MTL458: Operating Systems

Instructor: Dalu Jacob
Office: 527A, 99B
dalujacob@iitd.ac.in

TAs: Saket Kandoi
Priyal Jain
Anshika Prajapati

Evaluation: 60% → Minor + Major (25+35)
30% → 3 Assignments
10% → Best 1 out of 2 Quizzes.

No re-quizzes!

Passing criteria: 30%

Audit pass criteria: 35%, out of which 20% has to come from Major + minor.

Book: Operating Systems: Three easy pieces, Remzi H. Arpaci-Dusseau, Andrea C. Arpaci-Dusseau

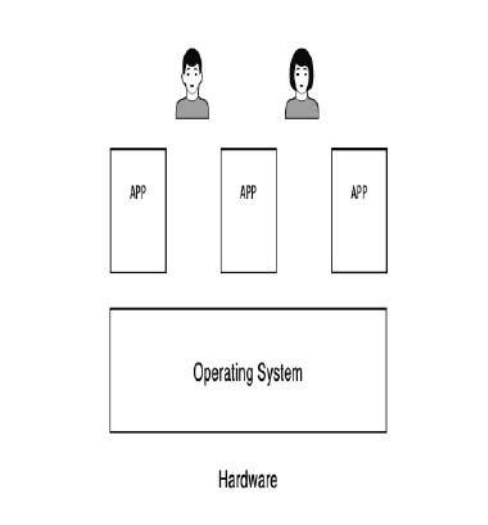
Operating Systems: Principles and practice - Thomas Anderson, Michael Dahlin.

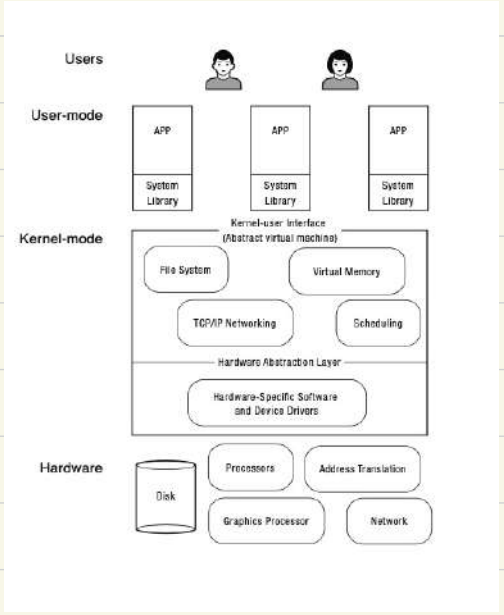
what is an operating system?

Manager ??

or an

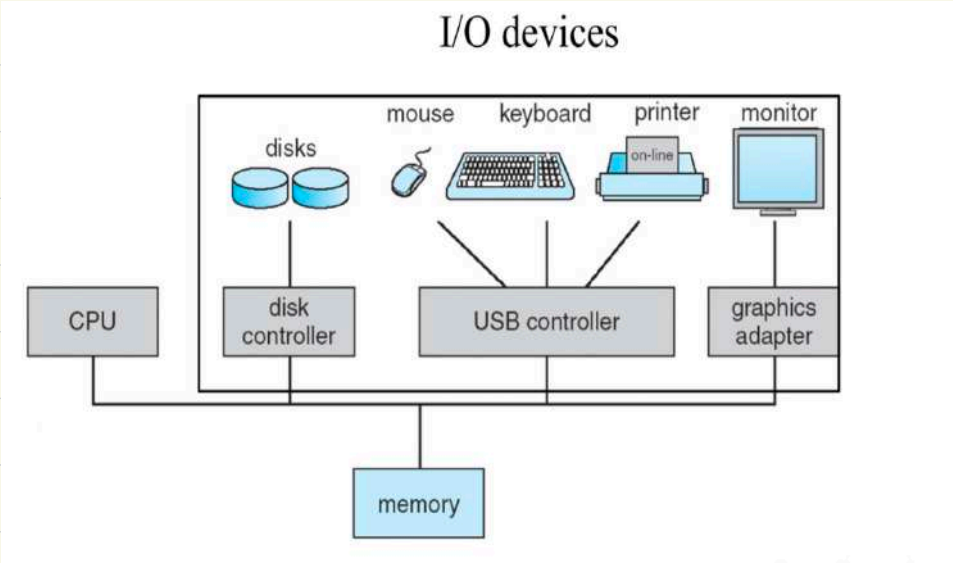
Illusionist ??





Operating system is a software (set of programs + a library of functions) to manage a computers resources for its users and applications.

What are the things to manage?



* CPU
* Memory
* Hard disk (file system)
* I/O devices

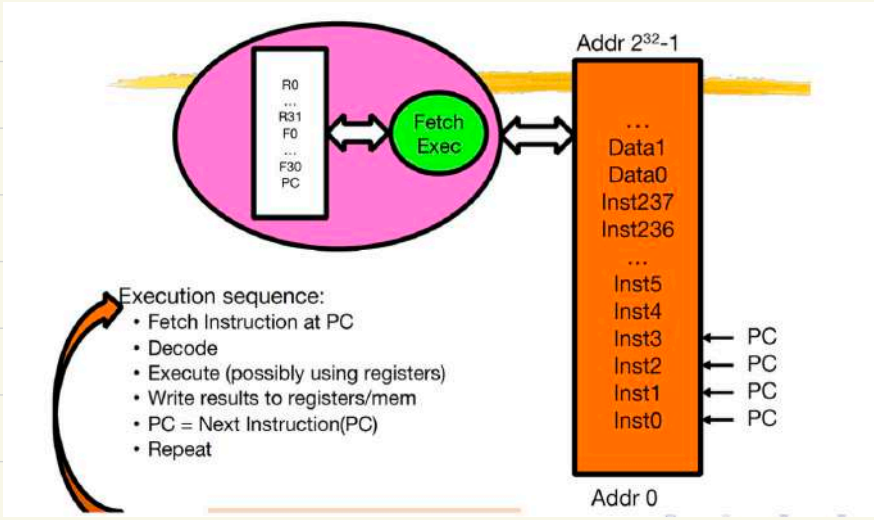
What happens when a program runs?

* The processor fetches an instruction from memory

* decodes it

- a compiler translates high level programs into an executable ("*.c" to "a.out")

- The exe contains instructions that CPU can understand, and data of the program (numbered with addresses)



* executes it

- instructions run on CPU : hardware implements an instruction set Architecture (ISA)
- CPU also consists of a few registers
  - eg:- a pointer to current instruction (program counter or PC)
  - operands of instructions, memory addresses, etc..

* update the registers, and repeat.

Fun stuffs that you can do while running a program?

Multitasking like a Pro
- run multiple programs side by side - like, do coding, listen to spotify, browse stack overflow, all at once!
- allowing programs to share memory
- enabling programs to interact with devices etc..

We assume Von Neumann model of computing
→ instructions is executed one at a time!

OS as a manager: As a resource manager, OS efficiently allocates and manages system resources among multiple programs.

Responsibilities:

* CPU scheduling: Deciding which process gets the CPU and when
* memory management: Allocating and freeing up memory for processes
* File system management: managing how data is stored and retrieved
* I/o management: Handling I/o devices like keyboard, mouse, disk
* Process management: creating scheduling, and terminating process

OS is like a factory manager assigning machines (CPU), workbenches (memory), and storage (disk) to different workers (programs) to ensure smooth operations.

OS as an illusionist : it hides the messy complexity of the hardware and gives the users the illusion of simplicity and continuity.

Illusions created by OS

* Virtual memory: Illusion of a large, private memory space for each process.
* multitasking: Illusion that multiple programs are running simultaneously
* Uniform interface: Illusion that hardware (printers/disks) work the same
* Infinite resources : Illusion that each user has full access to CPU, memory, etc..

OS is like a magician making you believe you have your own machine, even though you're sharing it with many others.

Together, the OS is both a manager that enforces control, and an illusionist that provides abstraction. This dual role makes modern computing efficient and user friendly.

Different kinds of operating system

Microsoft windows (non-Unix)

- most popular desktop operating system
- typically pre-installed by PC manufactures.

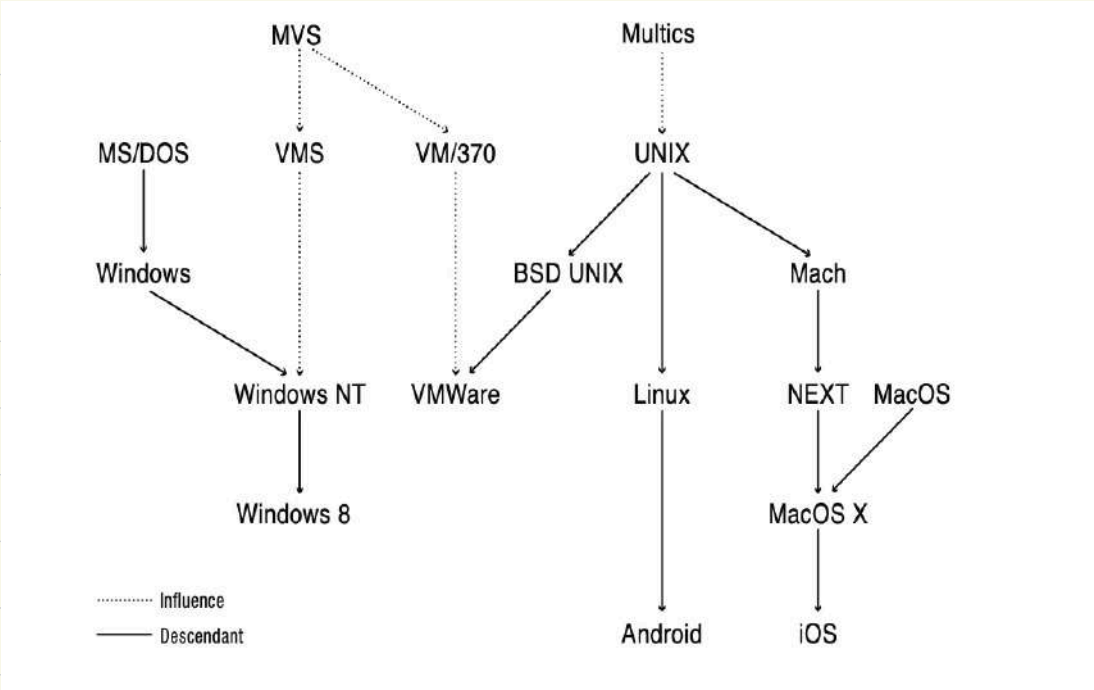
Unix systems

- Linux
- Mac OS X
- BSD
- solaris / open solaris
- commercial Unix.

Mainframe systems

Embedded systems

- Embedded Linux → Android
- Blackberry



Evolution of OS : Early operating systems

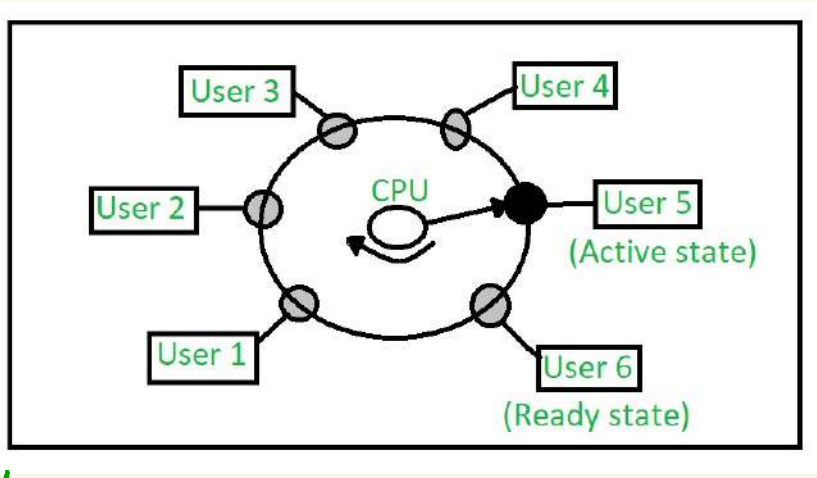
* One application at a time
  - Had complete control of hardware
  - OS was runtime library
  - users would stand in line to use computer

* Batch systems

- Keep CPU busy by having a queue of jobs
- Os would load next job while current one runs
- Users would submit jobs, and wait, and wait, and, ...

Time sharing operating systems

- multiple users on the computer at the same time
- multiprogramming
- Interactive performance: try to complete everyone's task quickly
- more importance to optimize user time, not computer time.

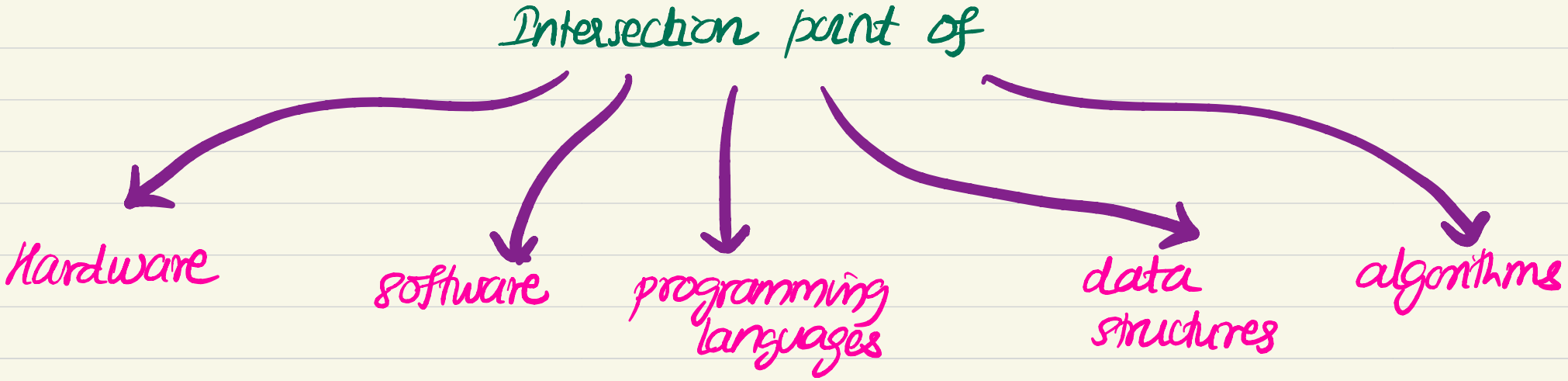


Today's operating systems

* smart phones
* Embedded systems
* Laptops
* Tablets
* Virtual machines
* Data center services

Refer to section 2.6 of Chapter-2 to learn more about history.

Why should we study OS?



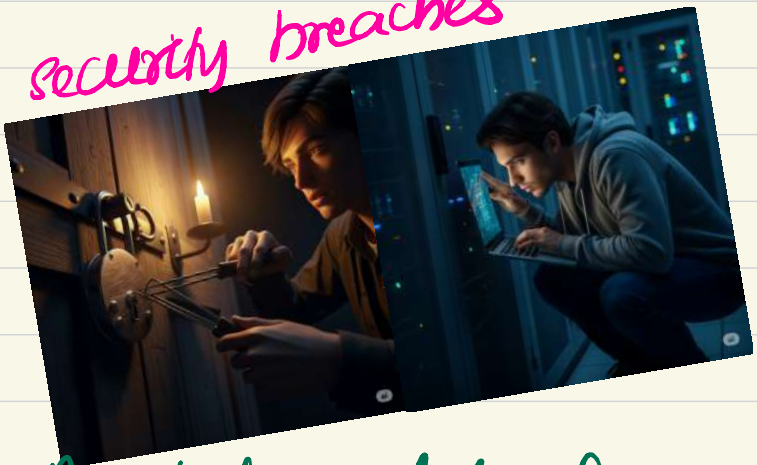
curiosity → "look under the hood"

Increasing need for specialized operating systems

- embedded OS for devices → cell phones, sensors, controllers,
- real time OS → aircraft controler, multimedia, etc...

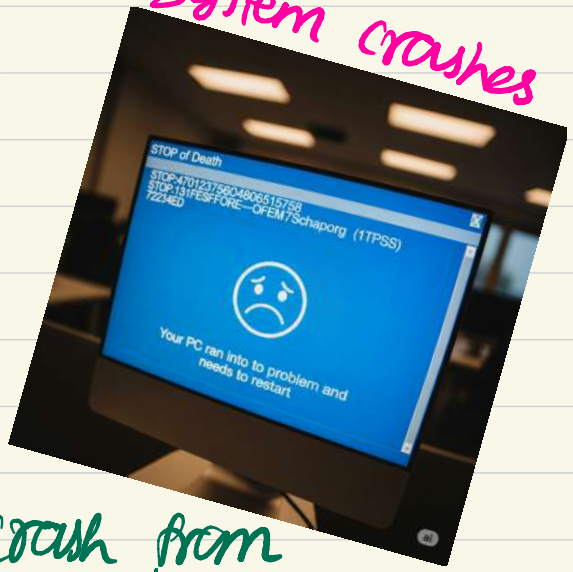
A world without an operating system!

Security breaches



App stealing data from another

System crashes



Crash from buggy apps

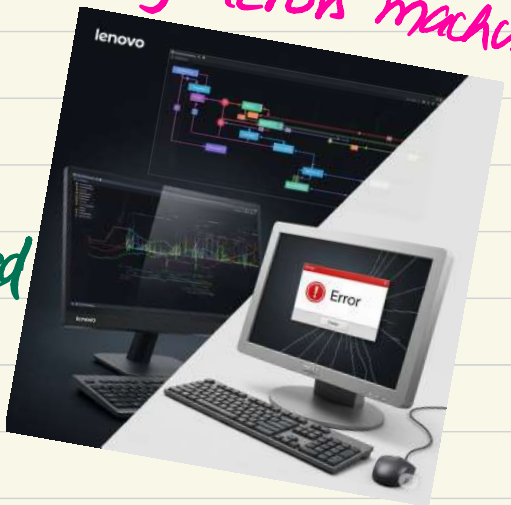
No multi-user support

Shared access without protection

Wasted resources

Resource hog slowing others

Incompatibility across machines



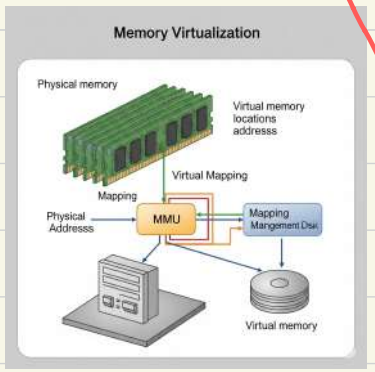
Hardware-tied software

what are the three pieces???

Virtualization

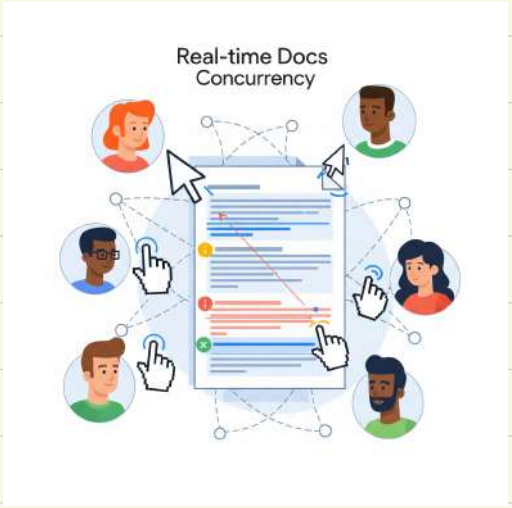
processor

memory



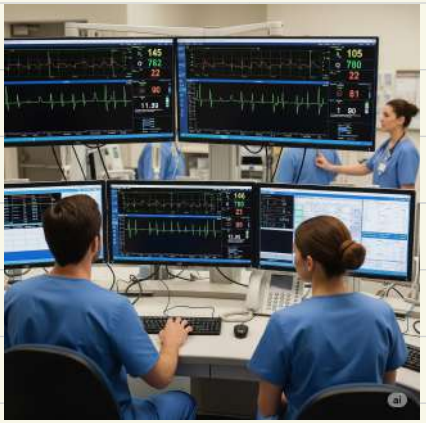
* OS takes a physical resource (like processor, memory, disk etc.) and transforms it into an easy-to-use virtual form of itself, creating an illusion that each process has its own computer.

Concurrency



* edits are synced in real-time
* no one overwrites others' work - concurrent access is managed.
- process and thread management

Persistance



* ensures your work, settings, and system data survive power-offs, crashes, or re-starts.
- crucial in safety-critical systems like health care, finance etc..

---

further reading: chapter 2, OSTEP