OPERATING SYSTEM

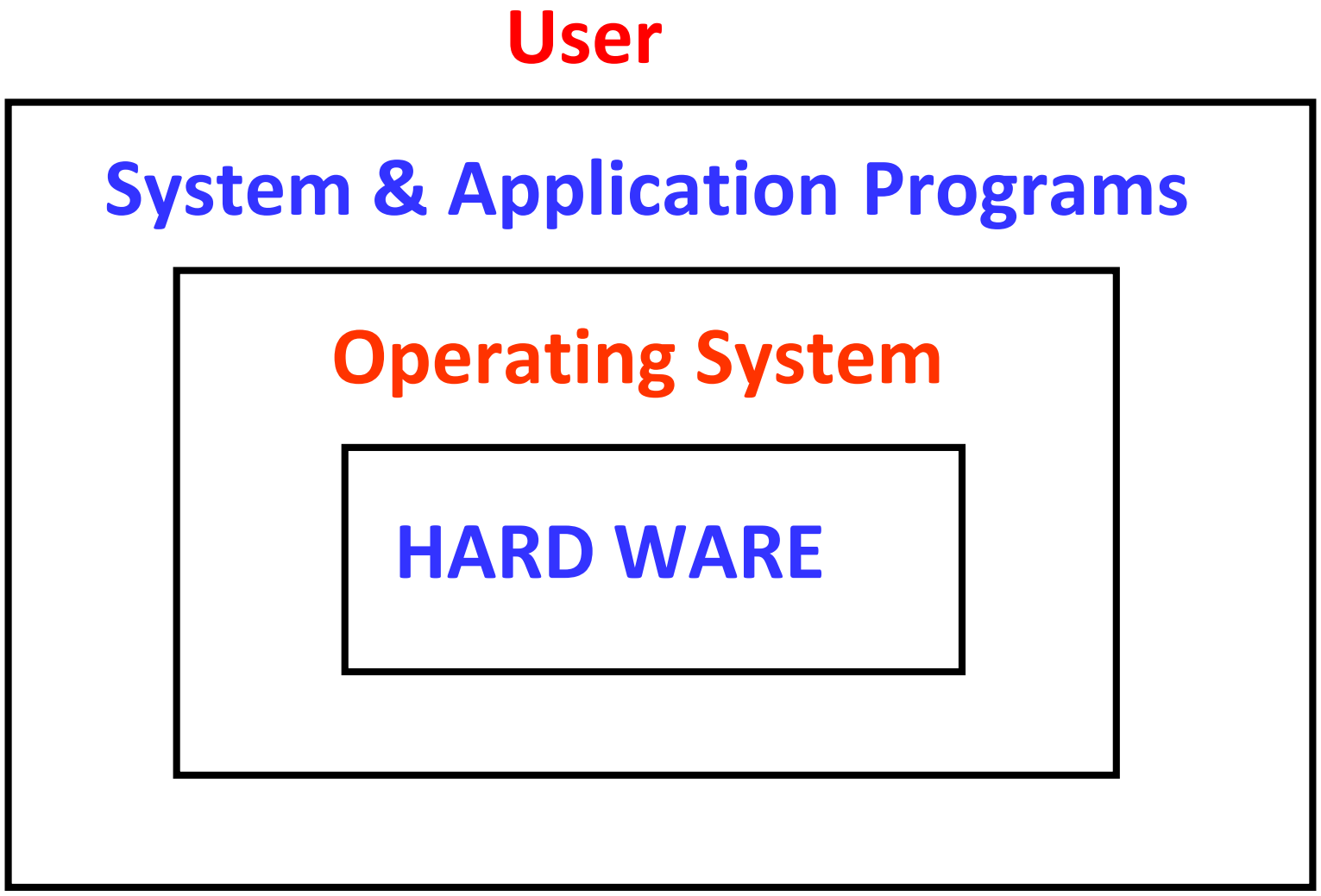
Module 1

Operating System - Introduction

- Operating System is a software, which makes a computer to actually work.
- It acts as an interface between the user and hardware.
- The OS organizes and controls the hardware resources like CPU, memory, I/O devices etc.
- It is a Software that controls the operation of a computer, directs the input and output of data, keeps track of files, and controls the processing of computer programs.

A computer system can be divided roughly into 4 Components-

- **Hardware**
- **OS**
- **System & Application programs**
- **Users.**


User
System & Application Programs
Operating System
HARD WARE

Software

- **System Software**
- **Application Software**

System Software : Is used for satisfying system needs.

Eg. Operating System, Compiler, Linker, Loader.

Application Software : is used for satisfying user needs.

Eg. Tally, Explorer, Paint, text editors .....

OS interfaces

- Users may also interact with the operating system with some kind of software user interface
  - ➢ **CLI (**command line interface) – it enables users to interact with the os by typing commands and these commands are interpreted and executed by the OS
  - ➢ **GUI (** graphical user interface) – this interface uses mouse clicks, menus, submenus, icons etc to interact with the OS. This is more user friendly approach.

Functions of OS

We can explore operating systems in two view points:

- User View
- System View

User View

- In a single-user experience, the goal will be to maximize the work that the user is performing.
- In multi-user environment OS is designed to maximize resource utilization

System View

- Here OS can be viewed as a resource allocator(OS must decide how to allocate them to specific programs and users) and a control program(OS manages the execution of user programs to prevent errors and improper use)

Functions of OS

- **Processor management**
  - Assignment of processors to different tasks.
  - The Operating System will Create the Priorities for the user i.e. it determines and maintains the order in which jobs are to be executed in the computer system

- **Process Management**
- **Program in execution is known as a process. A process needs many resources to do its task and when a process terminates, the os reclaims any reusable resources.**
  - create, execute and delete a process
  - cancel or resume a process
  - schedule a process
  - synchronization, communication and deadlock handling for processes.
  - Makes the Child Process after dividing the Large Processes into the Small Processes.

- **Resource Management**
  - OS allocates computer resources such as CPU time, main memory, secondary storage, and input and output devices to different processes for use.
  - The Operating System will identify at which Time the CPU will perform which Operation and in which Time the Memory is used by which Programs
  - OS also identifies When the Input and Output Devices are used by the which Programs

- **Memory management**
- **To improve cpu utilization & speed of computer, many programs are kept in memory and so we need a memory management technique.**
  - allocation of main memory and other storage areas to system programs, user programs and data
  - free memory
  - re-allocate memory to a program when a used block is freed
  - keep track of memory usage.

- **File Management**
  - creating and deleting a files/directories
  - storage of these files
  - transfer of these files from one location to another
  - editing or modifying file using text editors, other file manipulations etc.
  - Backup of files
  - Secure files

- **Data security and integrity management**
  - OS protects the resources of system
  - it keeps the programs and data separate and protects it from being destroyed by other users
  - Protection is done by User authentication, permissions(read, write), Encryption, and back-up of data etc.

- **Interpretation of commands and instructions**

  - Operating system provides an interface between the computer user and the computer hardware.
  - The user interface is a set of commands(CLI) or a graphical user interface(GUI) via which the user interacts with the applications and the hardware.

Evolution of Operating Systems

- In the early computers there were no Operating Systems.
- With the advancement of the commercial computer services we have come across a number of Operating Systems software in 1960's .
- Starting from the DOS, a lot much Operating Systems has got developed through out the stages like the UNIX, Windows etc.
- The most commonly-used Operating Systems for laptops and modern Desktops Operating Systems are the Microsoft Windows.

Os STRUCTURE

- Stages include
  - Serial Processing
  - Simple Batch Systems
  - Multi-programmed Batch Systems
  - Time Sharing Systems

Serial Processing

- Performs all the instructions into a **Sequence Manner**
- Instructions given by the user will be **executed in a FIFO Manner**
- only one job is done at a time and only on the completion of that job the other job starts.

- No operating system: direct interaction Programmer - computer hardware
- Machine operation controlled from a console

  with display lights(indication of errors) and toggle switches
- Typical input device: card reader;

  typical output device: printer
- No scheduling, programmers had to sign up for "computer time"
- "Job" included loading the compiler and the source program, saving compiled program, linking and loading the compiled program, program execution

- **Disadvantages**

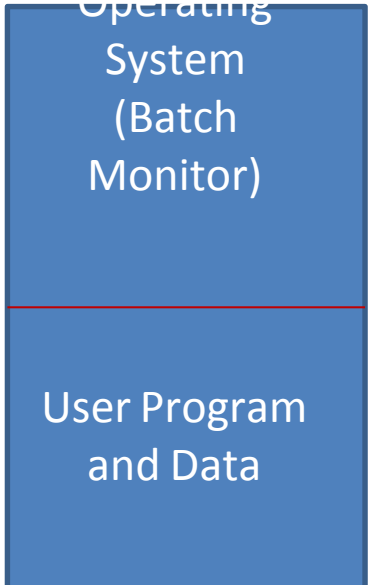
  - ➢ It is slow
  - ➢ Manual intervention is more
  - ➢ Poor utilization of system resources

Batch processing

- **Batch processing** is the execution of a series of identical non-interactive programs ("jobs") on a computer without manual intervention.
- i.e. Executing a series of non-interactive jobs all at one time.
- **In Batch processing** a series of identical non-interactive jobs are grouped together as a batch and the batch is executed all at once.
- batched jobs were executed automatically one after another saving its time by performing the activities (like loading of compiler) only for once.

- Requires the program, data and commands to be submitted together in the form of a job.
- It resulted in improved system utilization due to reduced turn around time(total time between submission of a process and its completion).
- **Batch monitor-** memory resident portion of the batch OS.

  - reads ,interprets and execute commands.



Batch processing

Advantages

- Very little user interaction
- Much improved system utilization.

Disadvantages

- **Non-interactive environment** - No interaction is possible with the user while the program being executed
- **Low CPU utilization** - CPU Sat idle during I/O operations and transmission from one job to another
- **Off-line Debugging** – cannot correct the errors at the time it occurs

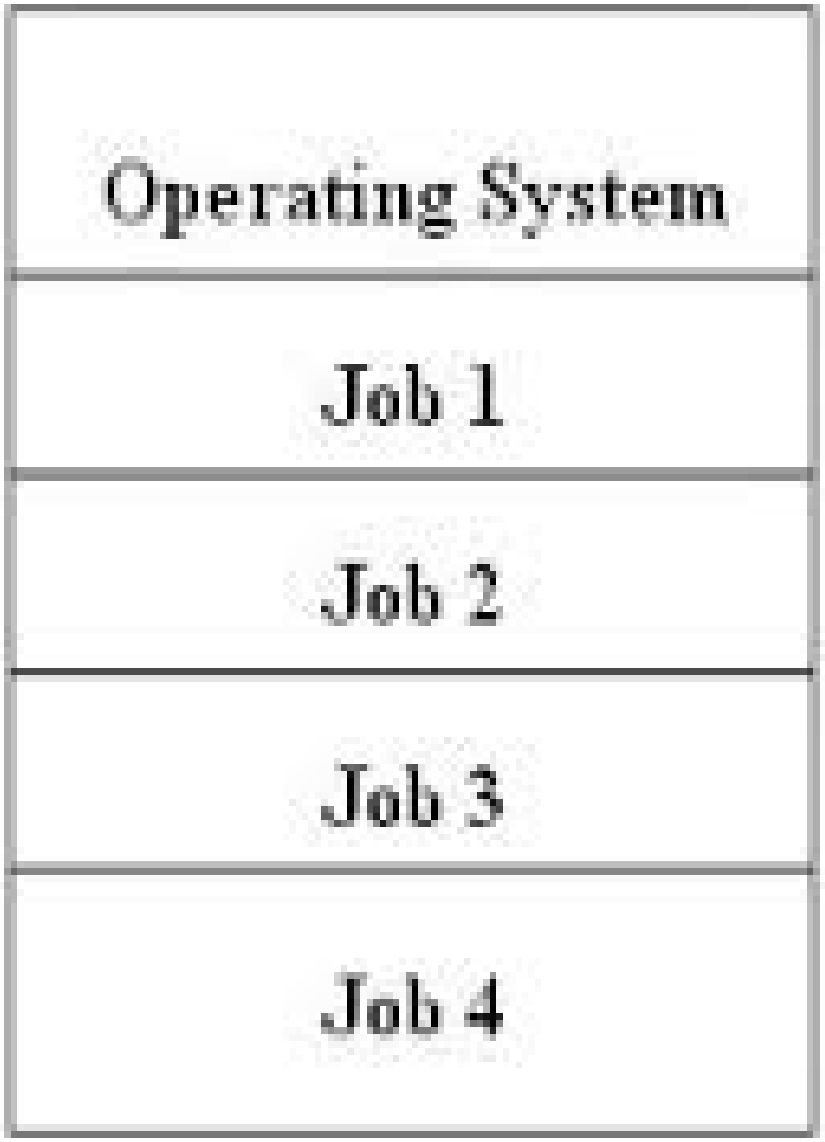
Multiprogramming

- Multiprogramming is a form of **parallel processing** in which several programs are run at the same time on a single processor.
- Multiprogramming is a feature of an **OS which allows running multiple programs simultaneously on one CPU.**
- **Job Pool –** all the jobs in the system are kept here waiting for allocation to main memory
- **Job Scheduling -** Job scheduling schedules or decides which process/jobs from **job Pool** will get allocation to main memory.

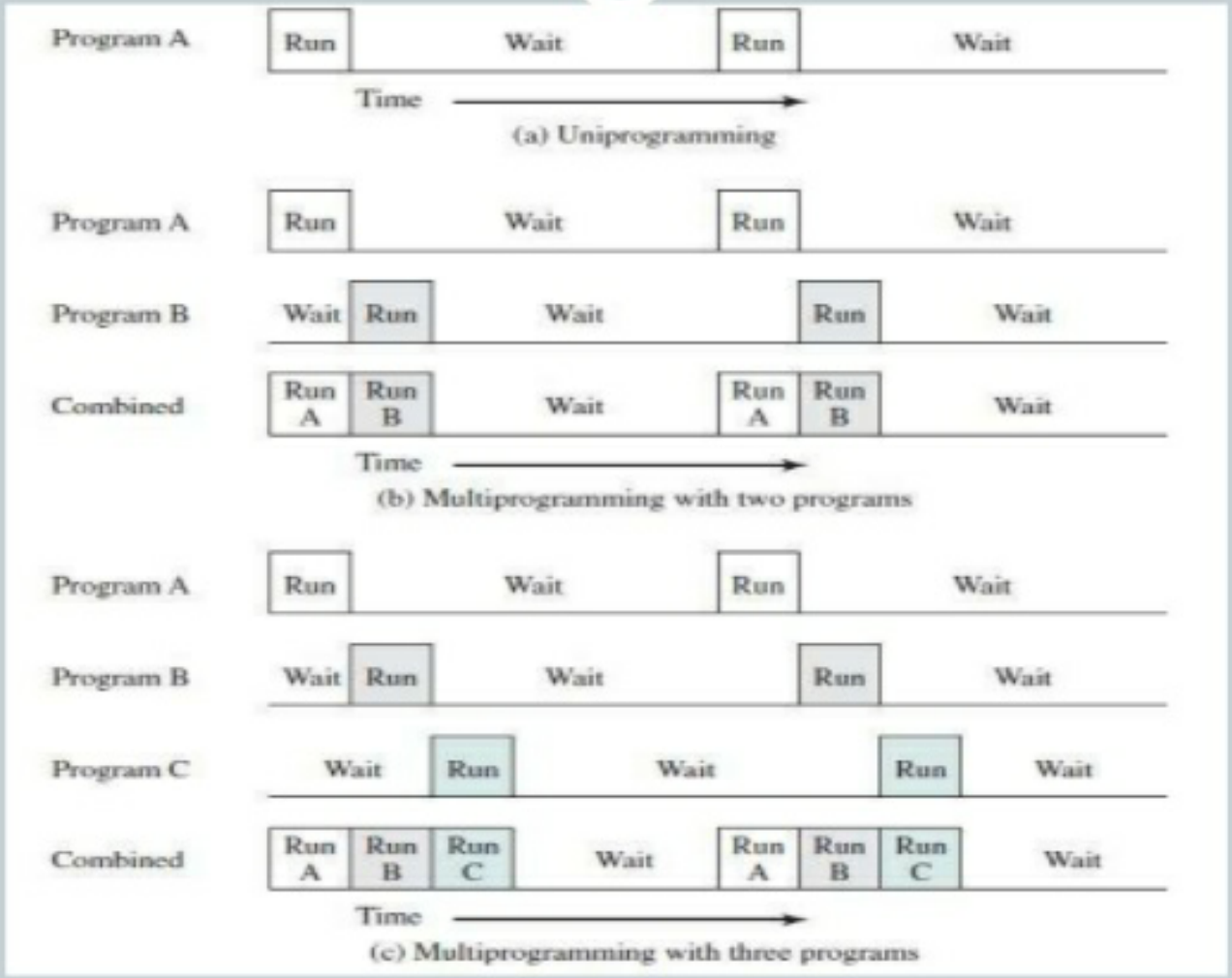
- **CPU Scheduling –** it is the process by which an Operating System decides which ready process/jobs in the main memory get to use the CPU.
- Multiprogramming is the first instance where OS makes decisions for users
- Supports multitasking, Multi access , Multi processing ,multiuser etc…

- Actually programs don’t run simultaneously, **but OS divides time for each program according to priorities**
- Sharing will be done by the OS according to **certain priorities**
- Multiprogramming is a Basic form of **parallel processing**

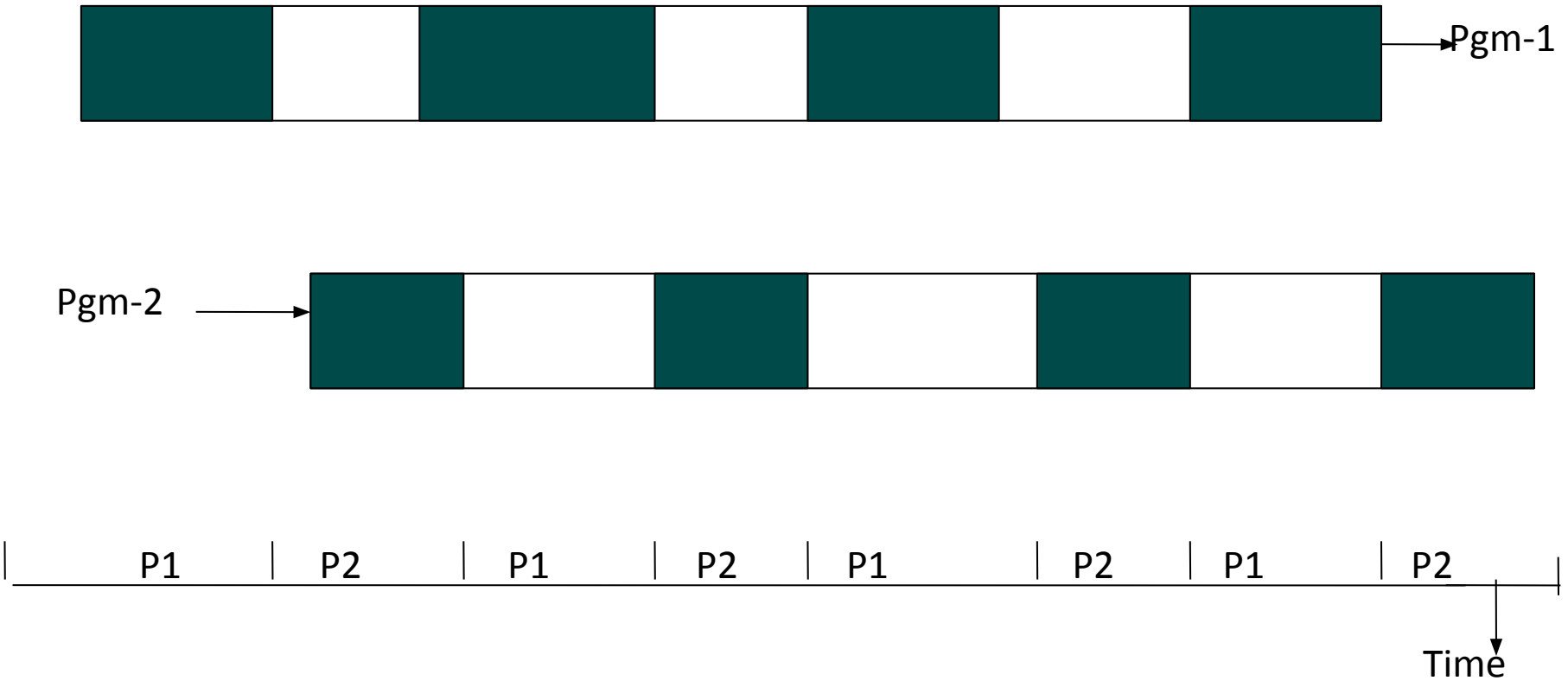
Memory Layout of Multiprogramming



Multiprogramming



Multi programming OS



Multi programming

Advantages

- **High CPU Utilization**
- **Support multiple user**
- **Efficient memory utilization**
- **More CPU throughput**

Disadvantages

- **Powerful memory management is required**
- **CPU Scheduling is must.**
- **User cannot interact with the job when it is being executed**

Time Sharing OS

- Time sharing is a logical extension of multiprogramming.
- In Time sharing, Processor's(CPU) **time is shared among multiple users or tasks** simultaneously
- **Time slice is defined by the OS, for sharing CPU time between processes.**

- **Time sharing provides sharing a computing resource among many users by means of multiprogramming and multi-tasking**
- Here CPU executes multiple jobs and the **user can interact** with each program(using keyboard or mouse) while it is running.
- Here the user feels that the entire system is dedicated to him/her as the **switches occur rapidly**.
- A program loaded into memory and executing is known as a process.

- **Time sharing systems**
  - Need Good memory management and protection
  - Virtual memory technique should be used for reasonable response time
  - Provides File management
  - Requires good CPU scheduling for concurrent execution of processes

Operating-System Structure

- ability to multiprogram.
- A single program cannot, in general, keep either the CPU or theI/O devices busy at all times.
- **Multiprogramming** increases CPU utilization by organizing jobs so that the CPU always has one to execute.

- The operating system keeps several jobs in memory simultaneously
- This set of jobs can be kept in the job pool
- The operating system picks and begins to execute one of the jobs in memory.
- the job may have to wait for some task, such as an I/O operation, to complete
- operating system simply switches to, and executes, another job.
- When *that* job needs to wait, the CPU is switched to *another* job,
- the CPU is never idle.
- The process of loading the job into memory for execution is known as job scheduling
- several jobs are ready to run at the same time, the system must choose among them. Making this decision is **CPU scheduling,**
- **virtual memory-**it enables users to run programs that are larger than actual **physical memory.**

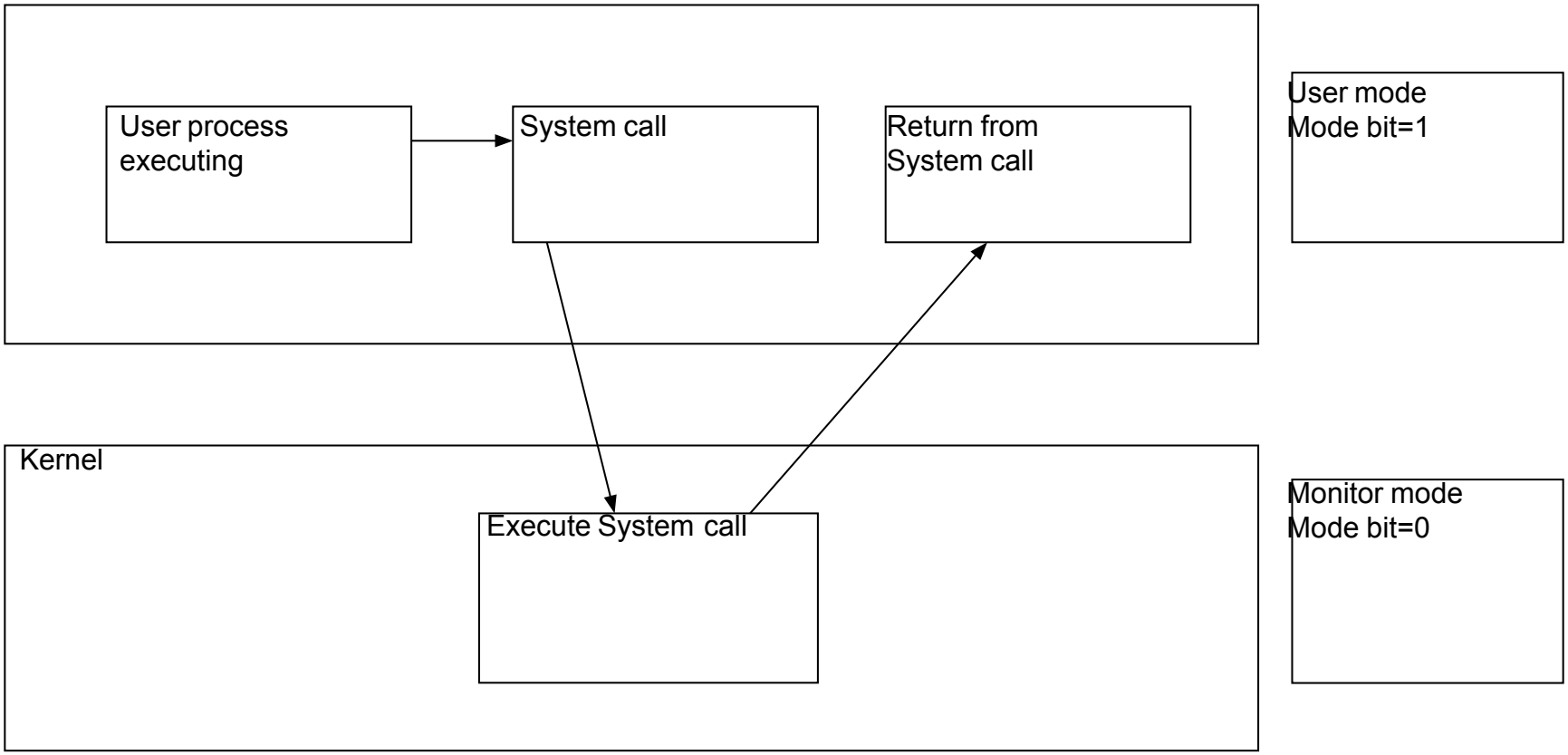
OS Operations

- To ensure proper functioning of the system
- protect the OS and other programs and data from any malfunctioning program.
- Protection is needed for all shared resources
- OS provides a hardware support which differentiates the different modes of execution.
- OS should ensure that an incorrect program does not adversely affect other programs.

- **Dual Mode Operation :**
  - Dual Mode Operation is used to protect the system from other malicious programs.
  - For this protection, it uses two modes of operations namely
  - **User mode and Monitor Mode** (Supervisor mode, System Mode, Kernel Mode, Privileged Mode)
  - There is a **mode bit** associated with the computer h/w to indicate the current mode of operation
    - 1 indicates User Mode & execution is done on behalf of user
    - 0 indicates Monitor Mode & execution is done on behalf of OS

- When the system is booted it is in **Monitor Mode** and after loading OS, the user processes are started in the **User Mode**
- The machine instruction which can cause harm are designated as **Privileged instructions** and are allowed to execute only in **Monitor Mode.**
- If these privileged instructions try to execute in user mode, then h/w treat it as an illegal instruction without executing

Dual Mode Operation



Timer

- There are chances that OS can stuck in an infinite loop and never returns to the OS.
- To avoid this a Timer and clock is maintained.
- The timer interrupts after a specified time and checks the sequence of events
- OS initializes the counter with the amount of allowable execution time for a process and the timer is decremented by 1 for every clock tick and when the timer value becomes negative, the interrupt is sent to terminate the program.

Operating system services

- **User Interface –** CLI, Batch , GUI
- **Program execution –** load,run, end
- **I/O operations –** I/O devices controlled by OS
- **File system manipulation** – files/dir manip, permissions
- **Communication** –shared memory & msg passing
- **Error detection –** aware of errors in h/w & user pgms
- **Resource allocation –** multiuser & multi job (cpu scheduling)
- **Accounting –** usage statistics(track of which user & how much**)**
- **Protection –** multi user env., secure using authentication

System call

- Provides the interface to the services made available by an OS.
- Eg : system calls needed to read and copy from one file to another.
- Can occur Either -directly or indirectly
- System call to library functions.
- Three general methods for parameter passing in system call - register
  - block
  - stack

What is the purpose of system calls?

- System calls allow user-level processes to request services of the operating system.

System calls grouped roughly in to Five

1. Process CNTRL- create ,end , abort, load , execute, wait time&Event etc
2. File mgnt- create & delete file, open & close, read,write, reposition, set & get file attributes
3. Device mgnt- request & release device, read,write, reposition, set & get device attributes
4. Information maintenance –get/set time&date, get/set system data, get & set process, file & device attributes
5. Communication – create & delete connection,send & receive msg, transfer status info, attach or detach remote devices