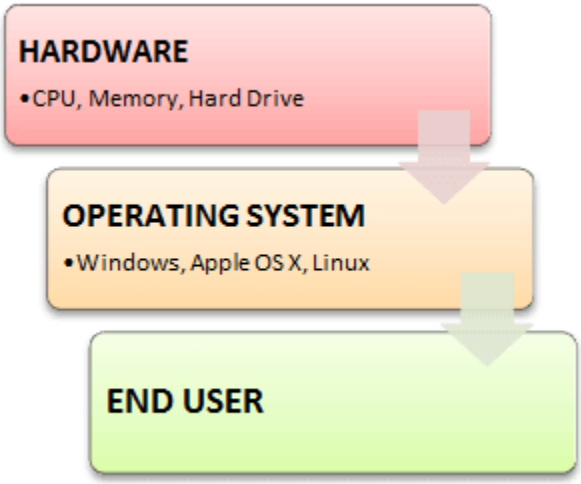
UNIX

What is an Operating System?

An operating system is a software which acts as an interface between the end user and computer hardware. Every computer must have at least one OS to run other programs. An application like Chrome, MS Word, Games, etc needs some environment in which it will run and perform its task.



History Of OS

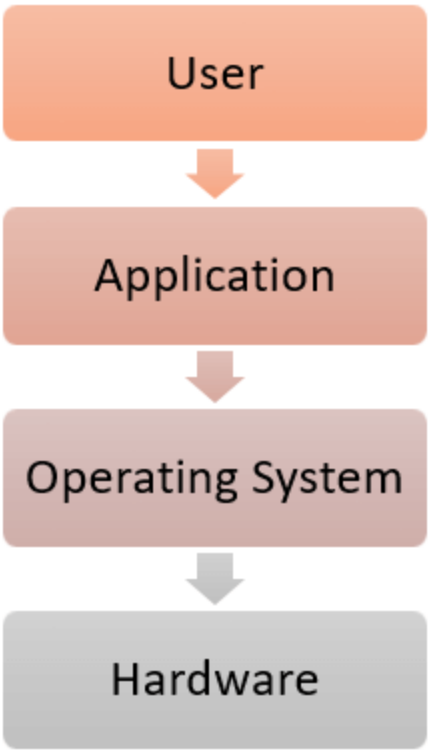
- Operating systems were first developed in the late 1950s to manage tape storage
- The General Motors Research Lab implemented the first OS in the early 1950s for their IBM 701 (Mainframe)
- In the mid-1960s, operating systems started to use disks
- In the late 1960s, the first version of the Unix OS was developed

- Several people went from SRI to Xerox PARC in the early 1970s. In 1973, Xerox PARC developed the Alto personal computer
- The first OS built by Microsoft was DOS. It was built in 1981 by purchasing the 86-DOS software from a Seattle company
- The present-day popular OS Windows first came to existence in 1985 when a GUI was created and paired with MS-DOS.

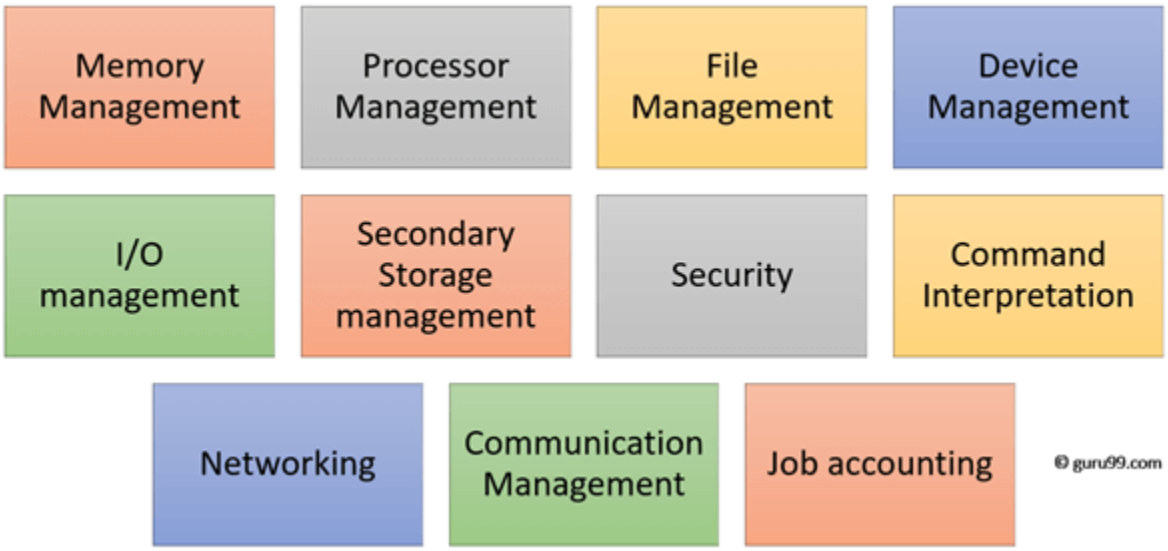
Features of Operating System

Here is a list commonly found important features of an Operating System:

- Protected and supervisor mode
- Allows disk access and file systems Device drivers Networking Security
- Program Execution
- Memory management Virtual Memory Multitasking
- Handling I/O operations
- Manipulation of the file system
- Error Detection and handling
- Resource allocation
- Information and Resource Protection



Functions of an Operating System



In an operating system software performs each of the function:

Process management:- Process management helps OS to create and delete processes. It also provides mechanisms for synchronization and communication among processes.

Memory management:- Memory management module performs the task of allocation and deallocation of memory space to programs in need of these resources.

File management:- It manages all the file-related activities such as organization storage, retrieval, naming, sharing, and protection of files.

Device Management: Device management keeps tracks of all devices. This module also responsible for this task is known as the I/O controller. It also performs the task of allocation and deallocation of the devices.

I/O System Management: One of the main objects of any OS is to hide the peculiarities of hardware devices from the user.

Secondary-Storage Management: Systems have several levels of storage which includes primary storage, secondary storage, and cache storage. Instructions and data must be stored in primary storage or cache so that a running program can reference it.

Security:- Security module protects the data and information of a computer system against malware threat and authorized access.

Command interpretation: This module is interpreting commands given by the and acting system resources to process those commands.

Networking: A distributed system is a group of processors which do not share memory, hardware devices, or a clock. The processors communicate with one another through the network.

Job accounting: Keeping track of time & resources used by various jobs and users.

Communication management: Coordination and assignment of compilers, interpreters, and another software resource of the various users of the computer systems.

=—----------

UNIX: Unix is a computer Operating System which is capable of handling activities from multiple users at the same time. The development of Unix started around 1969 at AT&T Bell Labs by **Ken Thompson and Dennis Ritchie.**

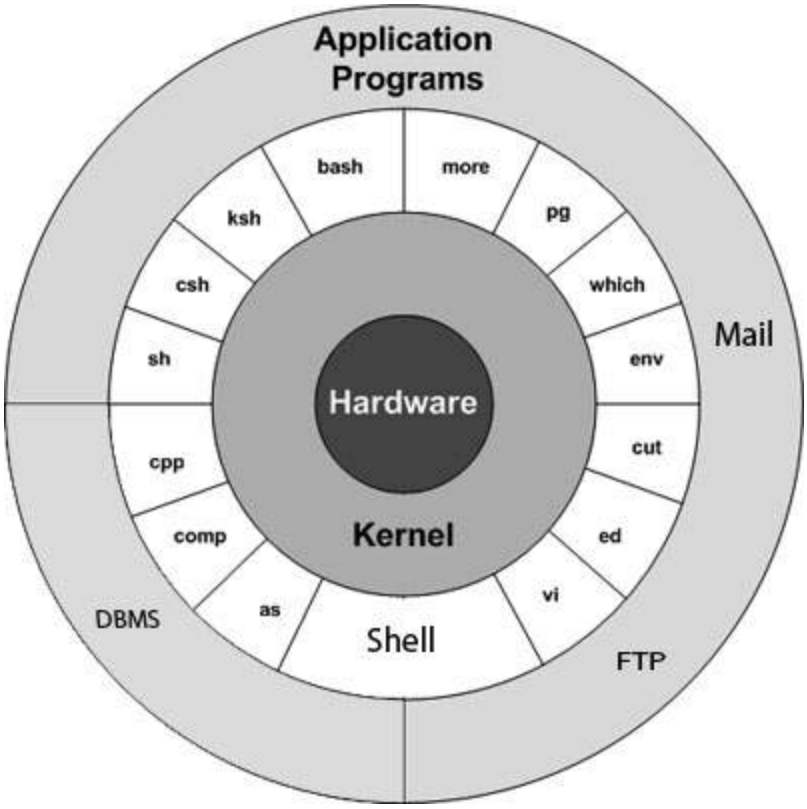
What is Unix?

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

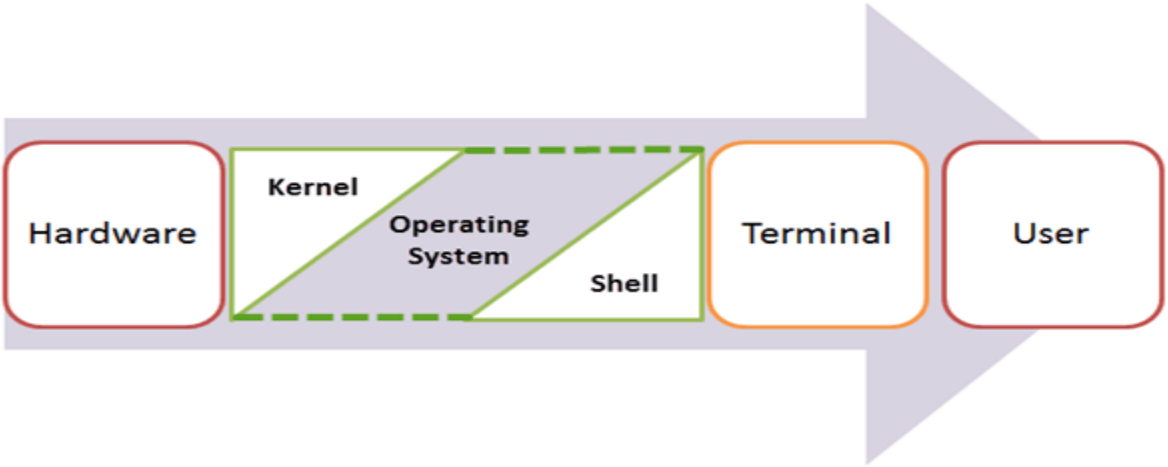
The computer programs that allocate the system resources and coordinate all the details of the computer's internals is called the operating system or the kernel.

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 Architecture?



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.
- **Shell Types**

  In Unix, there are two major types of shells −
  - Bourne shell − If you are using a Bourne-type shell, the $ character is the default prompt.
  - C shell − If you are using a C-type shell, the **%** character is the default prompt.
- The Bourne Shell has the following subcategories a−
  - Bourne shell (sh)
  - Korn shell (ksh)
  - Bourne Again shell (bash)
  - POSIX shell (sh)

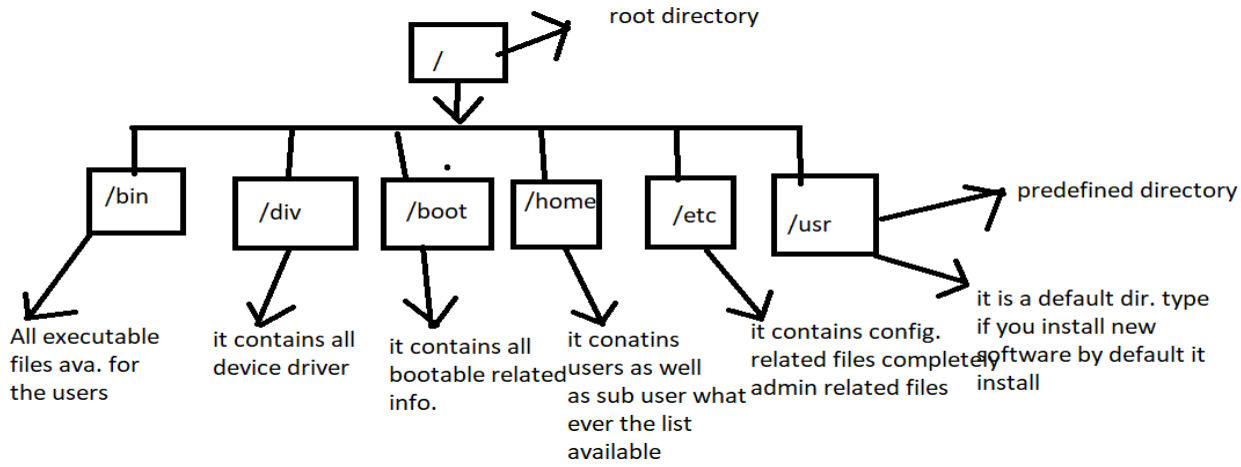
Installation of Ubuntu:

https://www.wikihow.com/Install-Ubuntu-on-VirtualBox

Intallation of VirtualBox:

https://www.virtualbox.org/wiki/Downloads

Directory Structure:



Commands:

- **PWD**-Print working dir
- **cd ~** put you in home directory

Command	Description
cd or cd ~	Navigate to HOME directory
cd ..	Move one level up
cd	To change to a particular directory

cd /	Move to the root directory

Listing Files

- **ls -l** --- lists your files in 'long format', which contains lots of useful information, e.g. the exact size of the file, who owns the file and who has the right to look at it, and when it was last modified.
- **ls** --- lists your files
- **ls -a** --- lists all files, including the ones whose filenames begin in a dot, which you do not always want to see. There are many more options, for example to list files by size, by date, recursively etc.
- **mv filename1 filename2** --- moves a file (i.e. gives it a different name, or moves it into a different directory (see below)
- **cp filename1 filename2** --- copies a file
- **rm filename** --- removes a file. It is wise to use the option rm -i, which will ask you for confirmation before actually deleting anything. You can make this your default by making an alias in your .cshrc file.
- **diff filename1 filename2** --- compares files, and shows where they differ
- **wc filename** --- tells you how many lines, words, and characters there are in a file

The History Command

History command shows all the commands that you have used in the past for the current terminal session. This can help you refer to the old commands you have entered and re-used them in your operations again.

File Permissions in Linux/Unix with Example

Linux is a clone of UNIX, the **multi-user operating system** which can be accessed by many users simultaneously. Linux can also be used in mainfraxx mes and servers without any modifications. But this raises security concerns as an unsolicited or **malign user** can **corrupt, change or remove crucial data**. For effective security, Linux divides authorization into 2 levels.

1. Ownership
2. Permission

The characters are pretty easy to remember.

r = read permission

w = write permission

x = execute permission

- = no permission

There are 2 ways to use the command -

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

Absolute(Numeric) Mode

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

Number	Octal Permission Representation	Ref
0	No permission	---
1	Execute permission	--x
2	Write permission	-w-
3	Execute and write permission: 1 (execute) + 2 (write) = 3	-wx
4	Read permission	r--
5	Read and execute permission: 4 (read) + 1 (execute) = 5	r-x
6	Read and write permission: 4 (read) + 2 (write) = 6	rw-
7	All permissions: 4 (read) + 2 (write) + 1 (execute) = 7	rwx

Symbolic Mode

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

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

The various owners are represented as -

User Denotations	
u	user/owner
g	group
o	other
a	all

Chmod o=rwx filename — others
chmod a+rwx file name —> all

Chmod g+r filename

Chmod g-r filename

Changing Ownership and Group

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

```
chown user
```

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

chown user:group file name

- **chmod options filename** --- lets you change the read, write, and execute permissions on your files. The default is that only you can look at them and change them, but you may sometimes want to change these permissions. For example, **chmod o+r filename** will make the file readable for everyone, and **chmod o-r filename** will make it unreadable for others again. Note that for someone to be able to actually look at the file the directories it is in need to be at least executable.

File Compression

- **gzip filename** --- compresses files, so that they take up much less space. Usually, text files compress to about half their original size,

but it depends very much on the size of the file and the nature of the contents. There are other tools for this purpose, too (e.g. **compress**), but gzip usually gives the highest compression rate. Gzip produces files with the ending '.gz' appended to the original filename.

- **gunzip filename** --- uncompresses files compressed by gzip

Creating & Viewing Files

The 'cat' command is used to display text files. It can also be used for copying, combining and creating new text files. Let's see how it works.

To create a new file, use the command

1. cat > filename
2. Add content
3. Press 'ctrl + d' to return to command prompt.

To view a file, use the command -

```
cat filename
```

The syntax to combine 2 files is -

cat file1 file2 > newfilename

Pipe:

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

one like a pipeline. The symbol '|' denotes a pipe.

cat filename | less

```
cat Filename | pg
```

or

```
cat Filename | more
```

cat sample | grep -v a | sort - r

grep command in Unix/Linux

The grep filter searches a file for a particular pattern of characters, and displays all lines that contain that pattern. The pattern that is searched in the file is referred to as the regular expression (grep stands for globally search for regular expression and print out).

Syntax:

grep [options] pattern [files]

Options Description

-c : This prints only a count of the lines that match a pattern

-h : Display the matched lines, but do not display the filenames.

-i : Ignores, case for matching

-l : Displays list of filenames only.

-n : Display the matched lines and their line numbers.

-v : This prints out all the lines that do not matches the pattern
-e exp : Specifies expression with this option. Can use multiple times.
-f file : Takes patterns from file, one per line.
-E : Treats pattern as an extended regular expression (ERE)
-w : Match whole word
-o : Print only the matched parts of a matching line,
with each such part on a separate output line.

Cat sample | grep apple

The 'sort' command

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

The syntax for this command is:

```
sort Filename
```

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

EX: Sort Sample
Filter: **We want to highlight** only the lines that do not contain the character 'a', but the result should be in reverse order.
cat sample | grep -v a | sort - r

Environment variables

Let's study some common environment variables -

Variable	Description
PATH	This variable contains a colon (:)-separated list of directories in which your system looks for executable files. When you enter a command on terminal, the shell looks for the command in different directories mentioned in the $PATH variable. If the command is found, it executes. Otherwise, it returns with an error 'command not found'.
USER	The username
HOME	Default path to the user's home directory
EDITOR	Path to the program which edits the content of files
UID	User's unique ID
TERM	Default terminal emulator

SHELL	Shell being used by the user

1. How do you connect to the Unix Server? Can you tell me another way to connect to the Unix server.
  ssh hostname
  telnet hostname
ssh remotename@remotehost.com

ipconfig getifaddr en1——> get ip address in Mac
ifconfig -a.—>get IP address In Linux

2. How to copy files from one server to another server in Unix.

Copy a single file from the local machine to a remote machine:

The *scp* command needs a source and destination to copy files from one location to another location. This is the pattern that we use:

```
scp local machine/path_to_the_file username@server_ip:/path_to_remote_directory

scp /Volumes/MacDrive/Distros/fedora.iso swapnil@10.0.0.75:/media/prim_5/media_server/
```

In the following example I am copying a local file from my macOS system to my Linux server (Mac OS, being a UNIX operating system, has native support for all UNIX/Linux tools).

Copy a local directory to a remote server:
If you want to copy the entire local directory to the server, then you can add the *-r* flag to the command:

```
scp -r localmachine/path_to_the_directory username@server_ip:/path_to_remote_directory/
```

3. How to check how much memory is available and how much is used in the server.

top
htop
Free

4. How to check that the Unix server is up and running.

ping

5. How many days has the Unix server is up and running.

Uptime

6. How to use top and htop command in Unix. Can tell me the difference.

- It has a nicer text-graphics interface, with colored output.
- It is easy to use and highly configurable.
- Allows for scrolling process list vertically and horizontally to see all processes and complete command lines.
- It also displays a process tree and comes with mouse support.

- Allows you to easily perform certain functions related to processes (killing, renicing etc) which can be done without entering their PIDs.
- Htop is also much faster than top

7. What is the difference between compare and diff commands.
diff file1 file2
cmp file1 file2

cmp will check byte by byte

8. How to check logs in Unix

Logs : **Log** files are a set of records that **Linux** maintains for the administrators to keep track of important events. They contain messages about the server, including the kernel, services and applications running on it. **Linux** provides a centralized repository of **log** files that can be located under the /var/**log** directory

—> cat /var/log/syslog
—> head syslog. Or head -n 50 /var/log/syslog
—> tail syslog.
—> tail -f /var/log/syslog. — real time logs
—> dmesg