

# Operating Systems

# Objectives

---

- At the end of the meeting, students should be able to:
  - explain the importance of operating system
  - discuss how operating systems have changed over time

# Definition

- An **Operating System** is a program, implemented in either a firmware or software, which acts as an interface between the user of a computer and the computer hardware.



# Examples

- **Windows**
  - XP, Vista
- **GNU/Linux**
  - Ubuntu, Mandriva
- **Mac OS**
  - Leopard, Snow Leopard



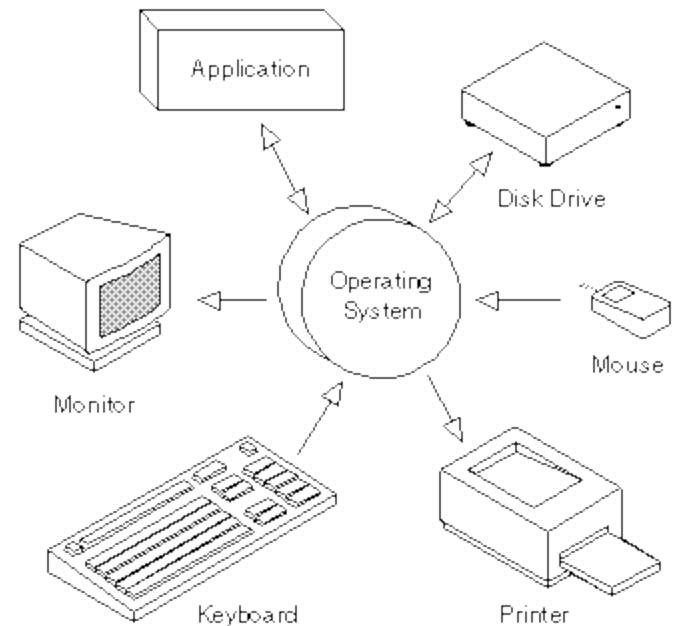
Windows Vista™



# Why is OS Needed?

## 1. Device Management

- Usually the users do not want to know the detail of how a particular peripheral has to be driven.
- Disk, Printer, Monitor/Keyboard, clock, etc.



# Why is OS Needed?

---

## 2. CPU Management

- Scheduling CPU for many programs.
- Making sure that the CPU is busy most of the time.

# Why is OS Needed?

## 3. Memory Management

- Allocating the main memory to several processes with the aim of making sure that a process that is about to take control of the CPU is already in the memory.

# Why is OS Needed?

## 4. File System Management

- Managing the organization of the secondary storage and providing file abstraction that is convenient to the applications running in the system and to computer users.





# Why is OS Needed?

## 5. Protection & Security

- One program does not read from or write to the space of another.
- Providing access controls to user files

# Why is OS Needed?

## 6. Communication & Resource Sharing

- Providing ways for processes to cooperate/communicate with one another.

# Why is OS Needed?

---

## 7. Utilities

- Date and time, disk cleanup, defragmenter, recycle bin, file search, etc.

# Why is OS Needed?

---

## 8. Command Interpreter

- Providing interface between the user and computer that allows high level commands to be issued by the user.

# Why is OS Needed?

---

- In the absence of an operating system, all the functions just described will have to be written by the user and incorporated in the user program.

# OS History

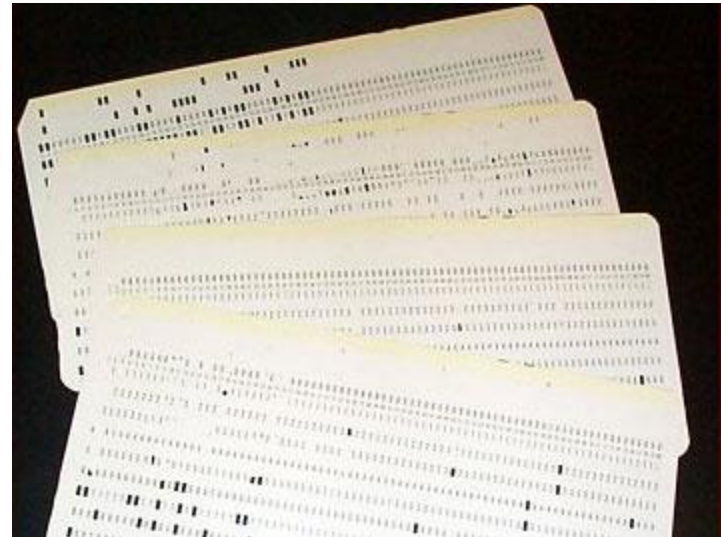
## 1. First Generation (1945-1955)

- No operating system.
- Each user was allocated some time slot, during the time slot, the user had exclusive access to the machine.
- User interacted with bare machine using machine language.
- Inconvenient, time wasted.

# OS History

## 2. Second Generation (1955-1965)

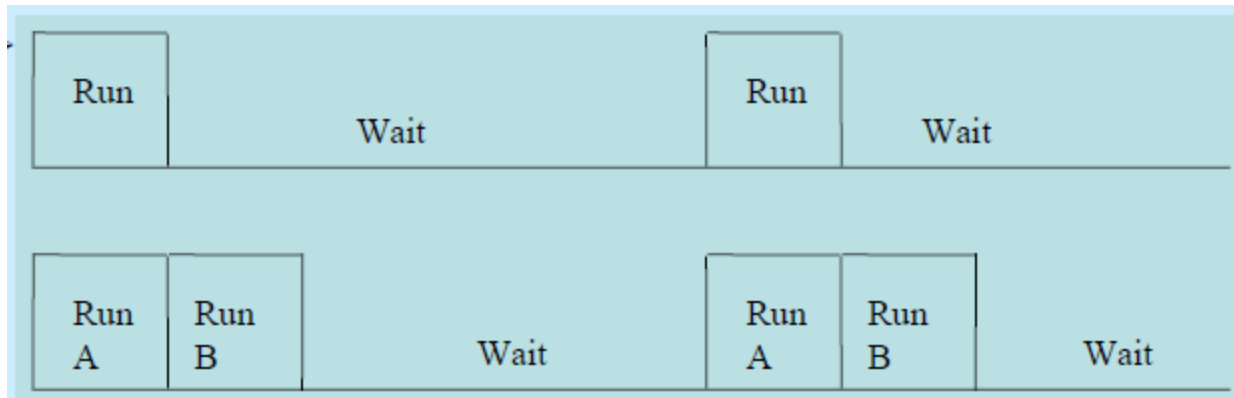
- Simple batch processing
- Initiates the execution of batch of jobs in the computer system.
- Job queue – FIFO.



# OS History

## 3. Third Generation (1955-1970)

- Multiprogramming
- Multiple programs are being processed concurrently.
- It allows a different job to run when current one is doing I/O.

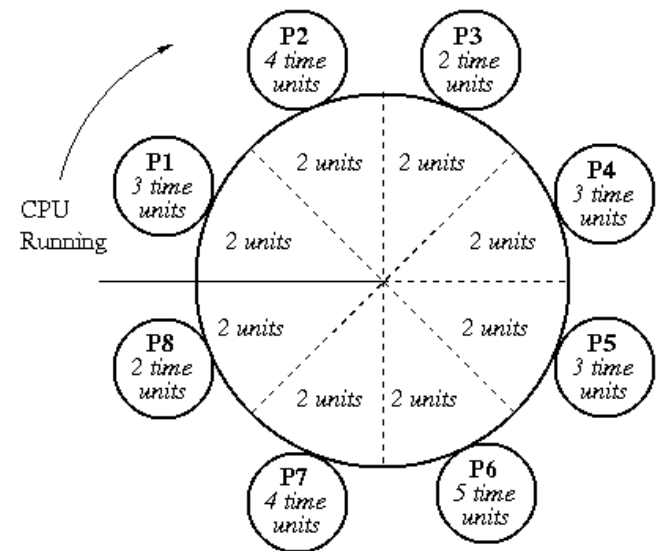




# OS History

## 4. Fourth Generation (1970)

- Multiprogramming timesharing systems
- In the demand of interactive processing.
- N users share the system and each one get  $1/n$  of the system time.
- It uses round robin method to share the CPU.



# OS History

## 5. Personal Computers (1980)

- User friendliness was most important.
- Still multiprogramming.
- May support multiple users.
- Via multiple terminal connections or networking.



# OS History

## 6. Parallel Systems (1985)

- A system that executes several instructions at the same time.

## 7. Distributed Systems (1980)

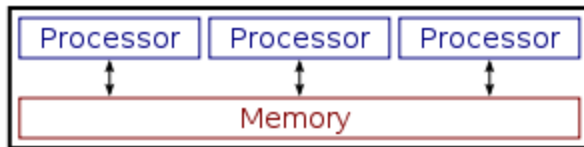
- A collection of independent computers that appears to the user as a single computer.

# Parallel vs. Distributed

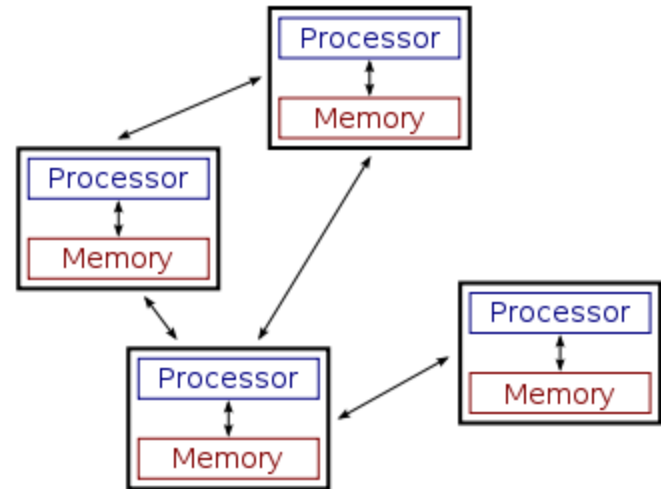
- Distributed OS are also referred to as *loosely coupled systems* whereas parallel processing systems are referred to as *tightly coupled systems*.
- A loosely coupled system is one in which the processors do not share memory and each processor has its own local memory whereas in a tightly coupled system there is a single system wide primary memory shared by all the processors.

# Parallel vs. Distributed

**Parallel**



**Distributed**



# Parallel vs. Distributed

- The processors of distributed operating systems can be placed far away from each other to cover a **wider geographic area** which is not the case with parallel processing systems.

# Parallel vs. Distributed

- The no. of processors that can be usefully deployed is very small in a parallel processing operating system whereas for a distributed operating system a larger no. of processors can be usefully deployed.

# OS History

---

## 8. Ubiquitous/Pervasive

- Mobile, hand-held phones.
- Special small devices: cameras, sensors, wearable devices.

## 9. High-Speed Internet & Wireless Communications