

Operating systems and Linux Course

Lecture 1:

- Topics to cover in this course:
 - Basic information and discussion on history and evolution of operating systems.
 - Linux and playing around with the linux shell.
 - Programmer development tools. (Mostly C development)

Lecture 2:

- Who needs OS?
 - Do all computers require OS?
 - No. Examples - Atari 2600, ZX Spectrum, (Both 8 bit machines), Texas instruments calculator. Processors execute instructions from the read-only memory (ROM) or the game cartridge(For Atari) sequentially in these devices and programmers have unrestricted access to all the memory addresses. These devices have processors, memory, I/O, but no OS managing these things.
 - Commodore 64 —> Complexity Increases
 - Complexity increases as third party external drives, floppy disks can be now connected and disconnected from the device.
 - Interface and basic language interpreter.
 - Now we need something to help us organize and manage these additional features.
 - We need something to help us make sense of the things that are happening with the processor.
 - Out of the box -> the commodore 64 did not have any OS. Option to install other unix like systems or systems from other vendors.
- Why need an OS?
 - Need some sort of a system to organize, manage and help make sense of different tasks, processes, and users. It could also be used to restrict access to certain parts of memory which could mess up our entire system's execution.
 - It all boils down to this :-
 - **No computer requires an operating system. People need OSs to make sense of computers.**
- OS Levels and Layers of abstraction:
 - Abstraction - We don't have to worry about the lower level details of how it is working. Example - file saving would be so much pain if we had to figure out the lower level details of how to be able to save it on a disk.
 - Thus users created this layer of abstraction called OS which helps us manage things on a lower level without having to worry about it all.
 - Components - We divide or split the computer system into smaller components to make things easier to understand.
 - Layers/Levels - These are classification of a component according to where it sits between the hardware and the user. The gap in between these two is where the OS operates.
 - What is the OS responsible for?

- Levels and Layers: -
 - User Processes: Shell/CLI, GUI, Web Server, Database Server, applications(all kinds)
 - Kernel: Memory management, Process management, Device drivers, system calls.
 - Hardware: Processor(CPU), Main memory(RAM), Disks, Network ports.
- Kernel is the heart of an operating system. It decides how we manage memory, how certain processes or tasks are allocated a certain amount of memory.
- One of the jobs that OS have is making sure that failure of some process or application should not affect other running processes/applications.
- Different kind of OSs approach these tasks differently.

History and OS Evolution:

MainFrames:

> First mainframes did not have any OS

First formal OS -> General motors -> a vendor as in terms of one OS for one machine

> IBM came on later

Minicomputer

> Digital Equipment Corporation

Coming up next week:

- History of OSs
 - Mainframe
 - Unix distributions
 - Evolution of UNIX
 - Birth of Linux