Operating system concepts

Types of OS

By Prof K R Chowdhary

JNV University

2023

Recommended Reading

- *Operating System concepts*, by Abraham Silerschatz, Peter Baer Galvin, and Greg Gagne (Important: Available online, pdf free downloadable)
- *Operating Systems*, by Stuart Madnick, John Donovan, McGraw Hill Education, 2017
- *Operating System Concepts and Design*, Milan Milenkovoc, Mcgraw-Hill.

Types of operating systems

1. Batch processing operating systems
2. Multiprogramming operating systems
3. Time sharing operating systems
4. Real-time operating system
5. Distributed operating System
6. Combination operating systems

Punch card machine

Figure 1: A punch card

Batch System

- Use of tape drives allow batching of jobs:
  - programmers put jobs on cards as before.
  - all cards read onto a tape.
  - operator carries input tape to computer.
  - results written to output tape.
  - output tape taken to printer.
- Computer now has a resident monitor :
  - initially control is in monitor.
  - monitor reads job and transfer control.
  - at end of job, control transfers back to monitor.
- Even better: spooling systems.
  - use interrupt driven I/O.
  - use magnetic disk to cache input tape.
  - file operation.
- Monitor now schedules jobs. . .

Multi-programming

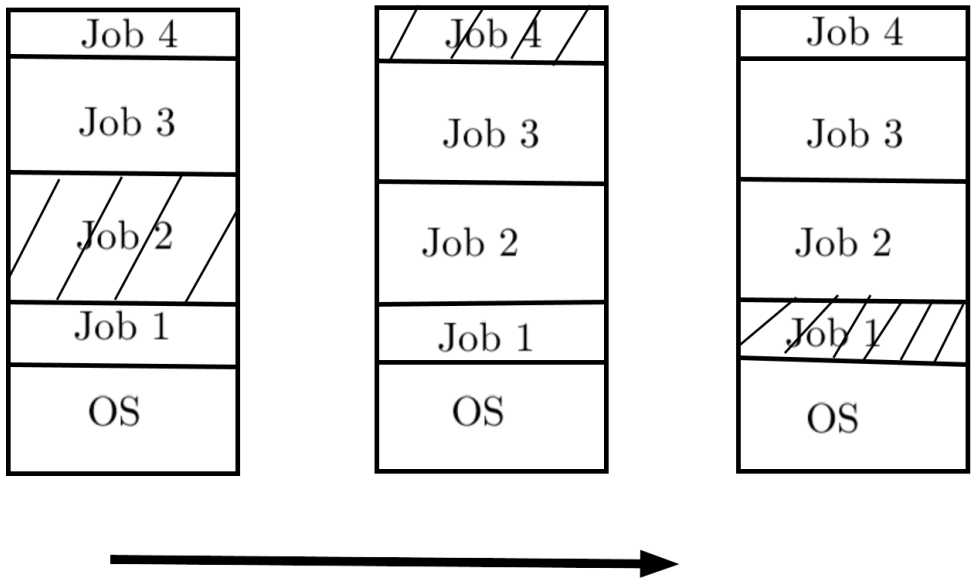

Figure 2: Multi-programming

Multi-programming....

- Use memory to cache jobs from disk $\Rightarrow$ more than one job active simultaneously.
- Two stage scheduling:
  1. select jobs to load: job scheduling.
  2. select resident job to run: CPU scheduling.
- Users want more interaction $\Rightarrow$ time-sharing:
- e.g. CTSS (compatible time-sharing system), TSO (time sharing options), Unix, VMS, Windows NT. . .
- DOS?

Real-time Operating Systems (RTOS)

Time interval required to process and respond to inputs is small

- Small *response time*
- They used when there are rigid time requirements
- Examples: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.
- Types of RTOS:
  - Hard real-time systems
  - Soft real-time systems : multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers, etc

Multi-process Operating Systems

- These are used to boost the performance that uses multiple porcesses in a single computer system.
- Multiple CPUs are linked together so that a job can be divided and executed more quickly.
- There may or may not be multiple CPUs
- Programs run as processes in round-robin fashion, like a grand master chess player plays with many learner chess players playing on different chess boards.

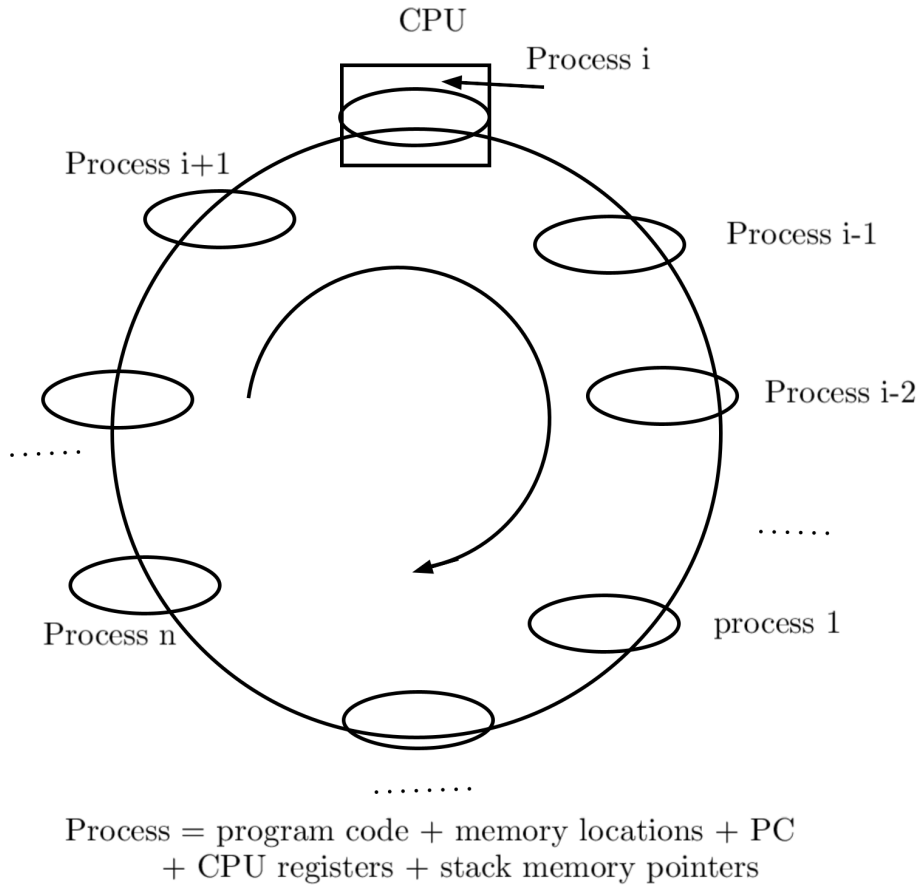

Figure 3: Multi-processing OS

Time-sharing OS

An operating system design that allows multiple users or processes to concurrently share the same system resources, such as the CPU, memory, and peripherals.

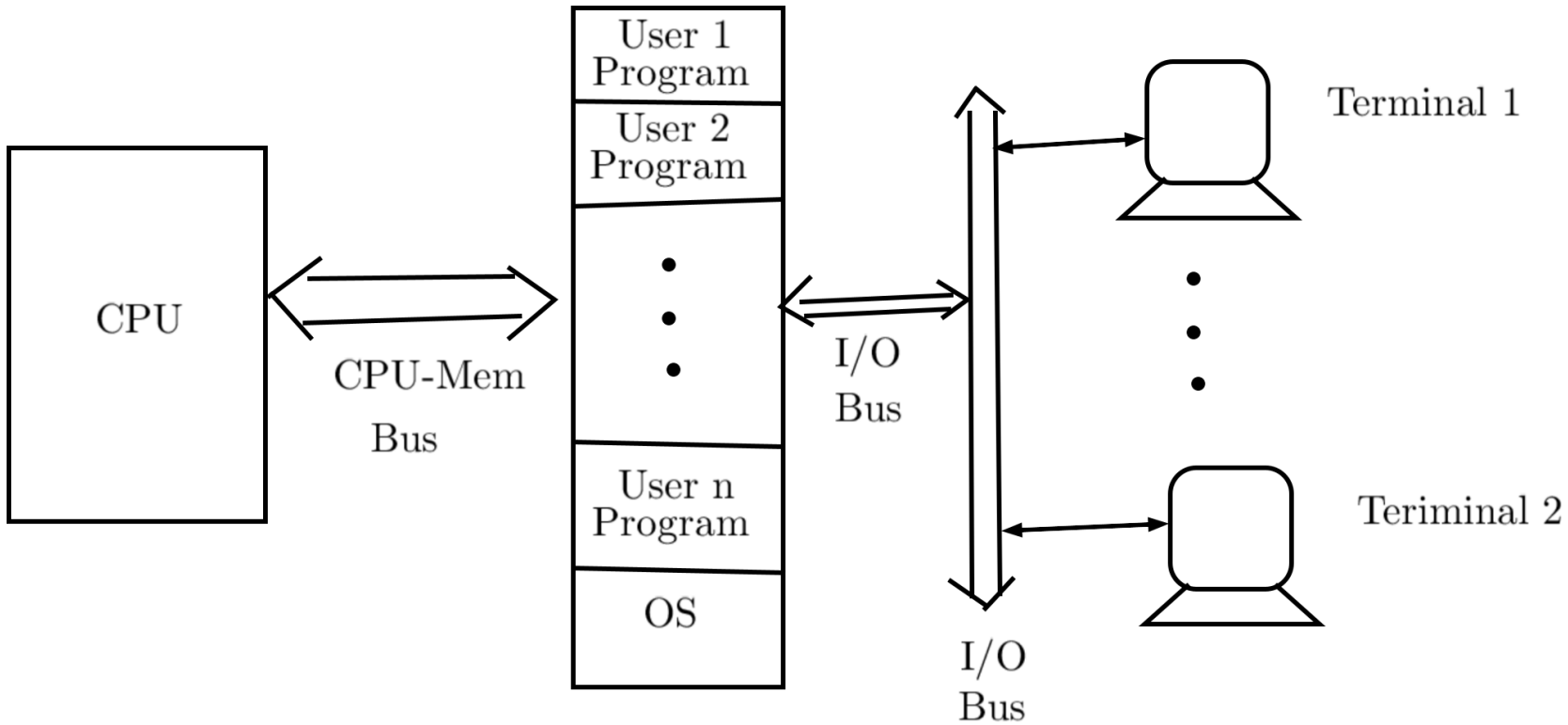

Figure 4: Time sharing operating system

Distributed OS

- A distributed operating system is one that looks to its users like an ordinary centralized operating system but runs on multiple, independent central processing units (CPUs).
- The key concept here is transparency. In other words, the use of multiple processors should be invisible (transparent) to the user.
- *Distributed Applications*: LinkedIn, FB, InstaGram, X, Google search. *Distributed OS*: Solaris, Ubuntu, Linux, OSF/1, Dynix, Locus
- *Terms:* Distributed Algorithms, distributed systems, distributed applications, distributed computing, distributed databases, distributed file systems, distributed ledger technology (block chain).
- A fundamental problem in distributed systems: Lack of global state information.

Monolithic Operating Systems

- Oldest kind of OS structure (are DOS, original MacOS)
- Problems:
  - Single process OS software
  - One application runs at a time
  - dedicated CPU
  - misuse I/O devices
  - etc. . .
- No good for fault containment (or multi-user).
- Need a better solution ?

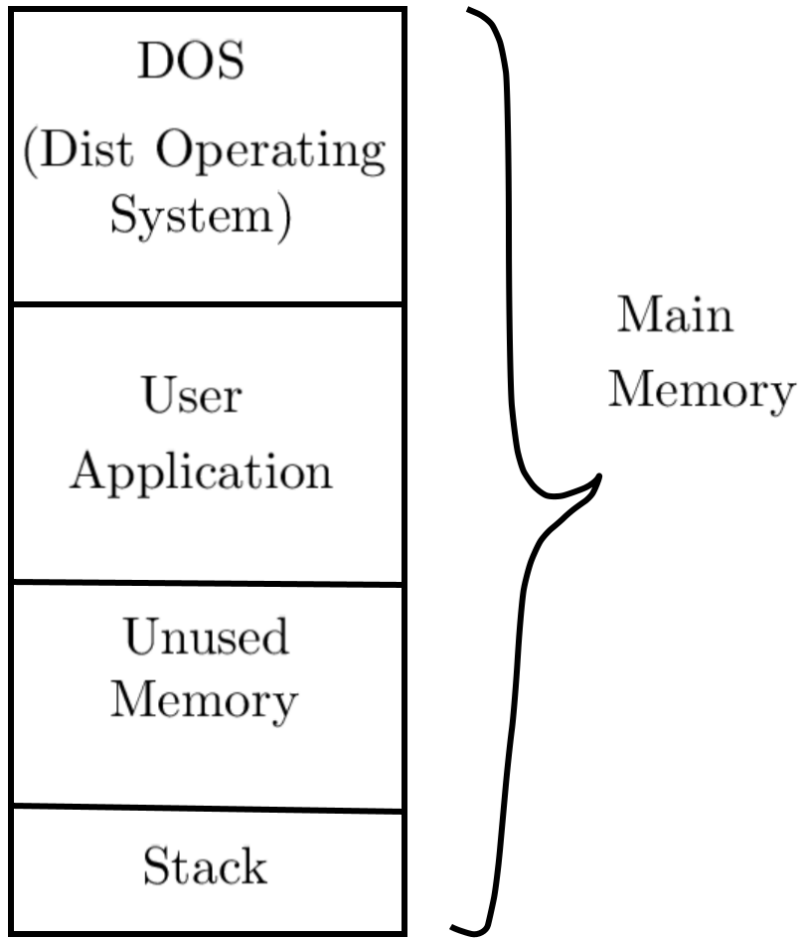

Figure 5: Monolithic os

Dual-Mode Operation (Why it is needed?)

- Stops buggy (or malicious) program from doing bad things
- provide hardware support to distinguish between two different modes of operation:
  - *User Mode* : executing on behalf of a user (i.e., application programs)
  - *Kernel Mode* : executing on behalf of the operating system

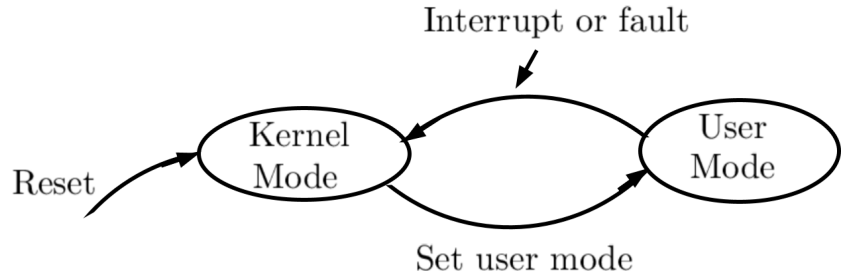

Figure 6: User and kernel mode

- Interrupt, e.g., pressing a key by user, fault, e.g., page-fault, i.e., required data is not found in RAM, so it should be brought from disk
- Mode bit in hardware, e.g. 0 = kernel, 1 = user mode
- Some machine instructions are possible in kernel mode only