# Operating-System Roles & Overview

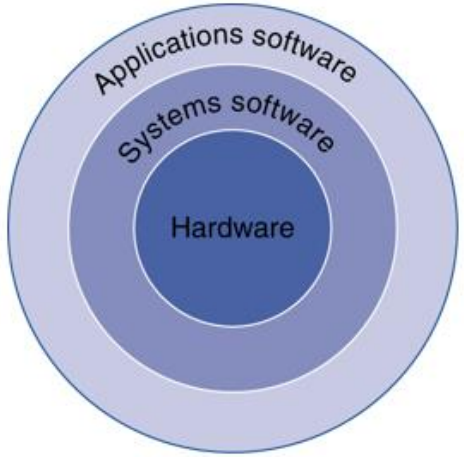
Spring 2022

# Role of OS

Operating System (OS) is system software, which acts as an interface between a user of the computer and the computer hardware. The main purpose of an Operating System is to provide an environment in which we can execute programs.

The main goals of the Operating System are:

- To make the computer system convenient to use.
- To make the use of computer hardware in efficient way.



# Role of OS

Operating System has three main responsibilities:

- Perform basic tasks such as recognizing input from the keyboard, sending output to the display screen, keeping track of files and directories on the disk, and controlling peripheral devices such as disk drives and printers.

- Ensure that different programs and users running at the same time do not interfere with each other.

- Provide a software platform on top of which other programs can run.

# Worksheet Q1

- Which below given tasks need help of operating system?

- Load a program into memory.
- Click your mouse button.
- Call a recursive function from a program
- Close an application.
- Printing a document.

# Design Goals of OS

- Convenience
    - Abstraction of hardware resources
    - Virtualization
- Efficiency
    - Usage of CPU and Memory
- Isolation
    - Protection from multiple processes

# History of OS

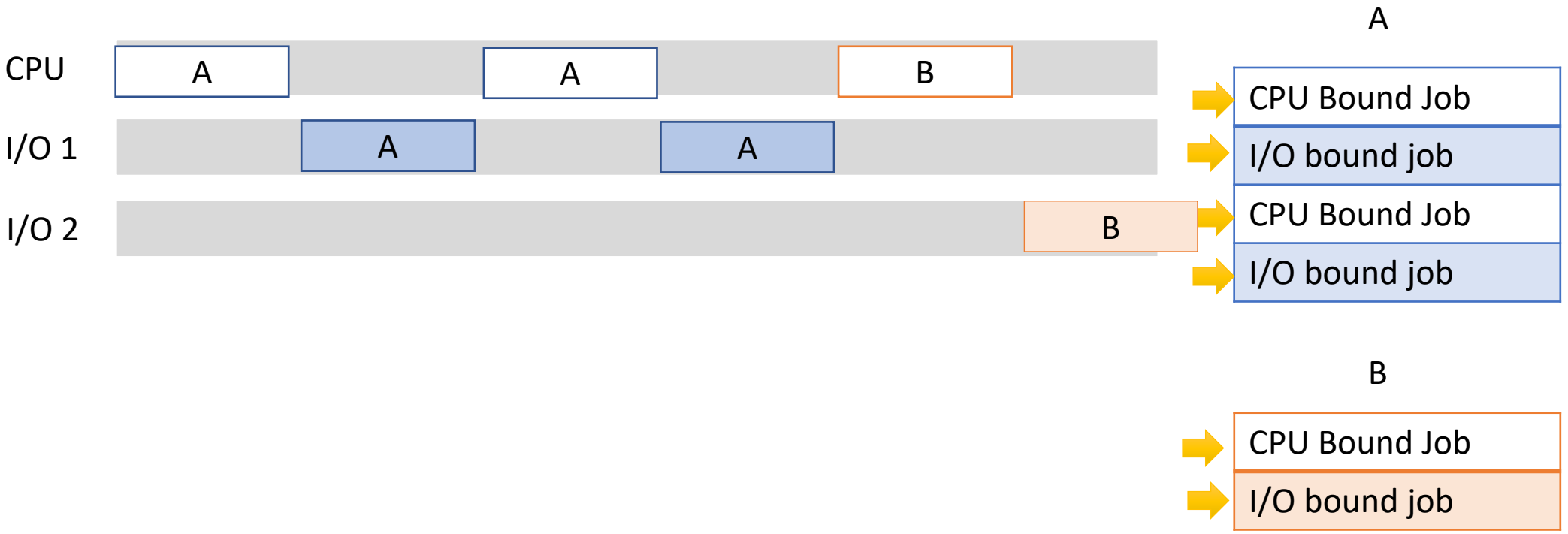
- Started as a library to provide common functionality across programs
- Evolved from procedure call to a system call.
- Evolved from running a single program to multiple processes concurrently.

# OS Resource management

- Optimize resources
  - Good Response time
  - Ensure meeting deadlines for time critical tasks
  - Safe communication between tasks

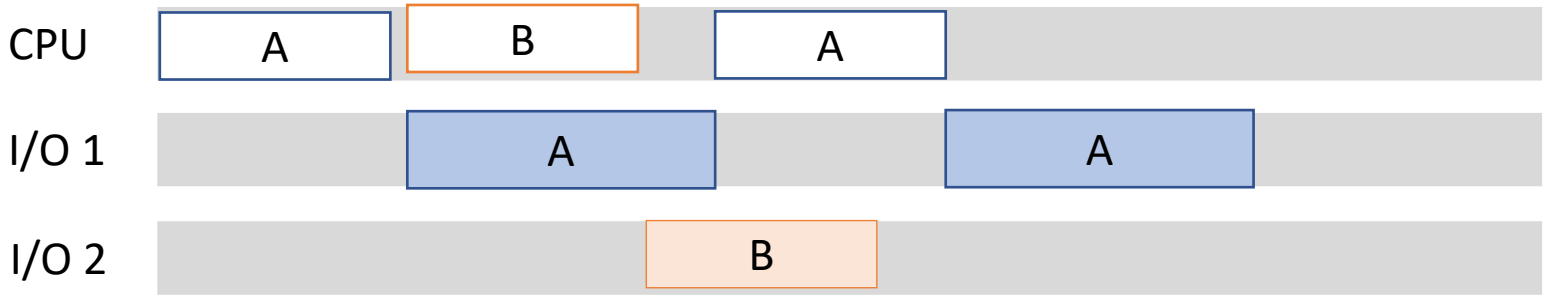
# Resource Allocation - Sequential

- Several programs active in memory and switches execution among the different programs to maximize the use of the CPU and other resources.
- Sequential



# Resource Allocation - Multiprogramming

- Multiprogramming




A

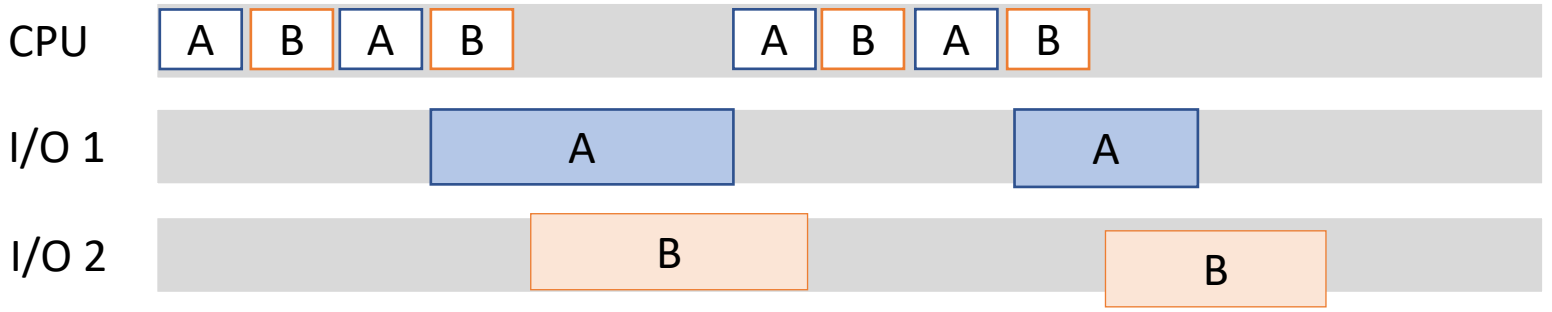
| CPU Bound Job |
| --- |
| I/O bound job |
| CPU Bound Job |
| I/O bound job |

B

| CPU Bound Job |
| --- |
| I/O bound job |

# Resource Allocation – Time sharing

- Timesharing




A

| CPU Bound Job |
| --- |
| I/O bound job |
| CPU Bound Job |
| I/O bound job |

B

| CPU Bound Job |
| --- |
| I/O bound job |
| CPU Bound Job |
| I/O bound job |

# Worksheet Q2

- Processes p1 and p2 share a single CPU and 2 independent I/O devices. Each process executes a compute-bound phase of 10-time units followed by an I/O-bound phase of 10 units.
- The total execution time when Sequential?

- The total execution time when Multiprogramming?

# Example

Processes p1 and p2 execute on a system with a single CPU and 2 identical I/O devices. Each process executes a compute-bound phase followed by an I/O-bound phase. The system uses multiprogramming without time-sharing. The following table shows the lengths of each phase.

| | Compute | I/O |
|---|---|---|
| p1 | 30 | 40 |
| p2 | 10 | 80 |

# Worksheet Q3

Processes p1 and p2 execute on a system with a single CPU and 2 identical I/O devices. Each process executes a compute-bound phase followed by an I/O-bound phase. The system uses multiprogramming without time-sharing. The following table shows the lengths of each phase.

| | Compute | I/O |
|---|---|---|
| p1 | 40 | 70 |
| p2 | 30 | 80 |

# Worksheet Q4

- Four processes execute on a system with a single CPU and a single I/O device. The system uses multiprogramming without time-sharing. Each process has a compute-bound phase of 10-time units followed by an I/O-bound phase of 20 units. A process cannot initiate the I/O phase until the I/O device becomes free. The processes start executing in the order p1, p2, p3, p4.