

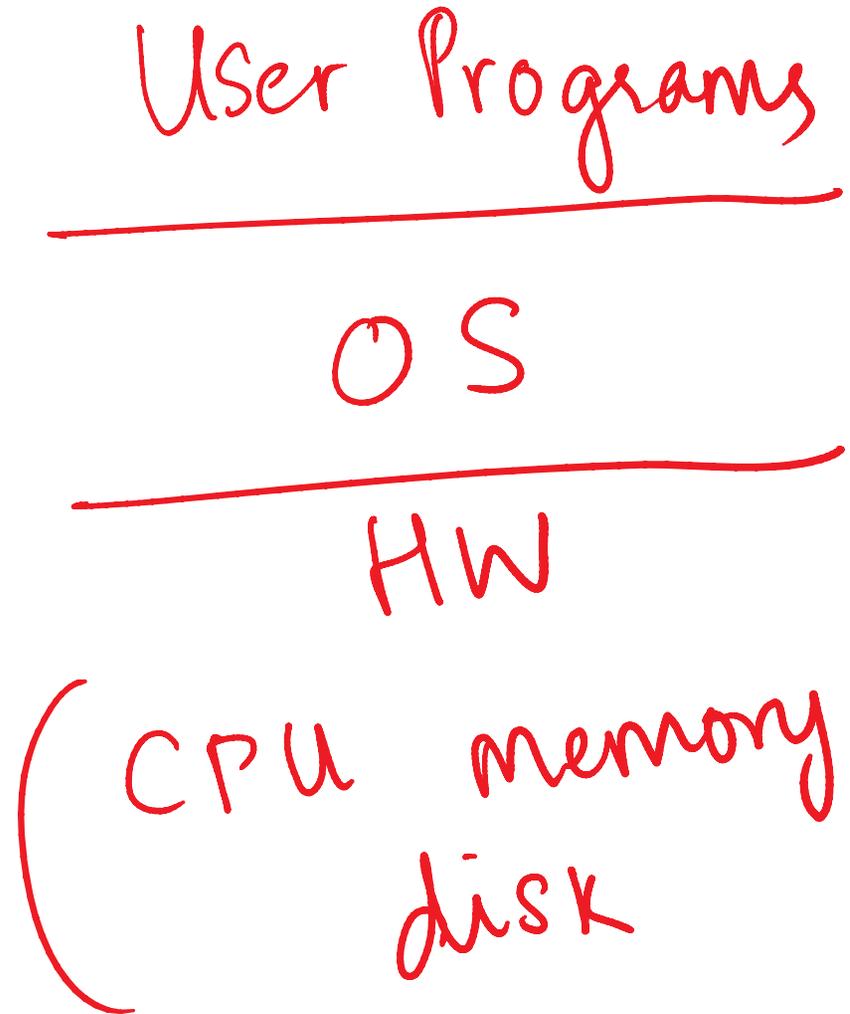
Lecture 1: Introduction to Operating Systems

Mythili Vutukuru

IIT Bombay

What is an operating system?

- Middleware between user programs and system hardware
- Manages hardware: CPU, main memory, IO devices (disk, network card, mouse, keyboard etc.)

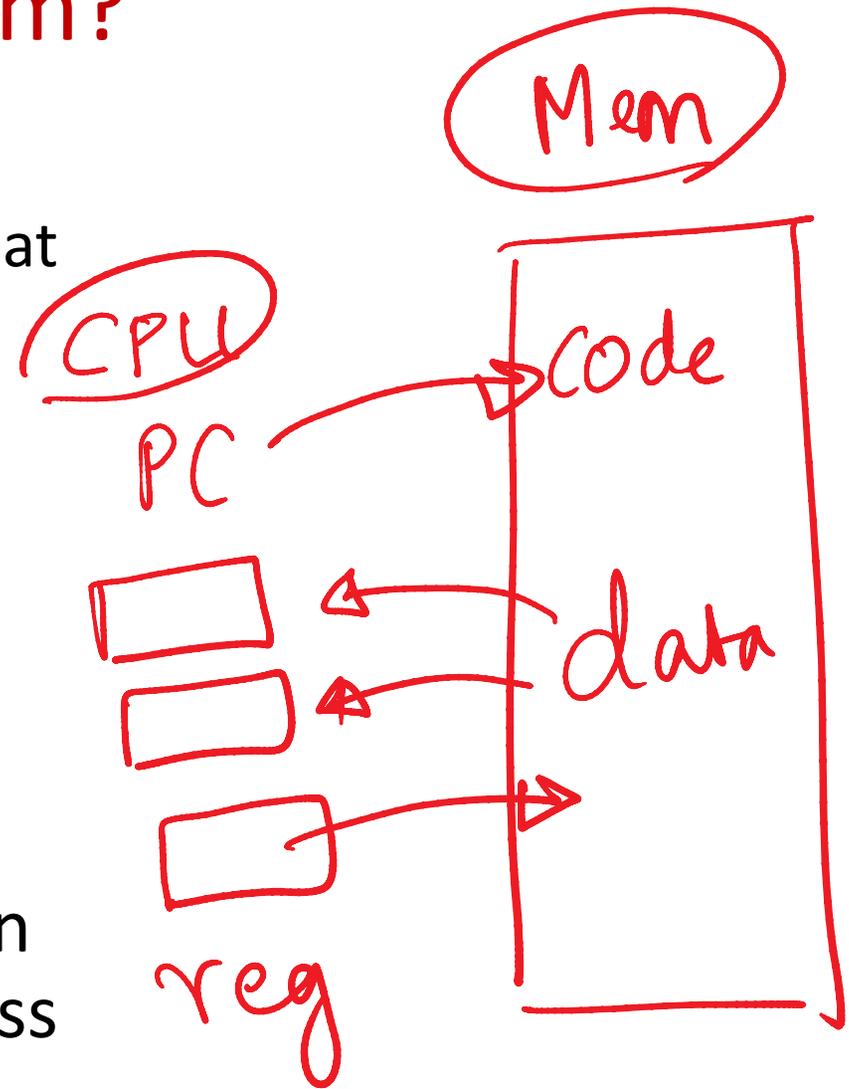


What happens when you run a program? (Background)

- A compiler translates high level programs into an executable (".c" to "a.out")
- The exe contains instructions that the CPU can understand, and data of the program (all numbered with addresses)
- Instructions run on CPU: hardware implements an instruction set architecture (ISA)
- CPU also consists of a few registers, e.g.,
 - Pointer to current instruction (program counter or PC)
 - Operands of instructions, memory addresses

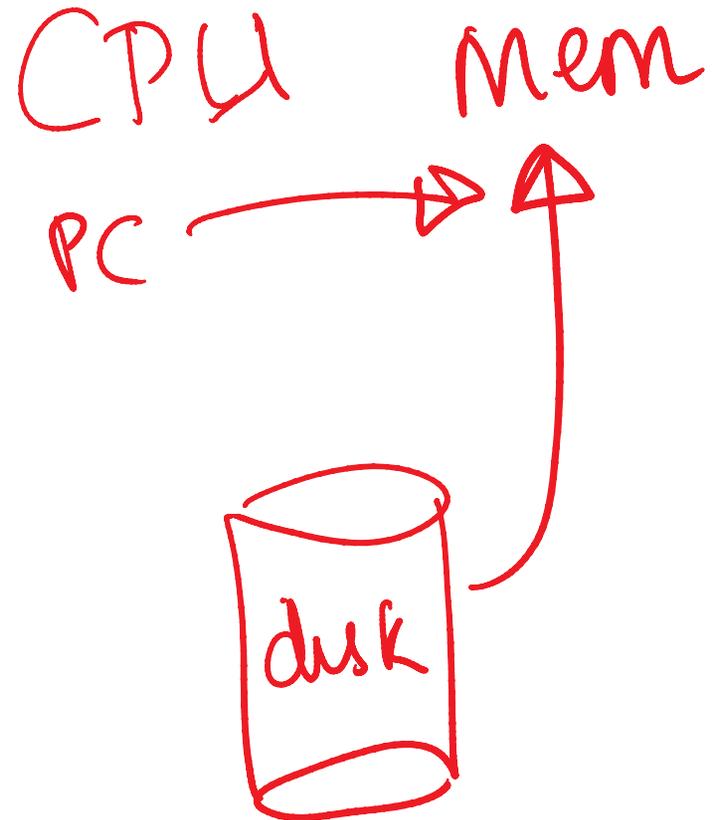
So, what happens when you run a program?

- To run an exe, CPU
 - fetches instruction pointed at by PC from memory
 - loads data required by the instructions into registers
 - decodes and executes the instruction
 - stores results to memory
- Most recently used instructions and data are in CPU caches for faster access



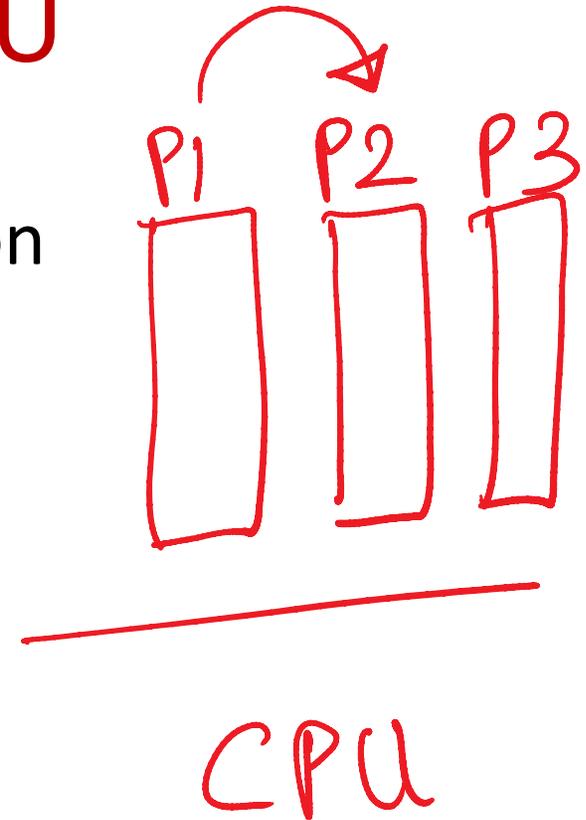
So, what does the OS do?

- OS manages program memory
 - Loads program executable (code, data) from disk to memory
- OS manages CPU
 - Initializes program counter (PC) and other registers to begin execution
- OS manages external devices
 - Read/write files from disk.



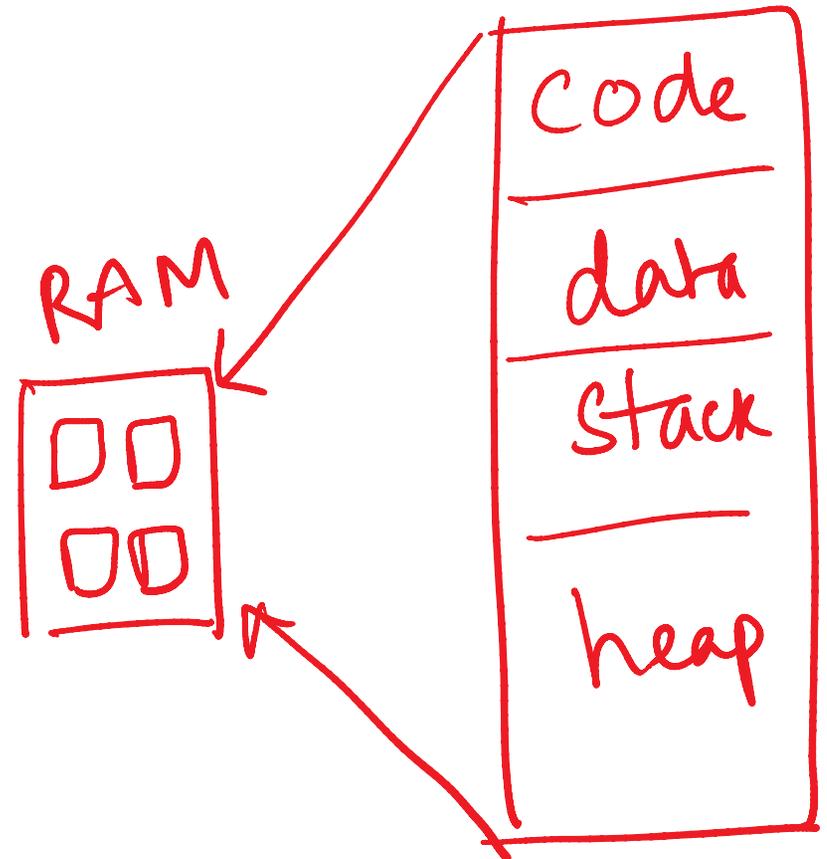
OS manages CPU

- OS provides the process abstraction
 - Process: a running program
 - OS creates and manages processes
- Each process has the illusion of having the complete CPU, i.e., OS virtualizes CPU
- Timeshares CPU between processes
- Enables coordination between processes



OS manages memory

- OS manages the memory of the process: code, data, stack, heap etc
- Each process thinks it has a dedicated memory space for itself, numbers code and data starting from 0 (virtual addresses)
- OS abstracts out the details of the actual placement in memory, translates from virtual addresses to actual physical addresses



OS manages devices

- OS has code to manage disk, network card, and other external devices: device drivers
- Device driver talks the language of the hardware devices
 - Issues instructions to devices (fetch data from a file)
 - Responds to interrupt events from devices (user has pressed a key on keyboard)
- Persistent data organized as a filesystem on disk

Design goals of an operating system

- Convenience, abstraction of hardware resources for user programs
- Efficiency of usage of CPU, memory, etc.
- Isolation between multiple processes

History of operating systems

- Started out as a library to provide common functionality across programs
- Later, evolved from procedure call to system call: what's the difference?
- When a system call is made to run OS code, the CPU executes at a higher privilege level
- Evolved from running a single program to multiple processes concurrently