

# UNIT 1

## INTRODUCTION OF OPERATING SYSTEM

Prepared By : Prof. Khyati G. Raiththa

# OBJECTIVES

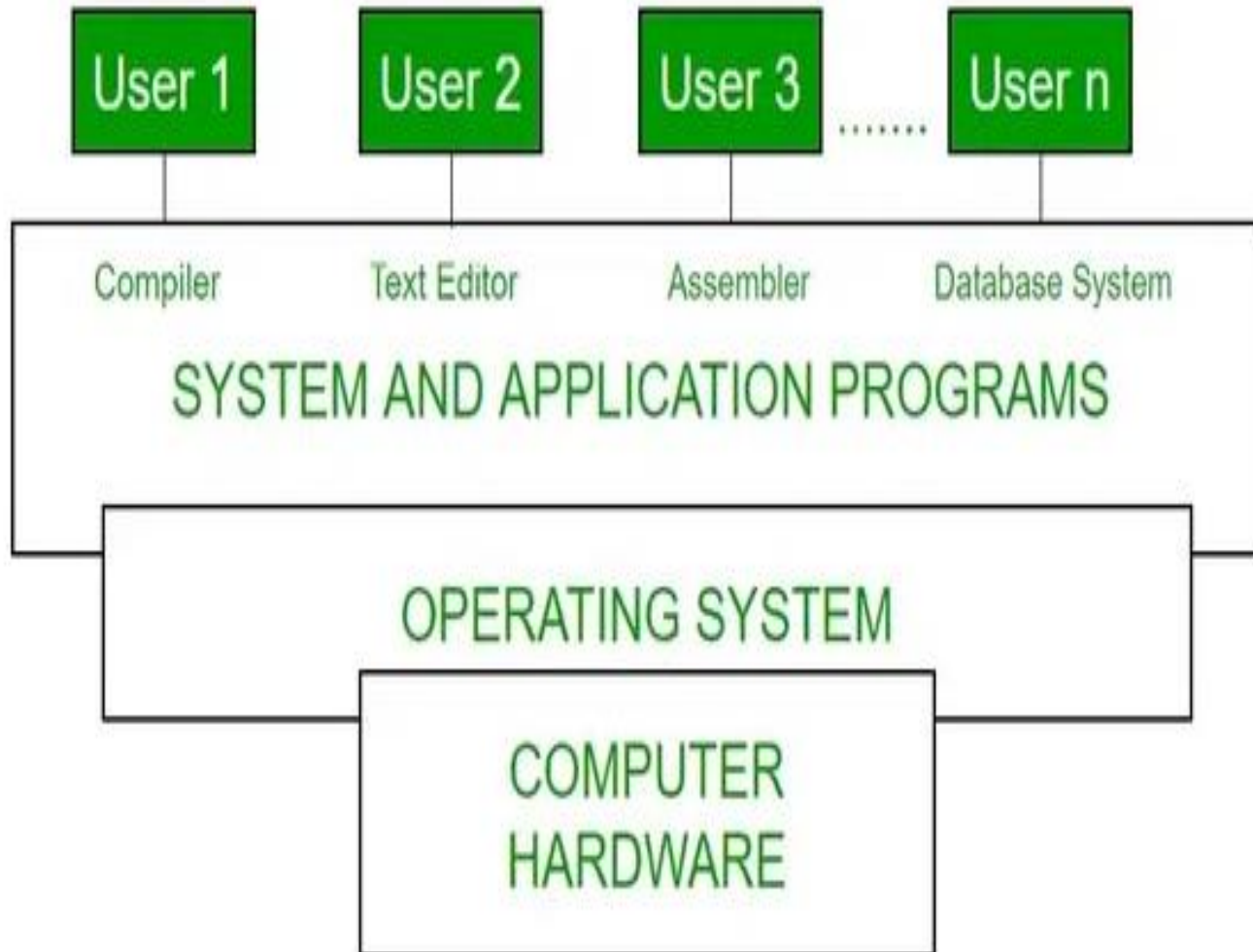
- Introduction of OS
- Objectives and Functions of OS
- Evolution of OS
- Major achievements
- Characteristics of Modern OS
- Functions of Kernel and Shell

# INTRODUCTION OS

# WHAT IS OS ??

- ◉ An Operating system is a bridge between computer hardware and user program.
- ◉ It act as intermediary between the user of computer and hardware of computer.
- ◉ An operating system is software that manages computer hardware

# CONCEPTUAL VIEW



## CONTI...

- ⦿ Every computer must have an operating system to run other programs.
- ⦿ The operating system coordinates the use of the hardware among the various system programs and application programs for various users.
- ⦿ It simply provides an environment within which other programs can do useful work.

# OBJECTIVES AND FUNCTIONS OF OS



# EFFICIENCY

- ⦿ Allows the computer system resources to be used in an efficient manner.
- ⦿ This is because the system configuration takes less time.
- ⦿ By default, the operating system handles system tasks such as allocating resources to processes and resolving conflicts between different programs and users.
- ⦿ This saves the user time and results in a more efficient result.



# CONVENIENCE

- ◉ In the absence of an operating system, users would have to deal with the hardware directly without access to the pre-configured utility packages that come with an operating system.
- ◉ This would make using a computer extremely inconvenient.
- ◉ Operating systems allow users to go right to work on the tasks they want to do without dealing with the burden of setting up the system beforehand.

# ABILITY TO EVOLVE.

- ◉ The user can fully utilize the computer hardware and software without having to deal with the accompanying difficulties.
- ◉ Hardware may updates, New services of software introduce.
- ◉ An OS should be constructed in such a way as to permit the effective development, testing, and introduction of new system functions at the same time without interfering with service.

# SECURITY.

- ◉ It prevents unauthorized access from other user or program.
- ◉ It should have password protection or some kind of protection technique.

# FUNCTIONS OF OS

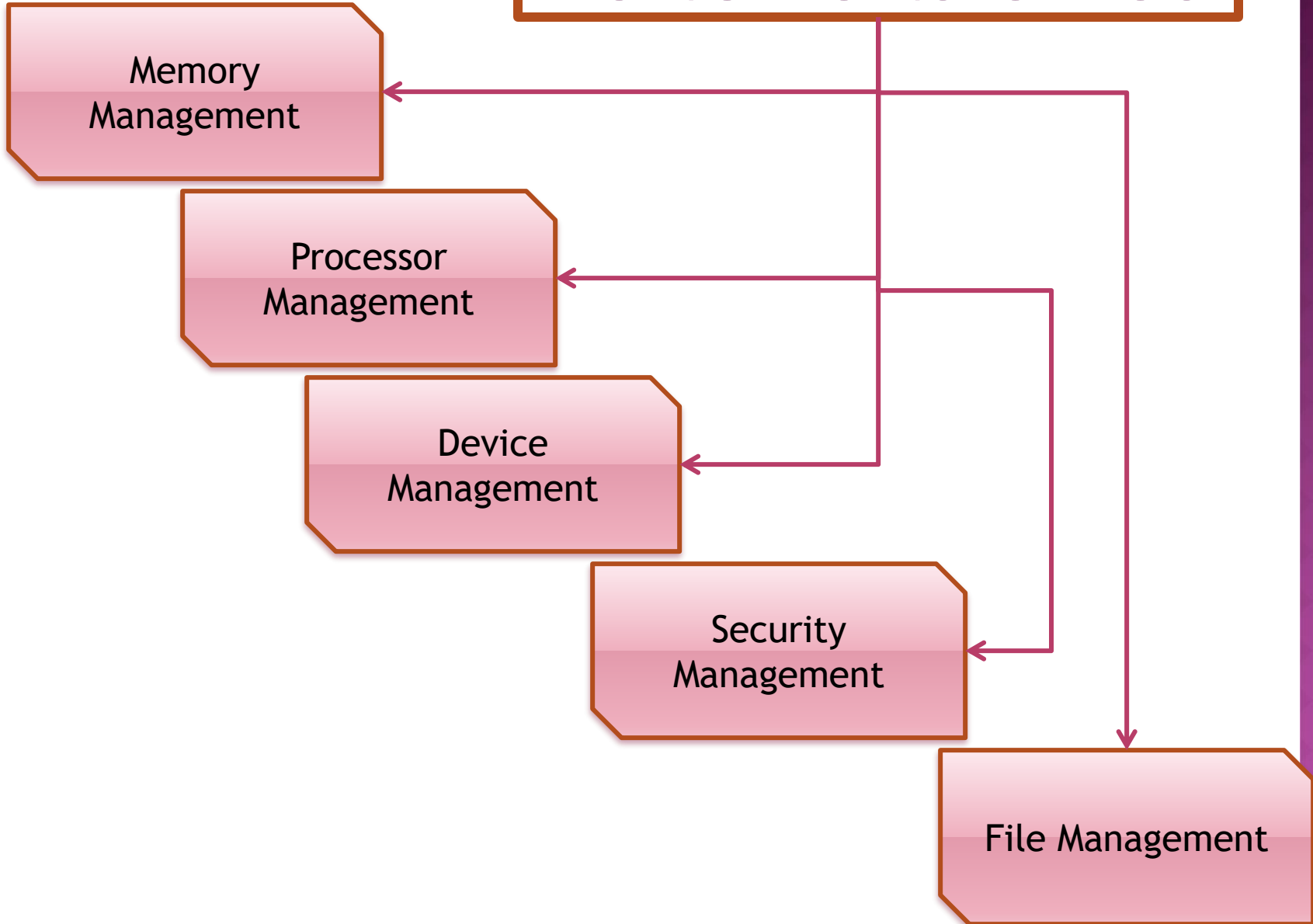
Memory  
Management

Processor  
Management

Device  
Management

Security  
Management

File Management



# MEMORY MANAGEMENT

- ◉ The operating system manages the Primary Memory or Main Memory.
- ◉ An operating system manages the allocation and de-allocation of memory to various processes.
- ◉ and also ensures that the other process does not consume the memory allocated to one process.

# PROCESSOR MANAGEMENT

- ◉ In a multi-programming environment, the OS decides the order in which processes have access to the processor.
- ◉ It also manage how much processing time each process has. This function of OS is called Process Scheduling.
- ◉ An operating system manages the processor's work by allocating various jobs to it and ensuring that each process receives enough time from the processor to function properly.

# DEVICE MANAGEMENT

- ◉ An OS manages device communication via its respective drivers.
- ◉ It performs the following activities for device management.
  - Allocates devices effectively and efficiently.
  - De-allocates devices when they are no longer required.
  - OS controls the working of these input-output devices.

# FILE MANAGEMENT

- ◉ An Operating System carries out the following file management activities.
  - It keeps track of where information is stored.
  - User access settings.
  - The status of every file, and more.
- ◉ These facilities are collectively known as the file system.
- ◉ An OS keeps track of information regarding the creation, deletion, transfer, copy, and storage of files in an organized way.



# SECURITY

- ◉ The operating system uses password protection to protect user data and similar other techniques.
- ◉ It also prevents unauthorized access to programs and user data.
- ◉ The operating system provides various techniques which assure the integrity and confidentiality of user data.

# EVOLUTION OF OS

Time Sharing OS

Multiprogrammed  
Batch OS

Simple Batch  
OS

Serial Processing  
OS

# SERIAL PROCESSING (1940-1950)

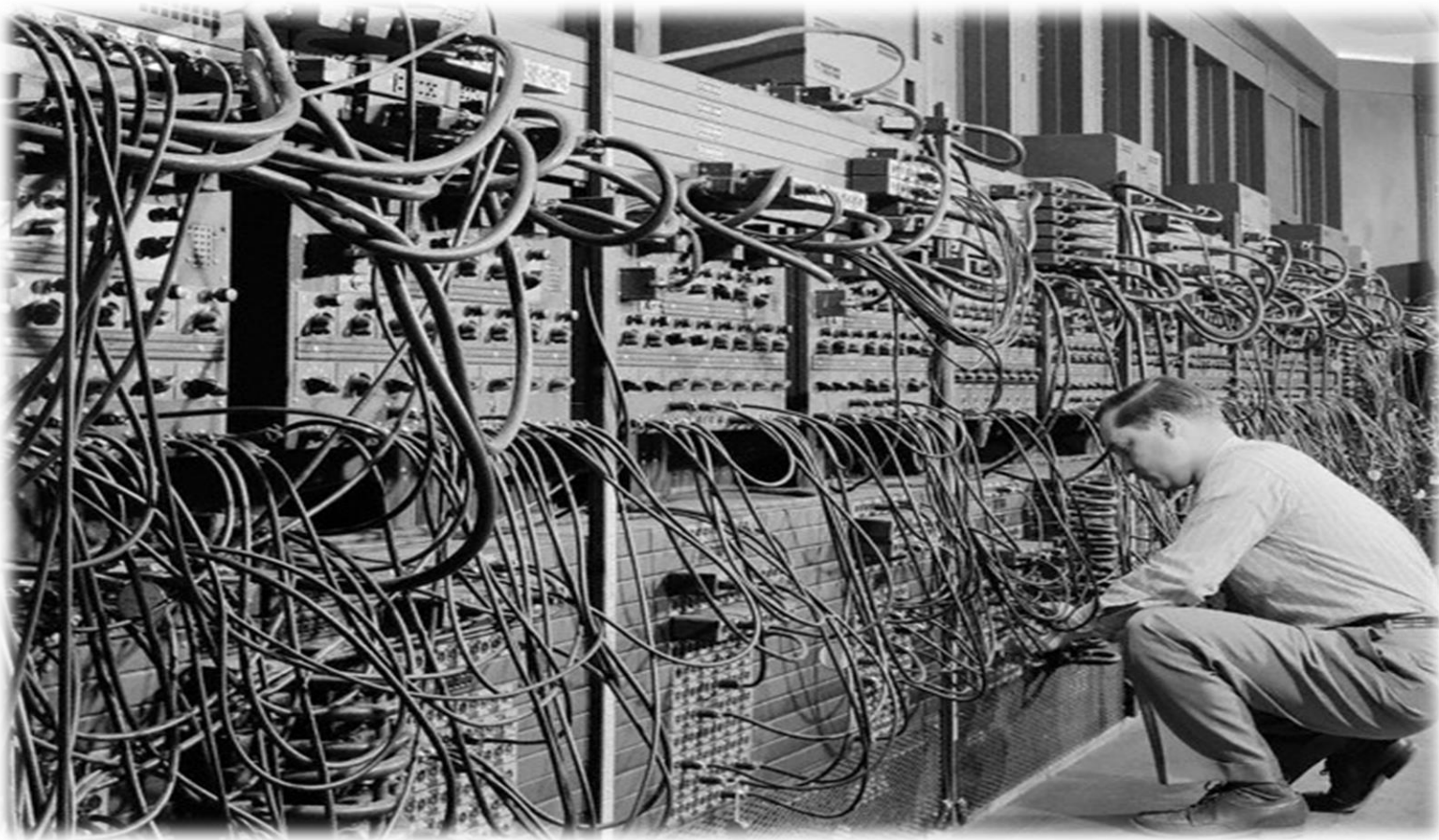
- Only one Process can carried out at one time is known as Serial Processing.
- This OS work :
  - No operating system.
  - It display lights, toggle switches, input device, and printer.
  - Scheduling (take half n hour block)
  - Setup included loading the compiler, source program, saving compiled program, and loading and linking.

# EXAMPLE OF SERIAL PROCESSING COMPUTER



# CONTI...

Programs were loaded into memory manually using switches, punched cards, or paper tapes.





# PROBLEMS WITH SERIAL PROCESSING

- ⦿ This systems presented two main problems:

## 1. Scheduling:

- users had to sign-up sheet to reserve computer time, users couldn't know precisely how long it will take to finish their program.

## 2. Setup time:

- A single program (job) requires so many steps and if any error occurs, user typically had to go back to the beginning of the setup sequence.
- Thus, a considerable amount of time was spent just in setting up the program to run

# SIMPLE BATCH PROCESSING (1955 - 1965)

- ◉ Early computers were very expensive, and therefore it was important to maximize processor utilization.
- ◉ The wasted time due to scheduling and setup time in serial processing was unacceptable.
- ◉ To improve utilization, the concept of a batch operating system was developed.

# CONTI...

- ◉ Batch is defined as a group of jobs with similar needs.
- ◉ This OS allows user to form batches.
  - The operator batched similar jobs together and then ran in the computer to speed up the processing.
  - The user submits the job on cards or tape to a computer operator.
- ◉ The idea behind this is the use of piece of software known as the monitor.



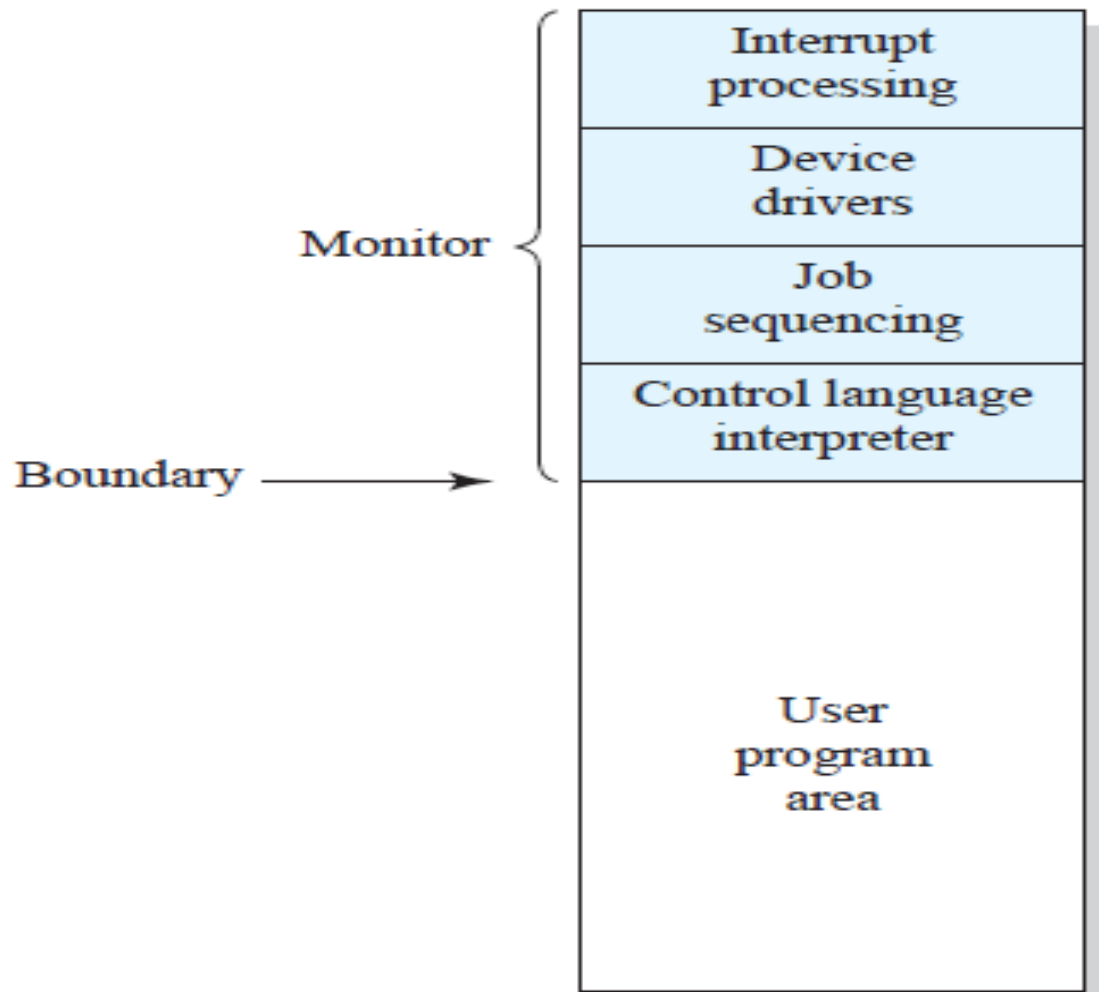
# WHAT IS MONITOR?

- ◉ It is the batch operating system.
- ◉ It is software that controls the sequence of events.
- ◉ Batch jobs together
- ◉ Program branches back to monitor when finished
- ◉ With each job, instructions are included in a primitive form of job control language (JCL).
- ◉ Special type of programming language Provides instruction to the monitor.
  - What compiler to use
  - What data to use

# MONITOR POINT OF VIEW

- ◉ The monitor controls the sequence of events.
- ◉ For this much of the monitor must always be in main memory for execution (Fig 2.3) and that portion is referred to as the resident monitor.
- ◉ The monitor reads in jobs one at a time from the input device
- ◉ As it is read in, the current job is placed in the user program area, and control is passed to this job.
- ◉ When the job is completed, it returns control to the monitor, which immediately reads in the next job.
- ◉ The results of each job are sent to an output device, such as a printer, for delivery to the user.

# BATCH OS



**Figure 2.3** Memory Layout for a Resident Monitor

# PROCESSOR POINT OF VIEW

- Once a job has been read in, the processor will encounter a branch instruction in the monitor that instructs the processor to continue execution at the start of the user program.
- The processor will then execute the instructions in the user program until it encounters an ending or error condition.
- The monitor performs a scheduling function:
- A batch of jobs is queued up, and jobs are executed as rapidly as possible, with no intervening idle time.

# HARDWARE FEATURES OF BATCH PROCESSING

## ◉ Memory protection

- Do not allow the memory area containing the monitor to be altered by the user program

## ◉ Timer:

- Prevents a job from monopolizing the system
- The timer is set at the beginning of each job. If the timer expires, the user program is stopped, and control returns to the monitor.

## ◉ Privileged instructions

- Certain machine level instructions can only be executed by the
- monitor otherwise an error occurs causing control to be transferred to the monitor.

## ◉ Interrupts

- Early computer models did not have this capability.
- This feature gives the OS more flexibility in relinquishing control to and regaining control from user programs.

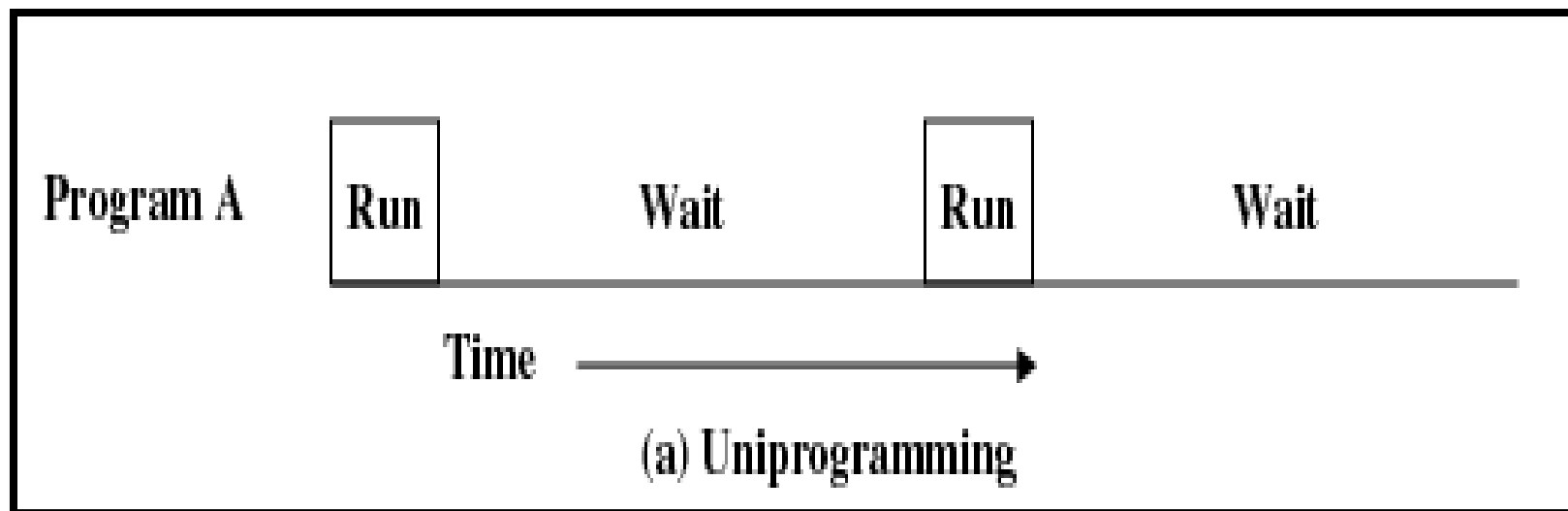
# DISADVANTAGES OF BATCH OS

- ◉ There have been two sacrifices:
- ◉ Some main memory is now given over to the monitor and some processor time is consumed by the monitor.
- ◉ Despite this the simple batch system improves utilization of the computer.
- ◉ Even with the automatic job sequencing provided by a simple batch operating system, the processor is often idle.

# CONTI...

## ◉ Uniprogramming:-

- In a single application system, uni-programming, the processor spends a certain amount of time executing, until it reaches an I/O instruction.
- It must then wait until that I/O instruction concludes before proceeding.
- This inefficiency is not necessary

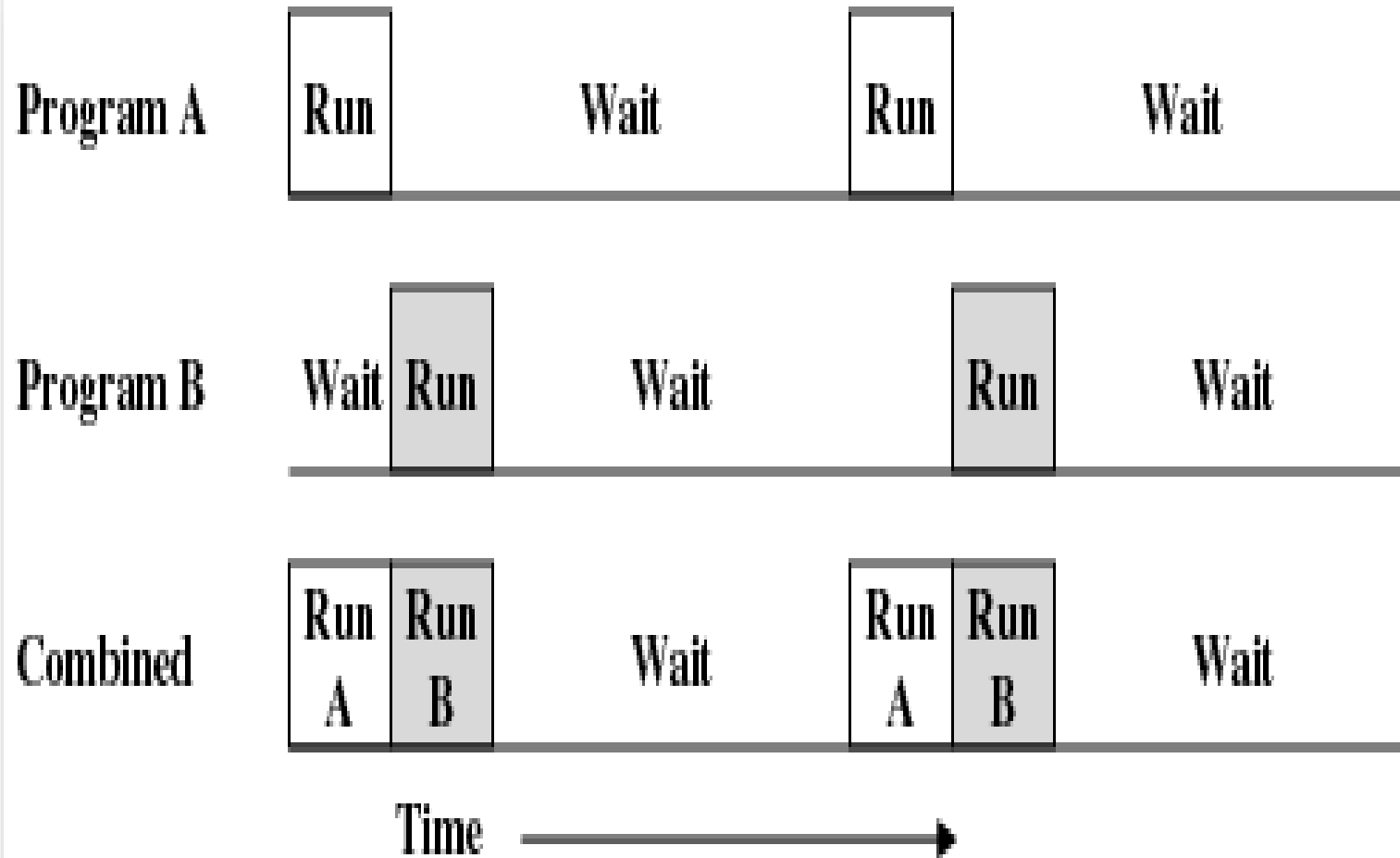


# MULTIPROGRAMMED BATCH SYSTEM(1970).

- ◉ **Multiprogramming:-**
- ◉ We know that there must be enough memory to hold the OS (resident monitor) and one user program.
- ◉ Suppose that there is room for the OS and two user programs.
- ◉ When one job needs to wait for I/O, the processor can switch to the other job, which is likely not waiting for I/O (Figure 2.5b).



# CONTI...



(b) Multiprogramming with two programs

# CONTI...

**Table 2.2** Effects of Multiprogramming on Resource Utilization

	Uniprogramming	Multiprogramming
Processor use	20%	40%
Memory use	33%	67%
Disk use	33%	67%
Printer use	33%	67%
Elapsed time	30 min	15 min
Throughput	6 jobs/hr	12 jobs/hr
Mean response time	18 min	10 min

# TIME SHARING SYSTEM

- ◉ With the use of multiprogramming, batch processing can be quite efficient.
- ◉ For many jobs, it is desirable to provide a mode in which the user interacts directly with the computer.
- ◉ Today, this requirement met by the use of a dedicated personal computer.
- ◉ That option was not available in the 1960s, when most computers were big and costly.
- ◉ So Time sharing was developed.
- ◉ Multiprogramming can also be used to handle multiple interactive jobs.
- ◉ This technique is referred to as time sharing, because processor time is shared among multiple users.

# TIME SHARING SYSTEM

**Table 2.3** Batch Multiprogramming versus Time Sharing

	Batch Multiprogramming	Time Sharing
Principal objective	Maximize processor use	Minimize response time
Source of directives to operating system	Job control language commands provided with the job	Commands entered at the terminal

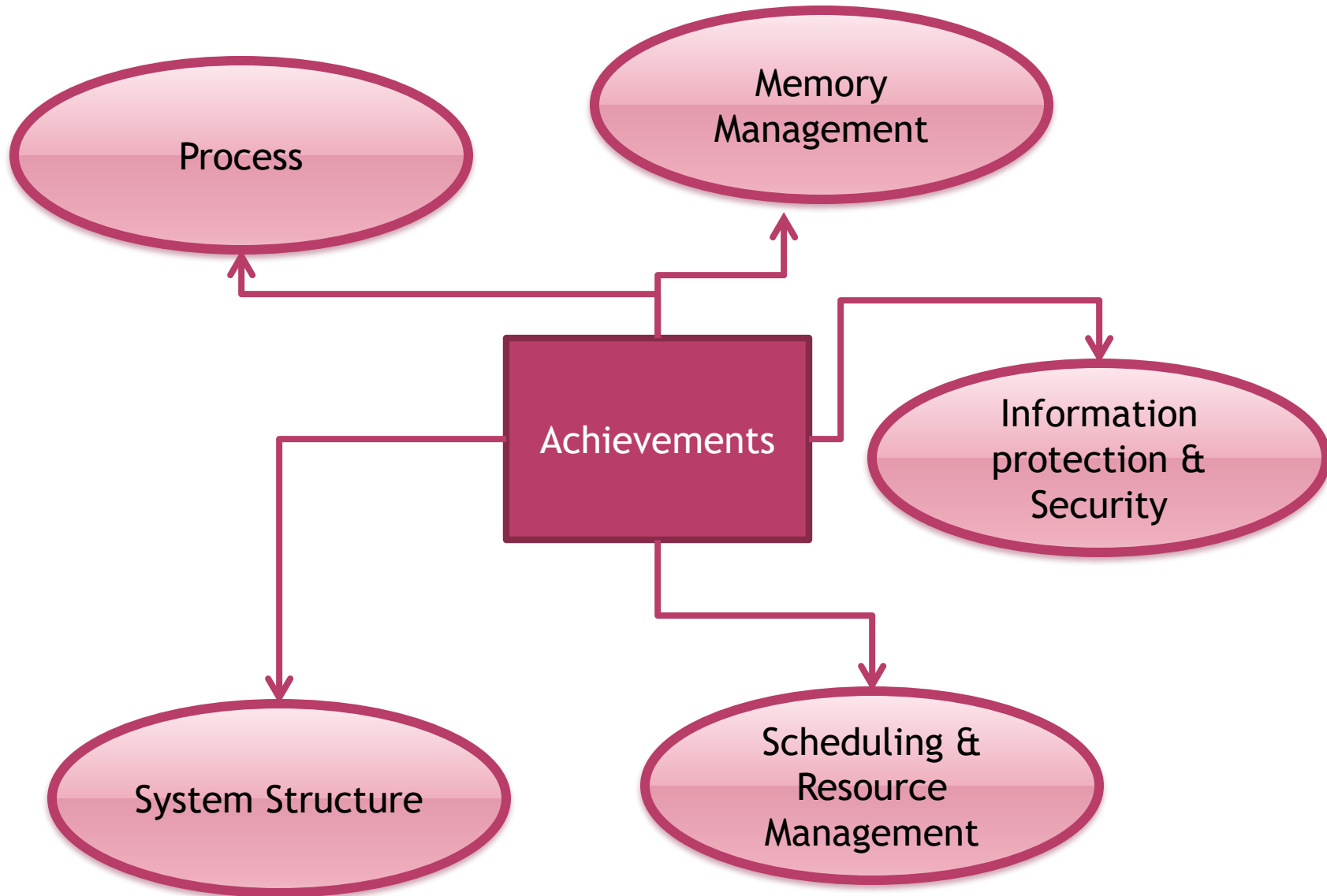
- Programs queued for execution in FIFO order.
- Like multiprogramming, but timer device interrupts after a quantum (time slice).
- Interrupted program is returned to end of FIFO
- Next program is taken from head of FIFO
- Control card interpreter replaced by command language interpreter.
- When OS finishes execution of one command, it seeks the next control statement from user.

# ISSUES WITH TIME-SHARING

- ◉ Multiple jobs in memory must be protected from each other's data.
- ◉ File system must be protected so that only authorized users can access.
- ◉ Contention for resources must be handled like Printers, storage etc

# Major Achievements By OS

# MAJOR ACHIEVEMENTS BY OS



# PROCESS

- Currently running program in system is known as process.
- The term '**process**' was first used by Daley and Dennis.R in 1960.
- Process required 3 things to work.
  - An executable program
  - - Associated data needed by the program
  - - Process state (or Context of the execution program)



# MEMORY MANAGEMENT

- Some principals OS has to manage in memory system.

## Process Isolation

OS prevents independent processes to interfering with each other's memory to use data or instructions.

## Automatic Memory allocation and management

Memory should be allocated dynamically as per process requirement.

# MEMORY MANAGEMENT

## Support of modular programming

Programmers should be able to work dynamically with program modules like define modules, create, destroy, and alter the size of modules.

## Protection and access control

OS is concerned with the problems of controlling the access of data or information from un-authenticate process.

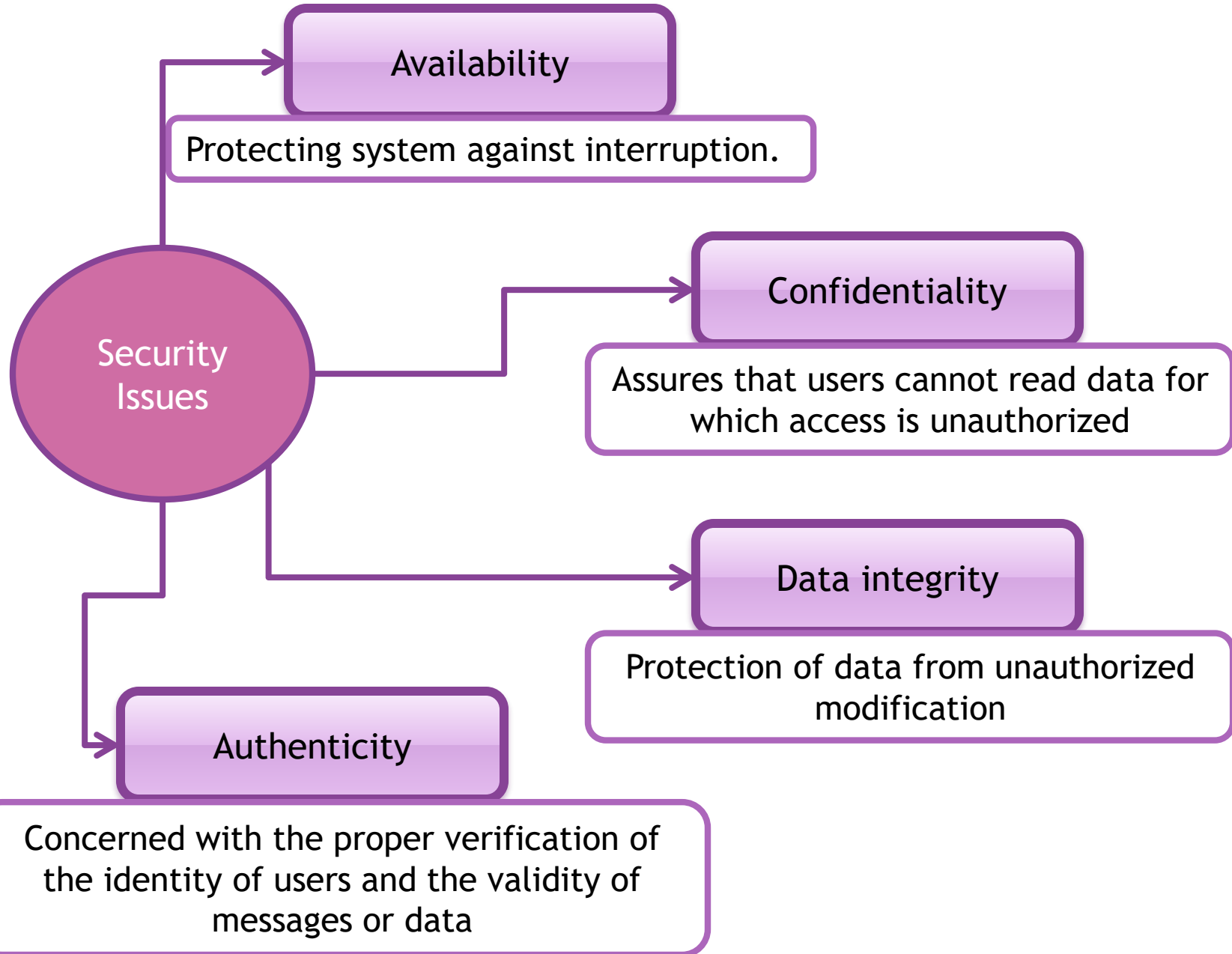
## Long-term Storage

Many application programs require storing information for long periods of time, even though after the computer has been powered down.

# INFORMATION PROTECTION & SECURITY

- ◉ Here the term protection means that secure the resources and information from unauthorized persons.
- ◉ The operating system follows a variety of methods for protection and security.
- ◉ Generally, some of the issues we are facing in terms of security are as below:

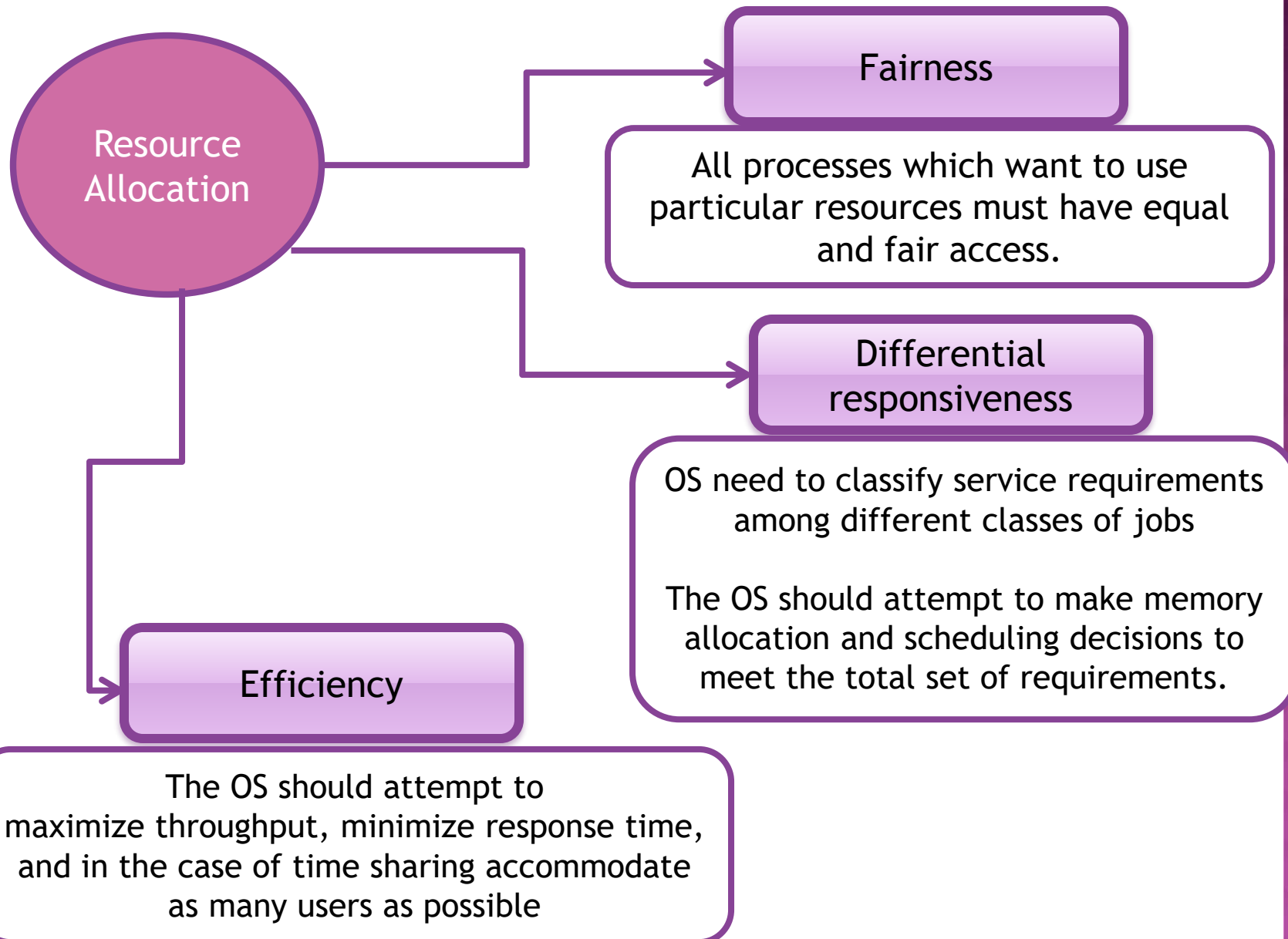
# SECURITY ISSUES



# SCHEDULING & RESOURCE MANAGEMENT

- ◉ A key responsibility of the OS is to manage the various resources available.
- ◉ Os also need to schedule use of resources by the various active processes.
- ◉ Resource allocation has some policies that are as follow:

# PRINCIPALS FOR RESOURCE ALLOCATION



# SYSTEM STRUCTURE

- ◉ OS needs to View the system as a series of levels.
- ◉ Each level performs a related subset of functions.
- ◉ Each level relies on the next lower level to perform more primitive functions.
- ◉ This decomposes a problem into different levels which are as follows:

# PROBLEMS IN LEVELS

## Level 1

- Consists of electronic circuits, where the objects are dealt with the registers, memory cells, and logic gates.

## Level 2

- The processor's instruction set

## Level 3

- Adds the concept of a procedure or subroutine, plus the call/return operations.

## Level 4

- Introduces interrupts, which cause the processor to save the current context and invoke an interrupt-handling routine.

These first four levels are not part of the OS but establish the processor hardware and number of more manageable sub problems.

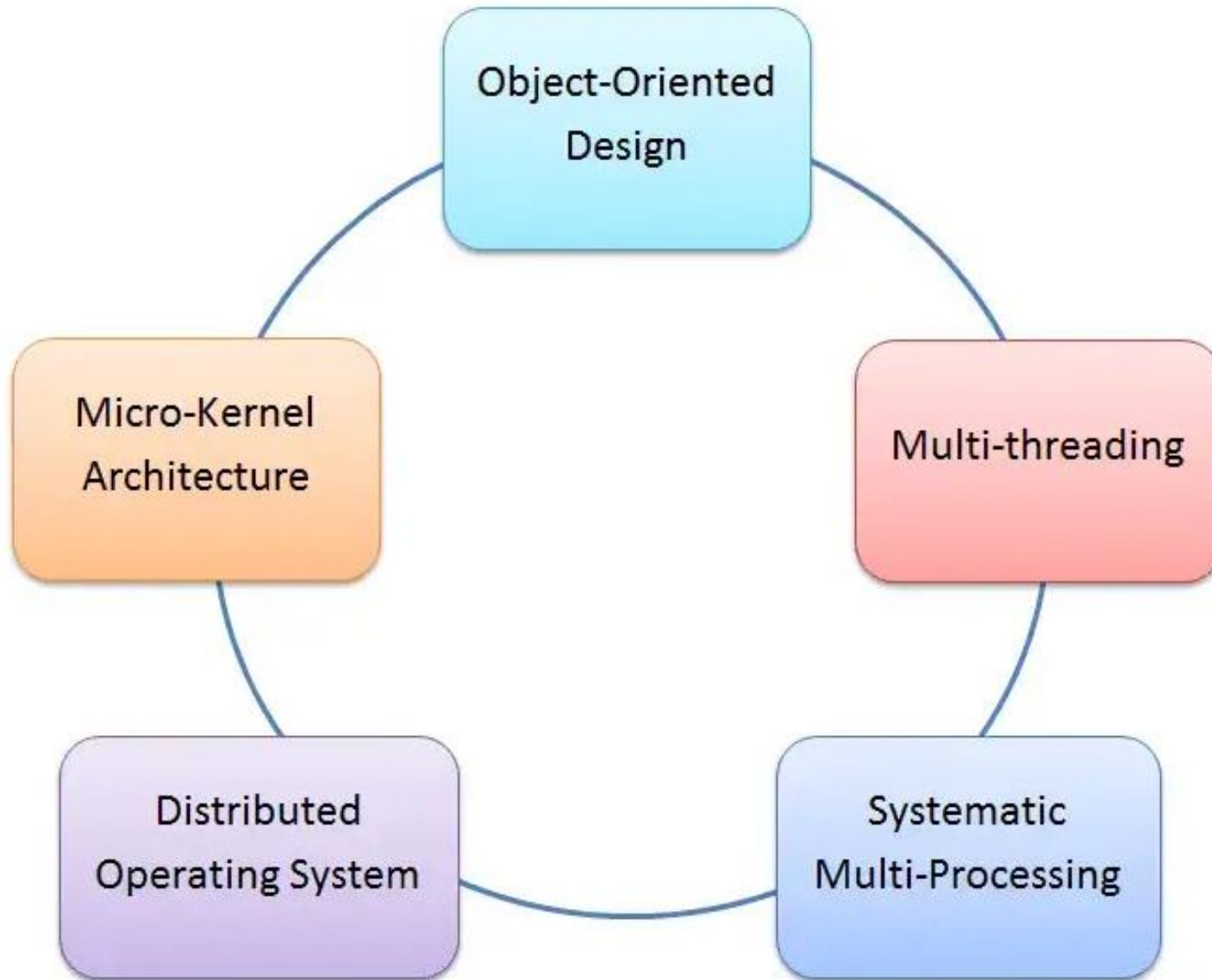


# Characteristics of Modern OS

# CHARACTERISTICS OF MODERN OS

- Huge lines of code, Huge number of unnecessary features aren't only the good Characteristics of Modern Operating System.
- So, we have to consider the design concepts before saying that it is a modern operating system.
- **The Characteristics of Modern OS are as below:**

# CHARACTERISTICS OF MODERN OS



# OBJECT-ORIENTED DESIGN

- ◉ **Object-Oriented Design** is the largest trend in designing the operating system.
- ◉ In which the OS creates the objects for each facility, and each object has a set of operations.

# MULTI-THREADING

- ◉ An application or program divided into several smaller tasks.
- ◉ These tasks executed by the processor concurrently.
- ◉ These smaller tasks are called threads.
- ◉ A thread is a lightweight process or a dispatchable unit of work.
- ◉ A thread has some of the characteristics of a process.
- ◉ The number of independent threads in an application executed by the processor at the time says Multi-threading.

# SYSTEMATIC MULTIPROCESSING (SMP)

- ◉ If a computer has more than one processor, these processors can share the memory and Input\Output facilities.
- ◉ These processors share the same job for execution then the system is called Systematic Multi-Processor System.
- ◉ The operating system designed for systematic processing is called **‘Systematic Multi-Processing Operating System’**.

# ADVANTAGES OF SMP

- ⦿ Performance

- ❖ Allowing parallel processing

- ⦿ Availability

- ❖ Failure of a single process does not halt the system

- ⦿ Incremental Growth

- ❖ Additional processors can be added.

- ⦿ Scaling

- ❖ Vendors can offer a range of products with different price and performance characteristics

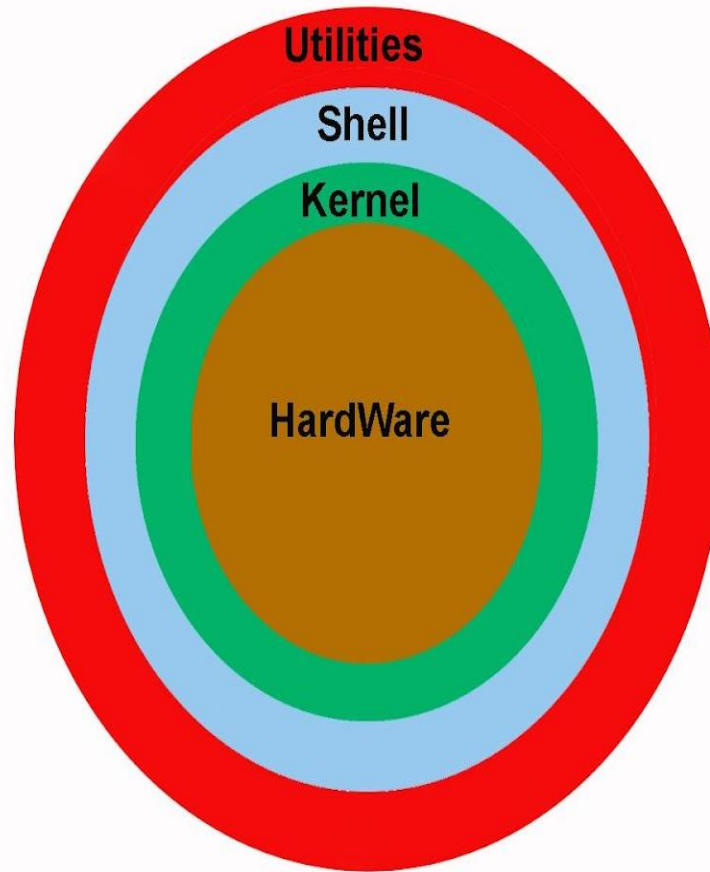
# DISTRIBUTED OPERATING SYSTEM:

- ◉ A Distributed Operating System is an operating system that runs on a network of computers.
- ◉ The operating system, memory files shared by the number of users in the network from the server.
- ◉ In a Distributed Operating System, each user thinks that running on a single large system with one operating system.
- ◉ The users don't need to know where the files are in the network.



# Functions of Kernel and shell

# GENERAL STRUCTURE OF OS



## CONTI...

- ◉ Both the Shell and the Kernel are the Parts of this Operating System.
- ◉ These Both Parts are used for performing any Operation on the System.
- ◉ When a user gives his Command for Performing Any Operation, then the Request Will goes to the Shell Parts,
- ◉ The Shell Parts is also called as the Interpreter which translate the Human Program into the Machine Language and then the Request will be transferred to the Kernel.
- ◉ So that Shell is just called as the interpreter of the Commands which Converts the Request of the User into the Machine Language.
- ◉ Kernel is also called as the heart of the Operating System and the Every Operation is performed by using the Kernel , When the Kernel Receives the Request from the Shell then this will Process the Request and Display the Results on the Screen.

# OPERATIONS PERFORMED BY KERNEL

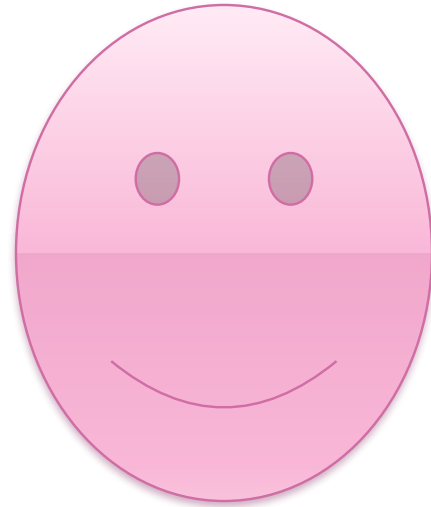
- ◉ It Controls the State the Process Means it checks whether the **Process is running or Process is Waiting** for the Request of the user.
- ◉ Provides the Memory for the Processes those are Running on the System Means Kernel Runs the Allocation and De-allocation Process.
- ◉ The Kernel also Maintains a **Time table** for all the Processes those are Running Means the Kernel also **Prepare the Schedule Time** means this will Provide the Time to various Process of the CPU.

## CONTI...

- ◉ When a Kernel determines that the Logical Memory doesn't fit to Store the Programs. Then he uses the Concept of the Physical Memory which Will Stores the Programs into Temporary Manner. Means the Physical Memory of the System can be used as Temporary Memory.
- ◉ Kernel also maintains all the files those are Stored into the Computer System and the Kernel Also Stores all the Files into the System as no one can read or Write the Files without any Permissions.

# USE OF SHELL

- ◉ A shell is a computer program that provides a text-based interface for interacting with the operating system.
- ◉ It enables users to enter commands, which are then executed by the operating system.
- ◉ shells also provide a scripting language, which can be used to automate tasks.
- ◉ The shell serves as the operating system's outer layer.
- ◉ If the user specifies the Command for Performing Any Operations, the Request for Shell Parts will be sent.
- ◉ The Shell Parts, also known as Interpreters, interpret human programs for machines.



*Thank you...*