Operating system: OS is a software that takes control of your machine when it is turned onn.

Whatever we are doing either using apps, or coding running we need to reach to the operating system.

It is like an interface between applications and the hardware.

Reason we need operating system:

1. Abstraction: let say if we are doing word processing without OS we also need to write code for moving the mouse curser, drives to save it to memory, display something on screen.

So OS provides basic functionality so that all these comes on system call.

1. Resource management: So we have limited resource int the machine and we r doing multiple processes like running crome, browser, firefox, word processor at the same time.

We need a manger to decide which process is going to the CPU we have limited ROM and multiple processes.

We need a manager to decide which process comes in which need to come out, according to the priorities of the process.

1. Protection: since we have multiple application, we want to ensure that one application did not impact the data on the other application.

Protection from other applications.

Protection from malware.

Example of OS:

Desktop: windows, linux, MacOS.

Mobile: Android, IOS.

There exist many OS for router, printer etc.

Types of OS:

1. According to the functionality that OS is providing:
2. Single tasking : These OS are basic OS they provide one task at one time.

The allow only one process to be these in the RAM and run at a time.

Example: MS DOS.

THE CPU ONLY TAKES TO THE PROGRAM WHICH ARE THERE IN THE MEMORY SO TO PROCESS SOMETHING IT SHOULD BE BROUGHT TO THE MEMORY.

In single tasking operating system only one process other than OS will exist in the RAM.

These are inefficient system.

1. Multiprogramming and multitasking OS: the idea is to use single CPU to run multiple process running concurrently.

So this type of OS allow multiple process in the memory and assign them smartly.

Multiprogramming: it is an idea of allowing multiple process to be there in the ram and run in the interleaved manner.

Multitasking is an extension of multiprogramming where we run the process in time slots.

1. Multithreading:

Threads : it is a sequence flow of task within a process.

In multithreading we have multiple thread working in the interleaved manner.

Advantage:

more responsive

example MS word: one thread is automatically formatting the content,

one thread which is taking input

that makes ms word more responsive.

1. Multiprocessing: till now we have only one processor now with multiprocessor probably it divides multiple process on multiple process or multiple thread to multiple processors.

Now a days we have dual core, quad core, octa core processors.

1. Multiuser: earlier windows system allows only one user to use the os at a time.

And unique system which are mainly for server could allow multiple consols to log in into the server os and use the system symultanously.

Multithreading and Multitasking difference:

Multitasking: listen to music and browsing web.

Multithreading: downloading something in browser and browsing.

Multithreading real world example:

1. Word processor: typing, saving, formatting, spelling check happens at the same time.
2. Web server: apache http server uses thread pool.
3. IDEs: modern IDEs do compiling errors check while you are writing.
4. Games: in modern games multiple objects are implemented as different theads.

In one game multiplayer different moves at the same time.

Program : Is something that is inside the hard disk, it may be anything like codes etc and then the compiler converts the high level code to binary code which is still in the hard disk. This program is then loaded to the ram and run by the processer.

Process : process is a program that is in execution.

Process-> it looks like

stack ->which control function call

heap-> which hold dynamic memory

text/ code ->the code or the binary code to be executed

data ->static and global variables