Lecture 1

What is An Operating System?

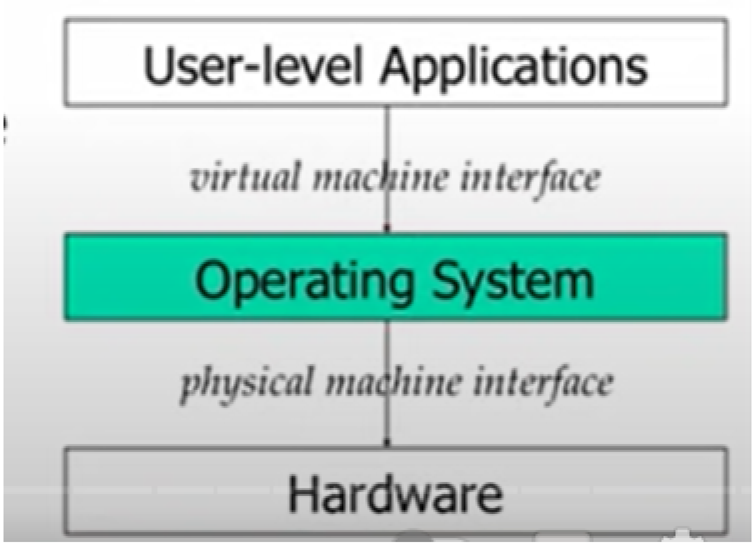
A very basic definition of operating system is that it is the interface between the user and the architecture.

1. Operating System is an Interface between user and architecture (hardware) abstracting the hardware and it divides resources between tasks
2. An operating system is a piece of software that manages all the resources of a computer system, both hardware and software, and provides an environment in which the user can execute his/her programs in a convenient and efficient manner by hiding underlying complexity of the hardware and acting as a resource manager.
3. implements a virtual machine that is easier to program than raw hardware.

kernel basically refers to the core of the operating systems.

What is the primary role of an operating system?

- interface
- abstracts hardware
- resource manager between tasks



This interferes poses an overhead which similar to the taxation system in the government, more the features of the operating system higher the overhead. Idea of trade-off in system design.

New Developments in OS:

- New applications spaces
- rapidly evolving hardware
- open-source OS (Linux)

Key Features of OS

- Services: OS provides standard services(interfere) which hardware implements. Examples: File systems, virtual memory, networking, CPU scheduling and time-sharing.
- Coordination: coordinates multiple applications and users to achieve fairness and efficiency. Example: Concurrency, memory protection, networking, and security.
  Hence OS should make the machine convenient to use and efficient.

Some Trade-Offs:

- performance and convenience of OS
- performance and the simplicity of OS
- putting functionality in hardware or software

History Of Operating Systems

1. Single-User Computers: one user at a time and computation and i/o couldn't be performed simultaneously . it need a person on console interacting with the program as it ran
2. Batch-Processing: we gave it programs on punch card we could feed it and it would sequentially execute it. Human schedules the tasks. Here too i/o and computation couldn't be performed simultaneously
3. Overlap and computation: This allowed the CPU to execute some other program while waiting for I/O of one program
4. Multiple programming: Several programs can run simultaneously .schedules tasks, protects memory of each tasks and is what we use today