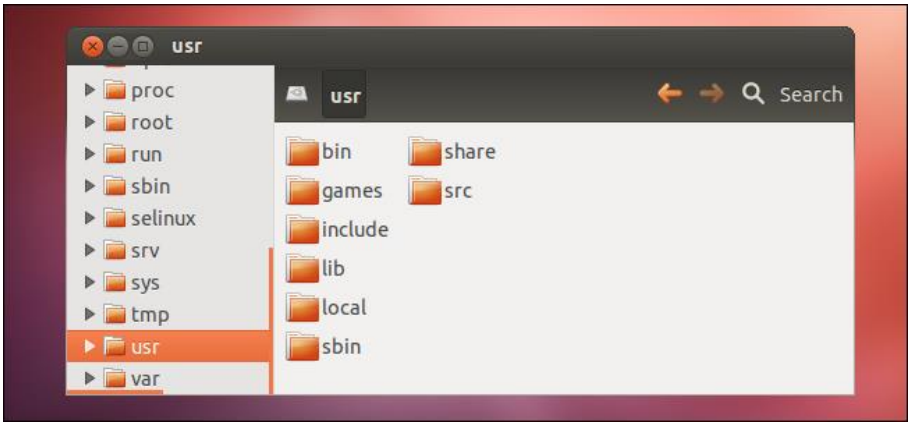
(14) /tmp: **Temporary Files.**

- The /tmp directory contains temporary files used by applications.
- Many programs use this to create lock files and for the temporary storage of data. Do not remove files from this directory unless you know exactly what you are doing!

- Many of these files are important for currently running programs and deleting them may result in a system crash.
- Usually, it won't contain more than a few KB anyway. On most systems, this directory is cleared out at boot or at shutdown by the local system. The basis for this was historical precedent and common practice.

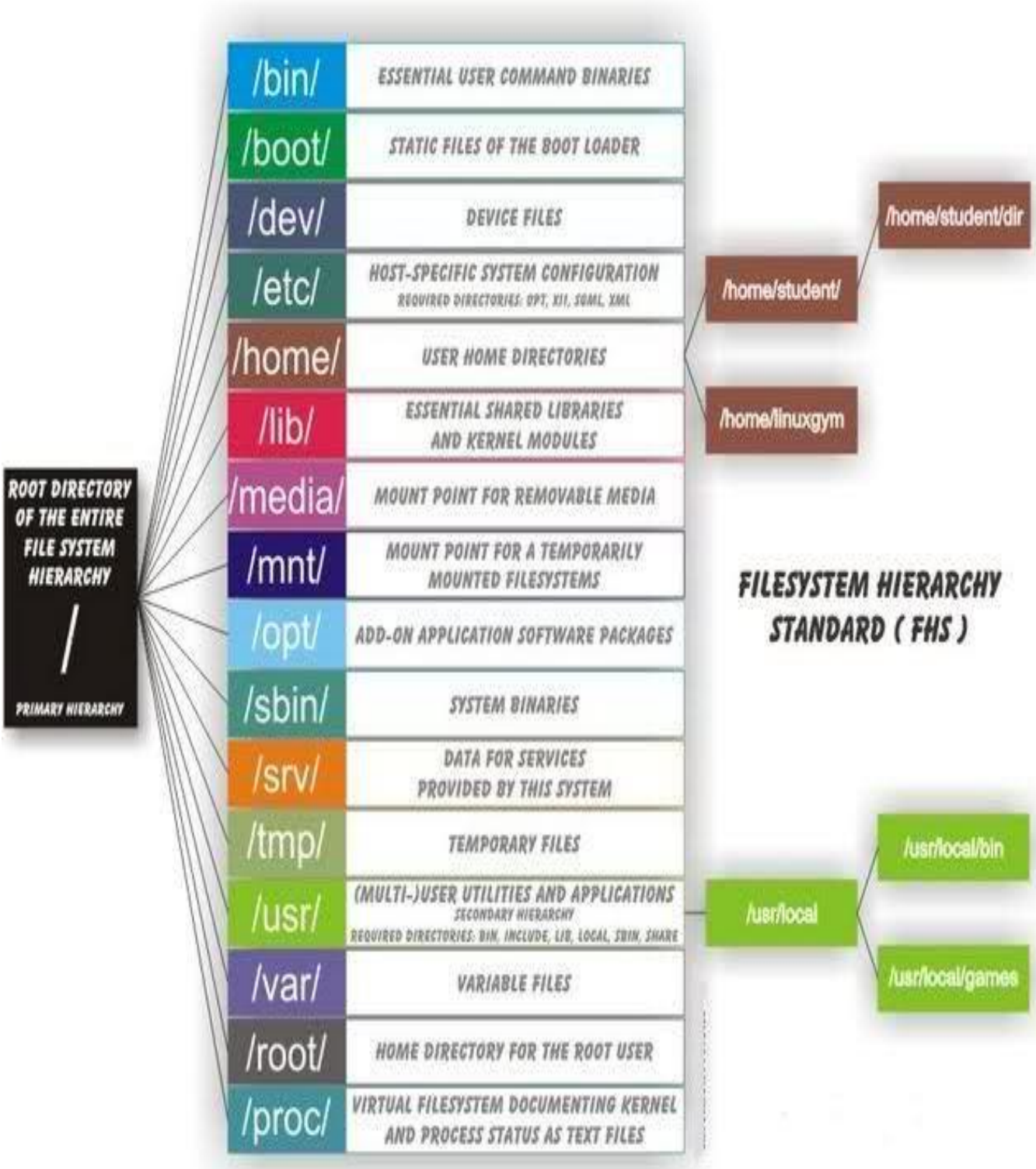
(15) /usr: **User Binaries & Read-Only Data.**

- The /usr directory contains system programs and other files for general users such as games, online help, and documentation.
- By convention, a user should not put personal files in this directory. This is one of the most important directories in the system as it contains all the user binaries, their documentation, libraries, header files, etc. and its supporting libraries can be found here.
- User programs like telnet, FTP, etc.... are also placed here.
- The /usr/local directory is where locally compiled applications install to by default — this prevents them from mucking up the rest of the system.



(16) /var: **Variable Data Files.**

- The /var directory contains directories of data used by various applications.
- In particular, this is where you would place files that you share as an FTP server (/var/ftp) or a Web server (/var/www).
- It also contains all system log files (/var/ log). Log files and everything else that would normally be written to /usr during normal operation are written to the /var directory.
- In time, FTP, HTTP, and similar services will move to the /srv directory to adhere to the Linux Standards Base.


ROOT DIRECTORY OF THE ENTIRE FILE SYSTEM HIERARCHY
/
PRIMARY HIERARCHY
/bin/
ESSENTIAL USER COMMAND BINARIES
/boot/
STATIC FILES OF THE BOOT LOADER
/dev/
DEVICE FILES
/etc/
HOST-SPECIFIC SYSTEM CONFIGURATION
REQUIRED DIRECTORIES: OPT, X11, SGML, XML
/home/
USER HOME DIRECTORIES
/lib/
ESSENTIAL SHARED LIBRARIES AND KERNEL MODULES
/media/
MOUNT POINT FOR REMOVABLE MEDIA
/mnt/
MOUNT POINT FOR A TEMPORARILY MOUNTED FILESYSTEMS
/opt/
ADD-ON APPLICATION SOFTWARE PACKAGES
/sbin/
SYSTEM BINARIES
/srv/
DATA FOR SERVICES PROVIDED BY THIS SYSTEM
/tmp/
TEMPORARY FILES
/usr/
(MULTI-)USER UTILITIES AND APPLICATIONS
SECONDARY HIERARCHY
REQUIRED DIRECTORIES: BIN, INCLUDE, LIB, LOCAL, SBIN, SHARE
/var/
VARIABLE FILES
/root/
HOME DIRECTORY FOR THE ROOT USER
/proc/
VIRTUAL FILESYSTEM DOCUMENTING KERNEL AND PROCESS STATUS AS TEXT FILES
/home/student/
/home/student/dir
/home/linuxgym
/usr/local
/usr/local/bin
/usr/local/games
FILESYSTEM HIERARCHY STANDARD (FHS)

File Types

- **Regular Files**
  - Regular files are ordinary files on a system that contains programs, texts, or data. It is used to store information such as text, or images.
  - These files are located in a directory/folder.
  - Regular files contain all readable files such as text files, Docx files, programming files, etc, binary files, image files such as JPG, PNG, SVG, etc, compressed files such as ZIP, RAR, etc.

Example:

```
shantanu@shantanu-Dell-System-XPS-L502X:~/Desktop/GFG$ ls -l
total 1172
drwxrwxr-x 2 shantanu shantanu    4096 Jul 18 20:41 GFG
-rw-rw-r-- 1 shantanu shantanu      15 Jul 18 20:48 GFGdocx.docx
-rw-rw-r-- 1 shantanu shantanu 1179367 Jul 18 20:46 GFGPhoto.png
-rwxrwxr-x 1 shantanu shantanu      42 Jul 18 20:55 GFGPython.py
-rw-rw-r-- 1 shantanu shantanu      68 Jul 18 20:41 GFG.rar
-rw-rw-r-- 1 shantanu shantanu      10 Jul 18 20:34 GFG.txt
-rw-rw-r-- 1 shantanu shantanu       0 Jul 18 20:58 shantanu.txt
shantanu@shantanu-Dell-System-XPS-L502X:~/Desktop/GFG$
```

- Or we can use the "***file*** *" command to find out the file type

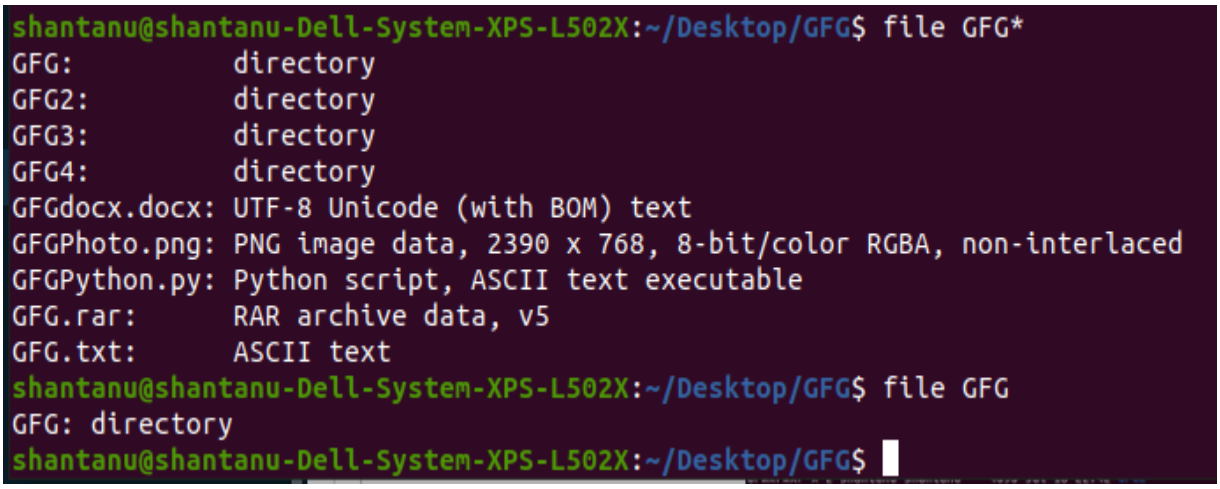
```
shantanu@shantanu-Dell-System-XPS-L502X:~/Desktop/GFG$ file *
GFG:          directory
GFGdocx.docx: UTF-8 Unicode (with BOM) text
GFGPhoto.png: PNG image data, 2390 x 768, 8-bit/color RGBA, non-interlaced
GFGPython.py: Python script, ASCII text executable
GFG.rar:      RAR archive data, v5
GFG.txt:      ASCII text
shantanu.txt: empty
shantanu@shantanu-Dell-System-XPS-L502X:~/Desktop/GFG$
```

- **Directory Files**
  - The sole job of directory files is to store the other regular files, directory files, and special files and their related information. This type of file will be denoted in blue color with links greater than or equal to 2.
  - A directory file contains an entry for every file and sub-directory that it houses. If we have 10 files in a directory, we will have 10 entries in the directory file.
  - We can navigate between directories using the ***cd*** command.
  - We can find out directory file by using the following command:

    ls -l | grep ^d

```
shantanu@shantanu-Dell-System-XPS-L502X:~/Desktop/GFG$ mkdir GFG2
shantanu@shantanu-Dell-System-XPS-L502X:~/Desktop/GFG$ mkdir GFG3
shantanu@shantanu-Dell-System-XPS-L502X:~/Desktop/GFG$ mkdir GFG4
shantanu@shantanu-Dell-System-XPS-L502X:~/Desktop/GFG$ ls -l | grep ^d
drwxrwxr-x 2 shantanu shantanu    4096 Jul 18 20:41 GFG
drwxrwxr-x 2 shantanu shantanu    4096 Jul 18 22:42 GFG2
drwxrwxr-x 2 shantanu shantanu    4096 Jul 18 22:42 GFG3
drwxrwxr-x 2 shantanu shantanu    4096 Jul 18 22:42 GFG4
shantanu@shantanu-Dell-System-XPS-L502X:~/Desktop/GFG$ ls -l
total 1184
drwxrwxr-x 2 shantanu shantanu    4096 Jul 18 20:41 GFG
drwxrwxr-x 2 shantanu shantanu    4096 Jul 18 22:42 GFG2
drwxrwxr-x 2 shantanu shantanu    4096 Jul 18 22:42 GFG3
drwxrwxr-x 2 shantanu shantanu    4096 Jul 18 22:42 GFG4
-rw-rw-r-- 1 shantanu shantanu      15 Jul 18 20:48 GFGdocx.docx
-rw-rw-r-- 1 shantanu shantanu 1179367 Jul 18 20:46 GFGPhoto.png
-rwxrwxr-x 1 shantanu shantanu      42 Jul 18 20:55 GFGPython.py
-rw-rw-r-- 1 shantanu shantanu      68 Jul 18 20:41 GFG.rar
-rw-rw-r-- 1 shantanu shantanu      10 Jul 18 20:34 GFG.txt
-rw-rw-r-- 1 shantanu shantanu       0 Jul 18 20:58 shantanu.txt
```

- We can also use the **file *** command



- **Special Files**

1. *Block Files:*
   - Block files act as a direct interface to block devices hence they are also called block devices.
   - A block device is any device that performs data Input and Output operations in units of blocks.
   - These files are hardware files and most of them are present in /dev.
   - We can find out block file by using the following command:

     ls -l | grep ^b

```
root@shantanu-Dell-System-XPS-L502X:/dev# ls -l | grep ^b
brw-rw----  1 root     disk      7,   0 Jul 18 10:47 loop0
brw-rw----  1 root     disk      7,   1 Jul 18 10:47 loop1
brw-rw----  1 root     disk      7,  10 Jul 18 11:32 loop10
brw-rw----  1 root     disk      7,   2 Jul 18 10:47 loop2
brw-rw----  1 root     disk      7,   3 Jul 18 10:47 loop3
brw-rw----  1 root     disk      7,   4 Jul 18 10:47 loop4
brw-rw----  1 root     disk      7,   5 Jul 18 10:47 loop5
brw-rw----  1 root     disk      7,   6 Jul 18 10:47 loop6
brw-rw----  1 root     disk      7,   7 Jul 18 10:48 loop7
brw-rw----  1 root     disk      7,   8 Jul 18 10:48 loop8
brw-rw----  1 root     disk      7,   9 Jul 18 10:49 loop9
brw-rw----  1 root     disk      8,   0 Jul 18 10:47 sda
brw-rw----  1 root     disk      8,   1 Jul 18 10:47 sda1
brw-rw----  1 root     disk      8,   2 Jul 18 10:47 sda2
brw-rw----  1 root     disk      8,   3 Jul 18 10:47 sda3
brw-rw----  1 root     disk      8,   4 Jul 18 10:47 sda4
brw-rw----  1 root     disk      8,   5 Jul 18 10:47 sda5
brw-rw----  1 root     disk      8,   6 Jul 18 10:47 sda6
brw-rw----+ 1 root     cdrom    11,   0 Jul 18 10:47 sr0
root@shantanu-Dell-System-XPS-L502X:/dev#
```

- We can use the **file** command also:

```
root@shantanu-Dell-System-XPS-L502X:/dev# file sda5
sda5: block special (8/5)
root@shantanu-Dell-System-XPS-L502X:/dev#
```

2. *Character device files:*
   - A character file is a hardware file that reads/writes data in character by character in a file.
   - These files provide a serial stream of input or output and provide direct access to hardware devices.
   - The terminal, serial ports, etc are examples of this type of file.
   - We can find out character device files by:

     ls -l | grep ^c

```
root@shantanu-Dell-System-XPS-L502X:/dev# ls -l | grep ^c
crw-r--r--  1 root     root     10, 235 Jul 18 10:47 autofs
crw-------  1 root     root     10, 234 Jul 18 10:45 btrfs-control
crw-------  1 root     root      5,   1 Jul 18 10:47 console
crw-------  1 root     root     10,  59 Jul 18 10:47 cpu_dma_latency
crw-------  1 root     root     10, 203 Jul 18 10:45 cuse
crw-------  1 root     root    238,   0 Jul 18 10:47 drm_dp_aux0
crw-------  1 root     root     10,  62 Jul 18 10:47 ecryptfs
crw-rw----  1 root     video    29,   0 Jul 18 10:47 fb0
crw-rw----  1 root     video    29,   1 Jul 18 10:47 fb1
crw-------  1 root     root     10,  58 Jul 18 10:47 freefall
crw-rw-rw-  1 root     root      1,   7 Jul 18 10:47 full
```

- We can use the ***file*** command to find out the type of file:

```
root@shantanu-Dell-System-XPS-L502X:/dev# file v*
v4l:          directory
vcs:          character special (7/0)
vcs1:         character special (7/1)
vcs2:         character special (7/2)
vcs3:         character special (7/3)
vcs4:         character special (7/4)
vcs5:         character special (7/5)
vcs6:         character special (7/6)
vcsa:         character special (7/128)
vcsa1:        character special (7/129)
vcsa2:        character special (7/130)
```

3. *Pipe Files:*
   - The other name for pipe is a “named” pipe, which is sometimes called a FIFO.
   - FIFO stands for “First In, First Out” and refers to the property that the order of bytes going in is the same as coming out.
   - The “name” of a named pipe is actually a file name within the file system. This file sends data from one process to another so that the receiving process reads the data first-in-first-out manner.

- We can find out pipe file by using the following command:

  ls -l | grep ^p

```
root@shantanu-Dell-System-XPS-L502X:/dev# ls -l | grep ^p
prw-r--r--  1 root     root           0 Jul 19 00:28 GFG
prw-r--r--  1 root     root           0 Jul 19 00:28 GFG1
prw-r--r--  1 root     root           0 Jul 19 00:28 GFG2
prw-r--r--  1 root     root           0 Jul 18 21:51 xyz
root@shantanu-Dell-System-XPS-L502X:/dev#
```

- We can use the *file* command to find out file type:

```
root@shantanu-Dell-System-XPS-L502X:/dev# mkfifo GFG
root@shantanu-Dell-System-XPS-L502X:/dev# mkfifo GFG1
root@shantanu-Dell-System-XPS-L502X:/dev# mkfifo GFG2
root@shantanu-Dell-System-XPS-L502X:/dev# file GFG*
GFG:  fifo (named pipe)
GFG1: fifo (named pipe)
GFG2: fifo (named pipe)
```

4. *Symbol link files:*
   - A symbol link file is a type of file in Linux which points to another file or a folder on your device. Symbol link files are also called Symlink and are similar to shortcuts in Windows.
   - We can find out Symbol link file by using the following command:

     ls -l | grep ^l

```
root@shantanu-Dell-System-XPS-L502X:/dev# ls -l | grep ^l
lrwxrwxrwx  1 root     root           3 Jul 18 10:47 cdrom -> sr0
lrwxrwxrwx  1 root     root           3 Jul 18 10:47 cdrw -> sr0
lrwxrwxrwx  1 root     root          11 Jul 18 10:45 core -> /proc/kcore
lrwxrwxrwx  1 root     root           3 Jul 18 10:47 dvd -> sr0
lrwxrwxrwx  1 root     root           3 Jul 18 10:47 dvdrw -> sr0
lrwxrwxrwx  1 root     root          13 Jul 18 10:45 fd -> /proc/self/fd
lrwxrwxrwx  1 root     root          12 Jul 18 10:45 initctl -> /run/initctl
lrwxrwxrwx  1 root     root          28 Jul 18 10:45 log -> /run/systemd/journal
/dev-log
lrwxrwxrwx  1 root     root           4 Jul 18 10:47 rtc -> rtc0
lrwxrwxrwx  1 root     root          15 Jul 18 10:45 stderr -> /proc/self/fd/2
lrwxrwxrwx  1 root     root          15 Jul 18 10:45 stdin -> /proc/self/fd/0
lrwxrwxrwx  1 root     root          15 Jul 18 10:45 stdout -> /proc/self/fd/1
root@shantanu-Dell-System-XPS-L502X:/dev#
```

- We can use the *file* command to find out file type:

```
root@shantanu-Dell-System-XPS-L502X:/dev# file core
core: symbolic link to /proc/kcore
root@shantanu-Dell-System-XPS-L502X:/dev#
```

5. *Socket Files:*
   - A socket is a special file that is used to pass information between applications and enables communication between two processes. We can create a socket file using the socket() system call. A socket file is located in /dev of the root folder or you can use the *find / -type s command to* find socket files.
   - find / -type s
   - We can find out Symbol link file by using the following command:

     ls -l | grep ^s

```
root@shantanu-Dell-System-XPS-L502X:/dev# ls -l | grep ^s
srwxr-xr-x  1 root     root           0 Jul 19 00:51 socket.sock
root@shantanu-Dell-System-XPS-L502X:/dev#
```

   - We can use the file command to find out file type:

```
root@shantanu-Dell-System-XPS-L502X:/dev# file socket.sock
socket.sock: socket
root@shantanu-Dell-System-XPS-L502X:/dev#
```

User Data Files

Data should be stored in a different partition of the file system of your OS. In Linux, personal data is stored in the/home/username folder. The User data, such as documents, photos, music, videos, eBooks, etc. are stored in Linux

System Data Files and Information

This chapter covers portable interfaces to data files, system identification functions, and the time and date functions.

- **Password File**

The UNIX System's password file, called the user database by POSIX.1, contains the following fields:

Description	struct passwd member	POSIX.1	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10
user name	char *pw_name	•	•	•	•	•
encrypted password	char *pw_passwd		•	•	•	•
numerical user ID	uid_t pw_uid	•	•	•	•	•
numerical group ID	gid_t pw_gid	•	•	•	•	•
comment field	char *pw_gecos		•	•	•	•
initial working directory	char *pw_dir	•	•	•	•	•
initial shell (user program)	char *pw_shell	•	•	•	•	•
user access class	char *pw_class		•		•	
next time to change password	time_t pw_change		•		•	
account expiration time	time_t pw_expire		•		•	

Historically, the password file has been stored in /etc/passwd and has been an ASCII file.

- root has a user ID of 0 (superuser)

- The encrypted password field contains a single character as a placeholder (x)
- Some fields can be empty
- The shell field contains the user's login shell. The default value for an empty shell field is usually /bin/sh. Another executable that prevents a user from loggin into a system:
- /dev/null
- /bin/false: exits with an unsuccessful (nonzero) status
- /bin/true: exits with a successful (zero) status
- Historically /etc/passwd had all of the user data, there was no shadow.
- However it was discovered that a dictionary attack could be done on the file, to discover passwords (if they are in the dictionary).
- Therefore it was decided to remove the passwords from /etc/passwd, the rest of the file remained, as it was used by many programs e.g. ls.
- The passwords were moved to /etc/shadow, and this file was made so that only the root can read it.
- /etc/passwd now has an x for the password field. /etc/shadow only shares the first field (the key-field / the user name). /etc/shadow has been expanded to contain other password management fields.

- **Shadow Passwords**
  - Systems store the encrypted password in another file, often called the **shadow password file**. Minimally, this file has to contain the user name and the encrypted password.

Description	struct spwd member
user login name	char *sp_namp
encrypted password	char *sp_pwdp
days since Epoch of last password change	int sp_lstchg
days until change allowed	int sp_min
days before change required	int sp_max
days warning for expiration	int sp_warn
days before account inactive	int sp_inact
days since Epoch when account expires	int sp_expire
reserved	unsigned int sp_flag

- The shadow password file should not be readable by the world. Only a few programs need to access encrypted passwords, e.g. login(1) and passwd(1), and these programs are often set-user-ID root. With shadow passwords, the regular password file, /etc/passwd, can be left readable by the world.

- **Group File**

  - The UNIX System's group file, called the group database by POSIX.1, contains the following fields:

Description	struct group member	POSIX.1	FreeBSD 8.0	Linux 3.2.0	Mac OS X 10.6.8	Solaris 10
group name	char *gr_name	•	•	•	•	•
encrypted password	char *gr_passwd		•	•	•	•
numerical group ID	int gr_gid	•	•	•	•	•
array of pointers to individual user names	char **gr_mem	•	•	•	•	•

  - The field gr_mem is an array of pointers to the user names that belong to this group. This array is terminated by a null pointer

- **Other Data Files**

- Numerous other files are used by UNIX systems in normal day-to-day operations.

- Services and networks:

- /etc/services
- /etc/protocols
- /etc/networks

- **Time and Date Routines**
  - Calendar times: number of seconds (represented in a time_t data type) that have passed: 00:00:00 January 1, 1970, Universal Time Coordinated (UTC). These calendar times represent both the time and the date. The UNIX System has always differed from other operating systems in:
  - keeping time in UTC instead of the local time
  - automatically handling conversions, such as daylight saving time
  - keeping the time and date as a single quantity

Executable File

- An executable is a file that contains a program – that is, a particular kind of file that is capable of being executed or run as a program in the computer.
- In a Disk Operating System or Windows operating system, an executable file usually has a file name extension of, bat, .com, or .exe.
- In the UNIX file system, binary executable files are generally stored in there own location.
- /bin (core binaries)
- /sbin (system binaries)
- /usr/bin (application binaries)

Files

- In Linux, everything is a file. In addition to data and executable files, Linux treats directories and even the various components of your computer as files.
- This means there are files that represent your keyboard, console, printer, CD-ROM, and even your system's RAM. These special files are called devices, and they are found in the /dev directory.

File Name Conventions

- Linux file names can be up to 256 characters long, but you really have to enjoy typing to get to that extreme.
- You can name a file pragy-publication-pvt-ltd-Mathura.y2011 if you wish, but you'll probably find that shorter names (and intelligent use of directories) will save lots of time and keystrokes in the course of a day.
- When naming files, you can use uppercase and lowercase letters, numbers, and certain special characters. It's a really good idea to stick with letters, numbers, and the dash, dot, and underscore characters to avoid trouble and confusion.
- Note: Don't use asterisks, backslashes, or question marks in Linux file names—these characters have special meaning to the shell and could cause your commands to do something quite different from what you intended. Also avoid using a dash as the first character of a file name, since most Linux commands will treat it as a switch.

- Files starting with a dot are hidden files. They behave just like any other file, except that the ls (list files) command will not display them unless you explicitly request it to do so. Your .profile file is an example of a hidden file.
- Also remember that Linux filenames are case sensitive, which can be difficult to get used to if you have a DOS background. Linux allows you to have unique files named goodstuff, GOODSTUFF, and GoodStuff in the same directory.
- It's best to always use lowercase in Linux unless you can think of a good reason to use uppercase or mixed case. If you're sharing or accessing a DOS file system with Linux, DOS will not be able to see the files that have uppercase or mixed-case file names.
- Unlike under DOS, the dot character (.) has no special meaning. You're not limited to the eight dot three (xxxxxxxx.yyy) style of naming because Linux treats the dot just like any other character; you can name a file Some.Yummy.CHEESECAKE.Recipes if you're so inclined.
- And here's another slight difference between Linux and DOS filesystems. Linux uses the forward-slash (/) in path names, and DOS uses the backslash (\).
- Note: A directory is merely a special type of file. So the rules and conventions for naming files apply also to directories.