UNIT -1
INTRODUCTION TO OPERATING SYSTEMS

Marwadi University

Computer Engineering Diploma

Unit no: 1
Introduction to Operating System

Operating System (09CE2405)

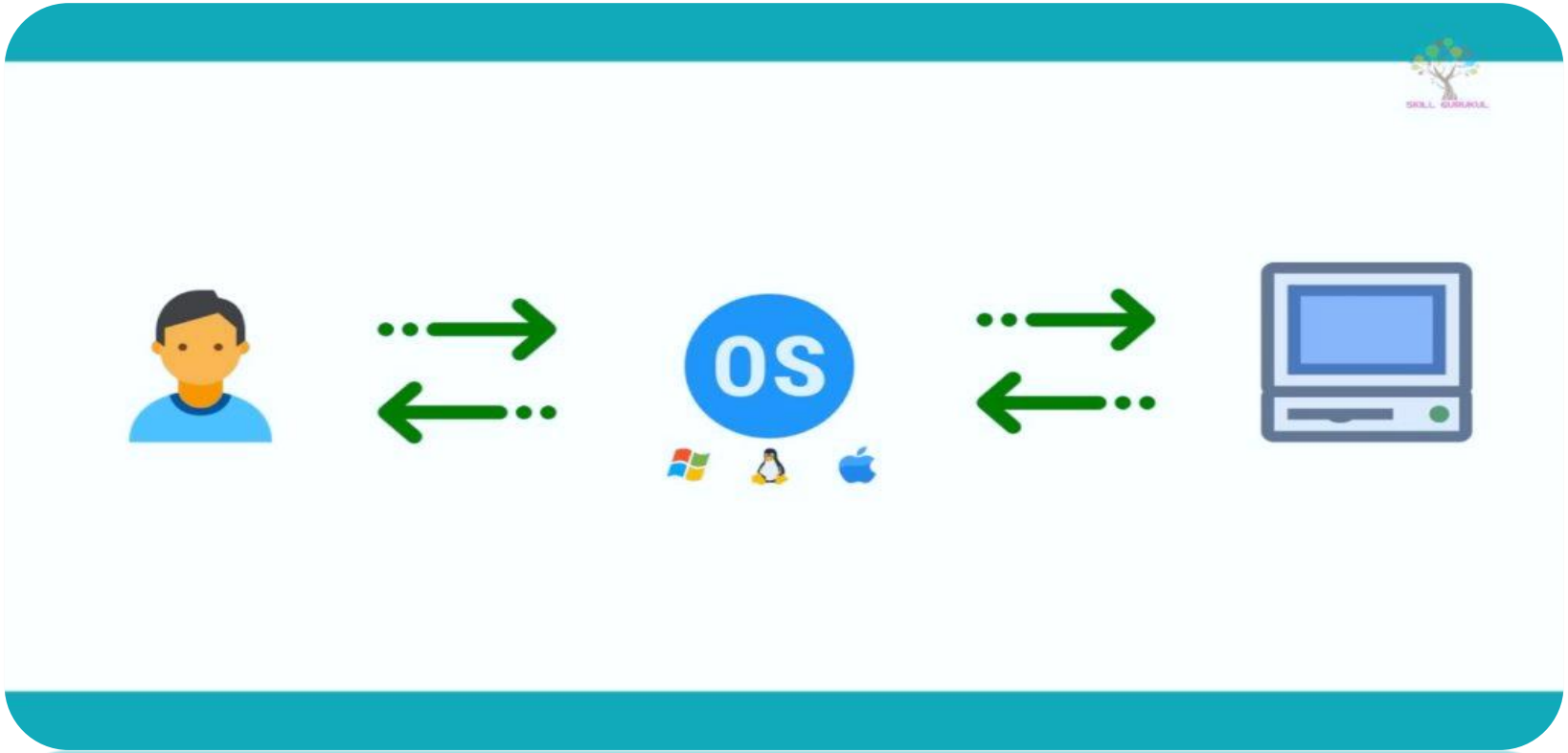
By: Foram Chovatiya

What is Operating System?

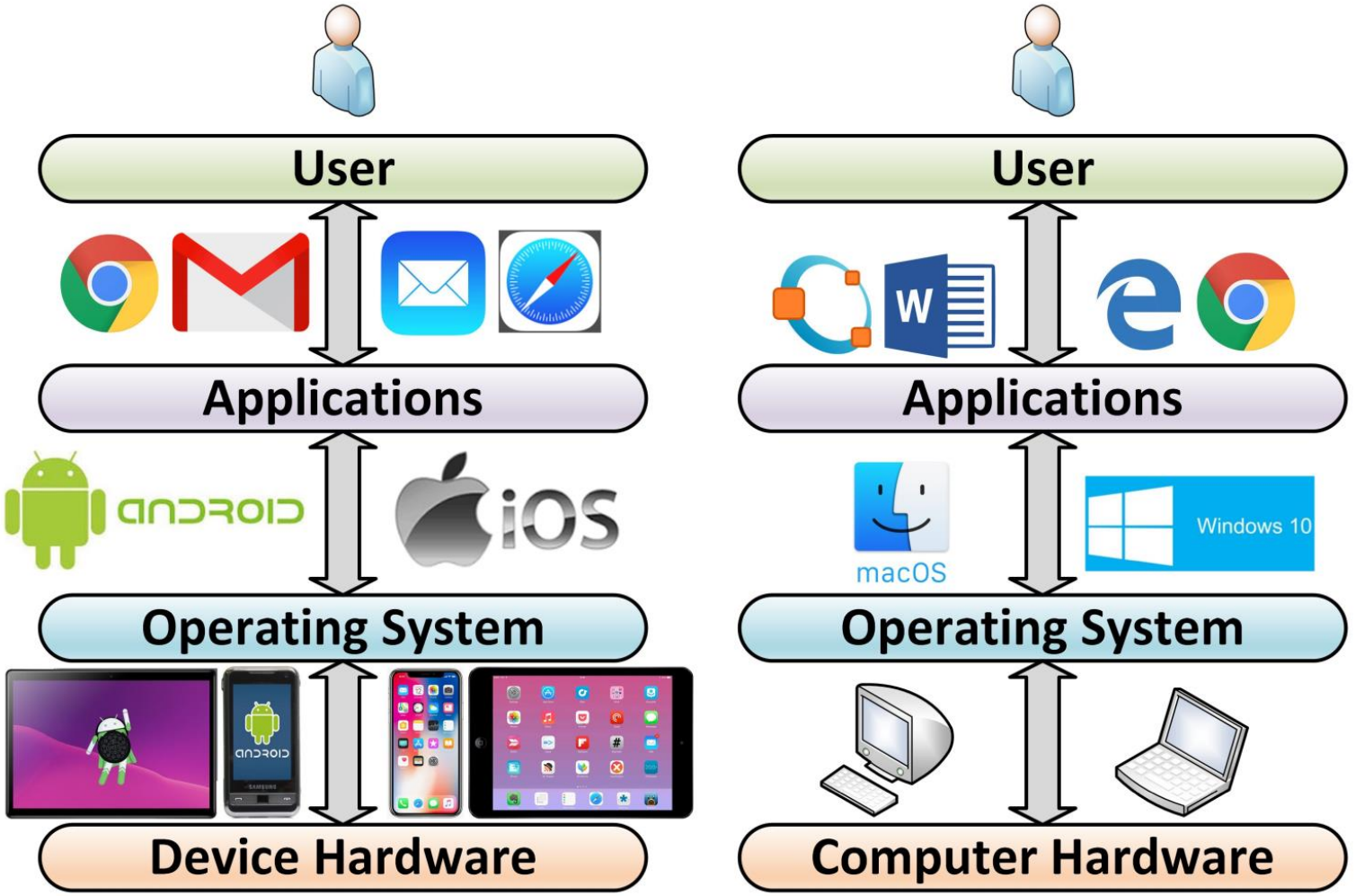
- An Operating System, or OS, is low-level software or System Software that enables a user and higher-level application software to interact with a computer's hardware, data and other programs stored on the computer.

Operating System

- Generally it acts as a bridge between user and Software.



Operating System



- So ultimately it Hide Hardware Complexity, provides Resource Management , Provide Isolation and Protection.

Purpose of Operating System

- To provide an environment for a computer user to execute programs on computer hardware in a convenient and efficient manner.
- To allocate the separate resources of the computer as needed to solve the problem given. The allocation process should be as fair and efficient as possible.
- **As a control program it serves two major functions:**

  (1) supervision of the execution of user programs to prevent errors and improper use of the computer, and

  (2) management of the operation and control of I/O devices.

Examples of Operating System

Operating System Concept

(1) Processes:

- A key concept in all operating systems is the process.
- A process is basically a program in execution.
- A process is fundamentally a container that holds all the information needed to run a program.

(2) Address Spaces

- An address space is a range of valid addresses in memory that are available for a program or process.
- That is, it is the memory that a program or process can access.
- The memory can be either physical or virtual and is used for executing instructions and storing data.

Operating System Concept

(3) Files

- A major function of the operating system is to hide the anomaly of the disks and other I/O devices and present the programmer with a nice, clean abstract model of device-independent files.
- System calls (A system call is a request for service that a program makes) are obviously needed to create files, remove files, read files, and write files.
- To provide a place to keep files, most PC operating systems have the concept of a directory as a way of grouping files together.
- A student, for example, might have one directory for each course he is taking, another directory for his electronic mail, and still another directory for his World Wide Web home page.

Operating System Concept

(4) Input/Output:

- All computers have physical devices for acquiring input and producing output
- Consequently, every operating system has an I/O subsystem for managing its I/O devices.
- Some of the I/O software is device independent, Other parts of it, such as device drivers, are specific to particular I/O devices.

(5) Protection:

- Computers contain large amounts of information that users often want to protect and keep confidential.
- It is up to the operating system to manage the system security so that files, for example, are accessible only to authorized users.

Function of OS

1) Security
2) Control over system performance
3) Job accounting
4) Error detecting aids
5) Coordination between other software and users
6) Memory Management
7) Processor Management
8) Device Management
9) File Management
10) Monitoring activities

Function of OS

1) 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.

2) Control over system performance:

- Monitors overall system health to help improve performance. records the response time between service requests and system response to having a complete view of the system health.
- This can help improve performance by providing important information needed to troubleshoot problems.

Function of OS

3) Job accounting:

- Operating system Keeps track of time and resources used by various tasks and users, this information can be used to track resource usage for a particular user or group of users.

- **4) Error detecting aids:**
- The operating system constantly monitors the system to detect errors and avoid the malfunctioning of a computer system.

Function of OS

5) Coordination between other software and users:

- Operating systems also coordinate and assign interpreters, compilers, assemblers, and other software to the various users of the computer systems.

6) Memory Management:

- The operating system manages the Primary Memory or Main Memory.
- For a program to be executed, it should be first loaded in the main memory.

An Operating System performs the following activities for memory management:

1) It keeps track of primary memory, i.e., which bytes of memory are used by which user program.
2) The memory addresses that have already been allocated and the memory addresses of the memory that has not yet been used.
3) It Allocates the memory to a process when the process requests it and deallocates the memory when the process has terminated or is performing an I/O operation.

Function of OS

- **7) Processor Management:**
- In a multi-programming environment, the OS decides the order in which processes have access to the processor, and how much processing time each process has.
- This function of OS is called process scheduling. These programs are called Special Control Program.

- **An Operating System performs the following activities for processor management.**
  1) Keeps track of the status of processes.
  2) The program which performs this task is known as a traffic controller.
  3) Allocates the CPU that is a processor to a process.
  4) De-allocates processor when a process is no more required.

Function of OS

8) Device Management:

- An OS manages device communication via their respective drivers.
- It performs the following activities for device management. The operating system decides which device to use for which task.
    1) Keeps track of all devices connected to the system.
    2) Decides which process gets access to a certain device and for how long.
    3) Allocates devices in an effective and efficient way. Deallocates devices when they are no longer required.

Operating System Services

- ✓ User Interface Program
- ✓ Memory Management
- ✓ Process Management (Scheduling)
- ✓ Device Management
- ✓ Execution I/O Operations
- ✓ File System Management
- ✓ System Performance
- ✓ Error Detection
- ✓ Job Accounting
- ✓ Protection and Security

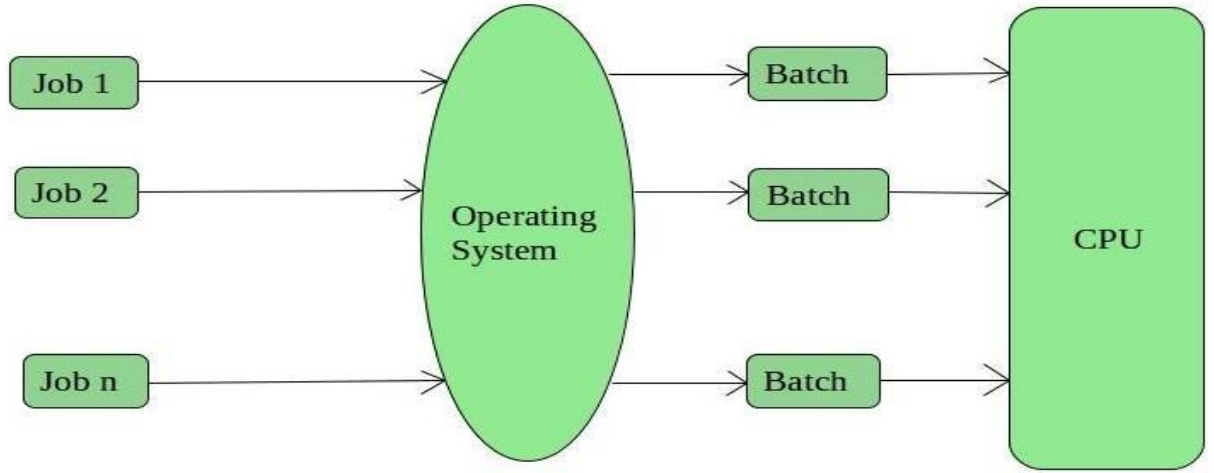
Types of OS

There are many types of Operating System which are listed below:

- Batch Operating System
- Multiprogramming Operating System
- Multiprocessing Operating System
- Real Time Operating System
- Multitasking/Time Sharing OS
- Distributed OS
- Network OS

Batch Operating System

- The Jobs were executed in batches. People were used to have a single computer which was called mainframe.
- In Batch operating system, access is given to more than one person; they submit their respective jobs to the system for the execution.
- The system put all of the jobs in a queue on the basis of **first come first serve(FCFS)** and then executes the jobs one by one.
- The users collect their respective output when all the jobs get executed.



Batch Operating System

Disadvantages of Batch OS:

(1) Starvation:

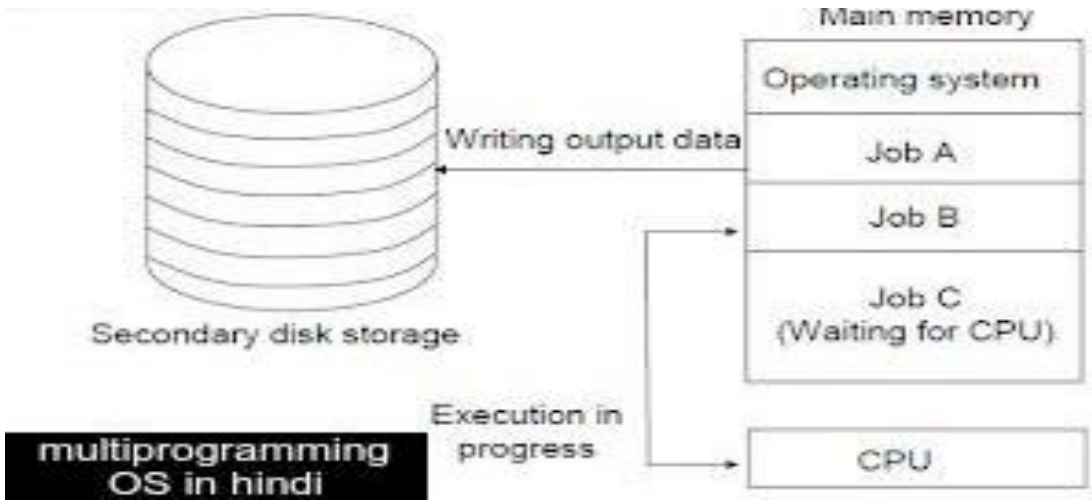
- Batch processing suffers from starvation.
- If there are five jobs J1, J2, J3, J4, J4 and J5 present in the batch.
- If the execution time of J1 is very high then other four jobs will never be going to get executed or they will have to wait for a very high time.
- **Starvation** is a problem with resource management where a process runs out of resources in the OS because those resources are being utilised by other processes. This issue primarily arises in a priority-based scheduling system where requests with high priority are processed first and those with low priority take longer to process.

(2) Not Interactive:

- Batch Processing is not suitable for the jobs which are dependent on the user's input.
- If a job requires the input of two numbers from the console then it will never be going to get it in the batch processing scenario since the user is not present at the time of execution.

Multiprogra-mming Operating System

- Multiprogramming is an extension to the batch processing where the CPU is kept always busy.
- A multiprogramming operating system may run many programs on a single processor computer.
- If one program must wait for an input/output transfer in a multiprogramming operating system, the other programs are ready to use the CPU.
- As a result, various jobs may share CPU time. However, the execution of their jobs is not defined to be at the same time period.



Multiprogra-mming Operating System

Types of the Multiprogramming Operating System:

- There are mainly two types of multiprogramming operating systems. These are as follows:

1) Multitasking Operating System
2) Multiuser Operating System

Multiprogramming Operating System

Advantages:

(1) It provides less response time.

(2) It may help to run various jobs in a single application simultaneously.

(3) It helps to optimize the total job throughput of the computer.

(4) Various users may use the multiprogramming system at once.

(5) Short-time jobs are done quickly in comparison to long-time jobs.

(6) It may help to improve turnaround time for short-time tasks.

(7) It helps in improving CPU utilization and never gets idle.

The resources are utilized smartly.

Multiprogramming Operating System

Disadvantages:

(1) It is highly complicated

(2) The CPU scheduling is required.

(3) Memory management is needed in the operating system because all types of tasks are stored in the main memory.

(4) The harder task is to handle all processes and tasks

Multiprocessing Operating System

- When more than one program or task executes at the same time it is called multiprocessing.
- Multiprocessing use more than one processor.
- In multiprocessing O.S. multiple processes are executed by multiple processor at a time.
- Here the CPU scheduling is required every time, O.S. will schedule the CPU.
- One CPU will execute one program at a time.
- Parallel computing is achieved.

Multiprocessing Operating System

Advantage:

1) Increased efficiency
2) Faster response time
3) Ability to Assign priorities of jobs
4) Multiple independent job can be executed in parallel
5) Single larger job can be partitioned into multiple parallel task

Disadvantages:

1) Large Memory
2) Memory Protection
3) Program status Preservation
4) Deadlock

Real Time Operating System

- Real-time operating systems (RTOS) are used in environments where a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within certain deadlines.
- such applications are industrial control, telephone switching equipment, flight control, and real-time simulations.
- In Real Time systems, each job carries a certain deadline within which the Job is supposed to be completed, otherwise the huge loss will be there or even if the result is produced then it will be completely useless.
- Military applications, Radar application, Satellite communication

Real Time Operating System

Advantage:

1) Increased efficiency
2) Deterministic behavior
3) Scalable
4) Portable
5) Task Scheduling and shifting

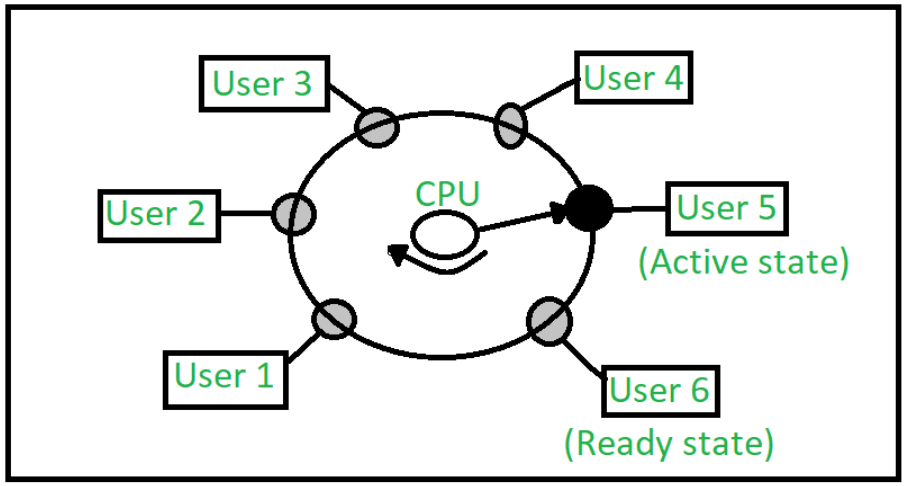
Disadvantages:

1) Large memory
2) Cost is more
3) Law support of variety of peripherals
4) Multiple Users – Multiple Mode

Multitasking/ Time Sharing OS

- A time shared operating system allows multiple users to share computers simultaneously.
- Each action or order at a time the shared system becomes smaller, so only a little CPU time is required for each user.
- As the system rapidly switches from one user to another, each user is given the impression that the entire computer system is dedicated to its use, although it is being shared among multiple users.
- A time shared operating system uses CPU scheduling and multi-programming to provide each with a small portion of a shared computer at once.
- Each user has at least one separate program in memory.

Multitasking/ Time Sharing OS



In above figure the user 5 is active state but user 1, user 2, user 3, and user 4 are in waiting state whereas user 6 is in ready state.

(1) Active State –

The user's program is under the control of CPU. Only one program is available in this state.

(2) Ready State –

The user program is ready to execute but it is waiting for it's turn to get the CPU. More than one user can be in ready state at a time.

(3) Waiting State –

The user's program is waiting for some input/output operation. More than one user can be in a waiting state at a time.

Multitasking/ Time Sharing OS

Advantages of Timesharing operating systems are as follows

- Each task gets an equal opportunity.
- Less chances of duplication of software.
- CPU idle time can be reduced.

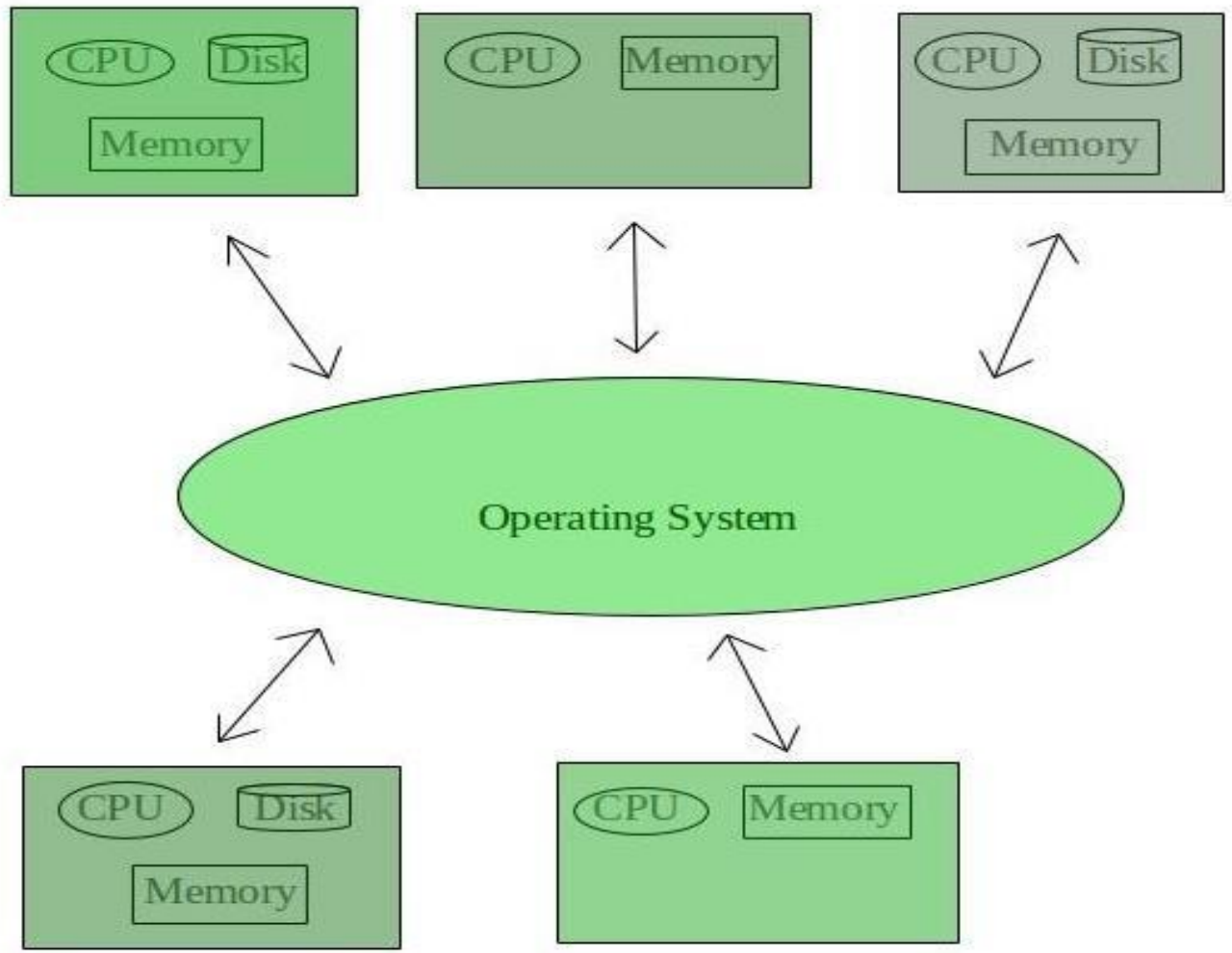
Disadvantages of Time-sharing operating systems are as follows

- Reliability problem.
- One must have to take of security and integrity of user programs and data.
- Data communication problem.

Distributed Operating System

- In DOS, Multiple computer are connected into close communication to share the files and resources.
- User don’t know where its programming is executing.
- The major benefit of working with these types of operating system is that it is always possible that one user can access the files or software which are not actually present on his system but on some other system connected within this network

Distributed Operating System



Distributed Operating System

Advantage:

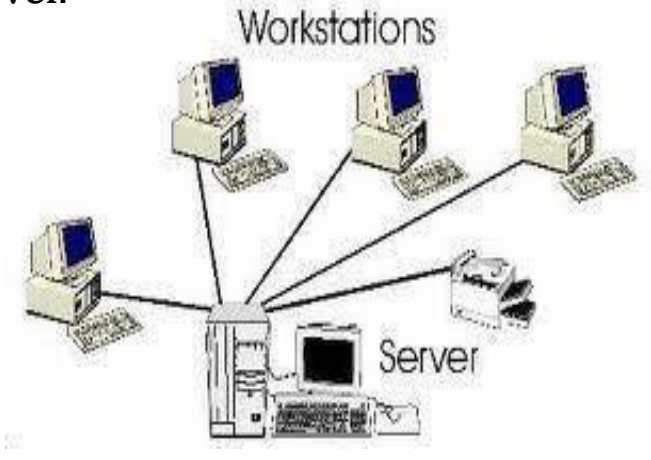
1) Communication and Resource sharing is possible
2) Scalability
3) Reliable
4) Potential for incremental Growth

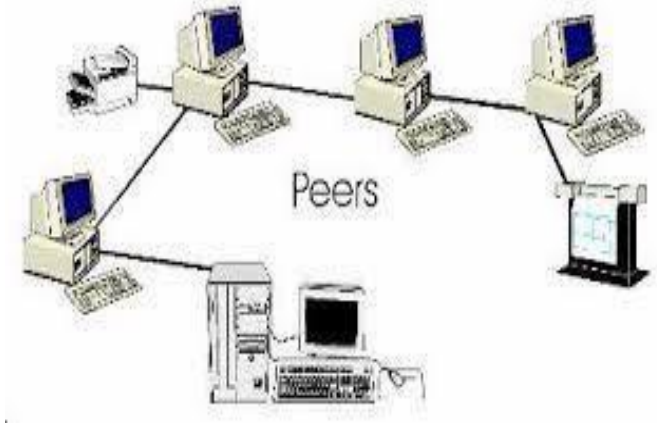
Disadvantages:

1) Network connectivity must be required
2) Security and privacy

Network OS (NOS)

- OS share the activities of multiple computer across the network.
- NOS work as director for sharing this network.
- Allow shared file and printer access among multiple computers in a network
- Example Windows Server 2008, MAC OS X
- There are two types of NOS
  - Peer to Peer
  - Client / Server.





Network OS (NOS)

Functions of Network OS

- Creating and managing user accounts on the network.
- Controlling access of resources on network.
- Provide communication services between the devices on the network.
- Monitor and troubleshoot network.
- Configuring and Managing the resources on the network.

Network OS (NOS)

Advantage:

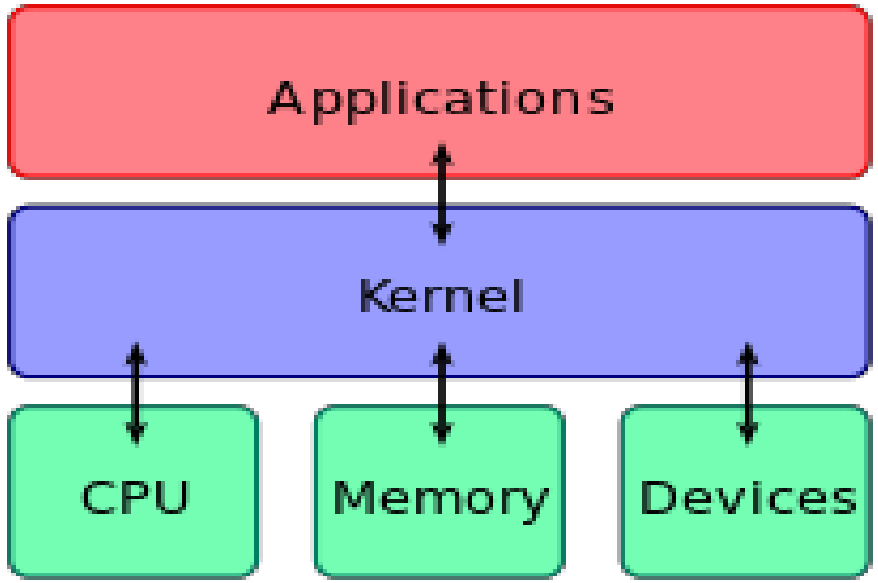
1) Centralized Server are highly stable
2) Security is server managed
3) Upgrade to new technologies from the server is very easy to all client system

Disadvantages:

1) Cost is very high of buying and running a server
2) Dependency on central location server
3) Regular maintenance and updates are required.

Kernel

- Kernel is central component of an operating system that manages operations of computer and hardware.
- It basically manages operations of memory and CPU time.
- It is core component of an operating system. Kernel acts as a bridge between applications and data processing performed at hardware level using inter-process communication and system calls.



Kernel

- Kernel loads first into memory when an operating system is loaded and remains into memory until operating system is shut down again.
- It is responsible for various tasks such as disk management, task management, and memory management.
- It decides which process should be allocated to processor to execute and which process should be kept in main memory to execute.

Kernel

Objectives of Kernel:

1) To establish communication between user level application and hardware.
2) To decide state of incoming processes.
3) To control disk management.
4) To control memory management.
5) To control task management.

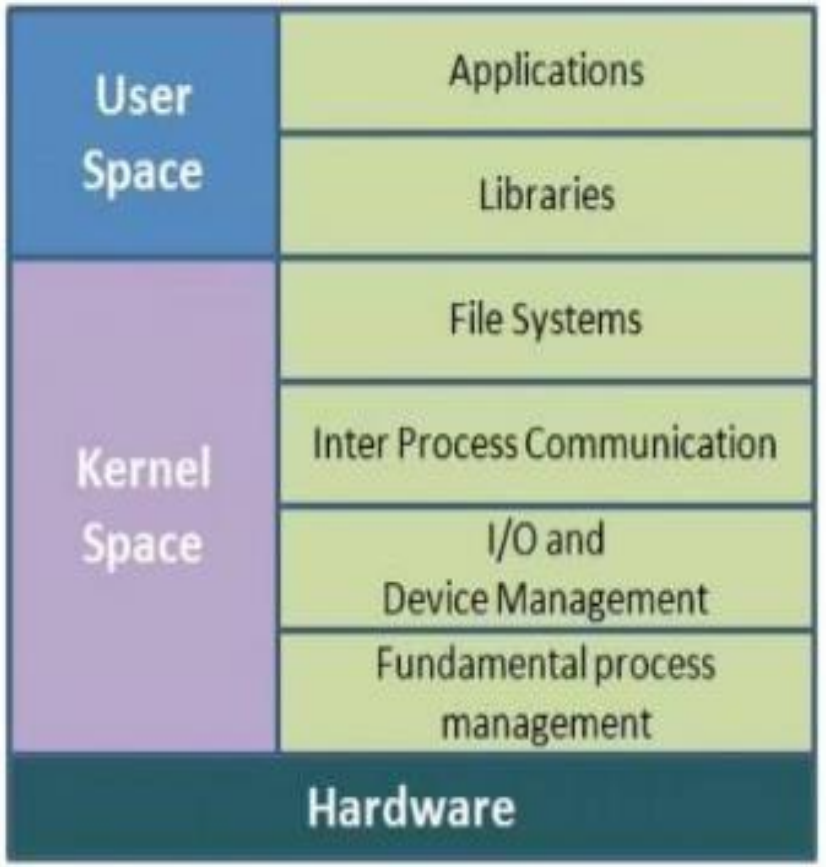
Types of Kernel

- **There are basically two types of kernel used in Operating System.**

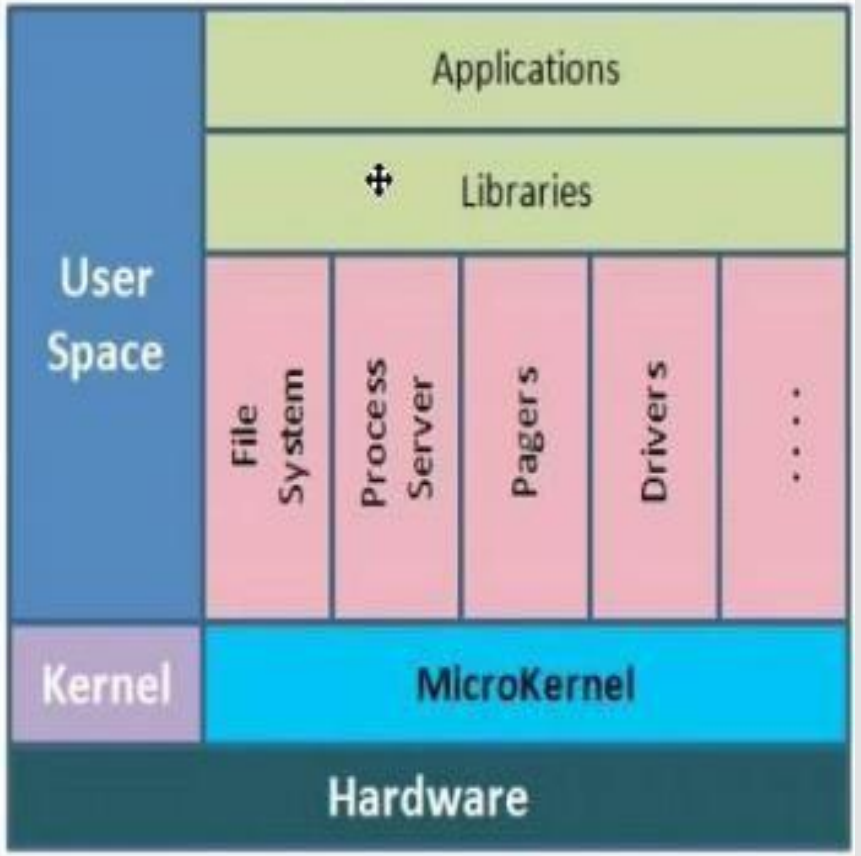
1) **Monolithic Kernel**
2) **Microkernel**

Types of Kernel

- **There are basically two types of kernel used in Operating System.**



Monolithic



Microkernel

Monolithic Kernel

- The entire O.S. is placed inside the kernel.
- It runs as single large process
- As all the services are placed inside the kernel, they have a single address space, they can directly communicate with each other using function calls.
- It is bigger in size
- It is easy to implement/code
- Performance is high (As kernel can invoke any function directly as everything is placed in the kernel)
- Less Secure (If one service fails, entire system crashes)

Monolithic Kernel

Advantage :

- It has good performance.

Disadvantage :

- It has dependencies between system component and lines of code in millions.

- **Example:**

  Unix, Linux, Open VMS, XTS-400 etc.

Microkernel Kernel

- The microkernel is one of the kernel's classifications.
- Being a kernel, it handles all system resources.
- On the other hand, the user and kernel services in a microkernel are implemented in distinct address spaces.
- User services are kept in user address space, while kernel services are kept in kernel address space.
- It aids to reduce the kernel and OS's size.

Microkernel Kernel

Advantage :

- It is more stable.
- Microkernel architecture is compact and isolated, so it may perform better.
- Server failure is treated the same as any other user program failure.
- It adds new features without recompiling.

Disadvantage :

- There are lots of system calls and context switches.
- In a microkernel system, providing services are more costly than in a traditional monolithic system.
- The performance of a microkernel system might be indifferent and cause issues.

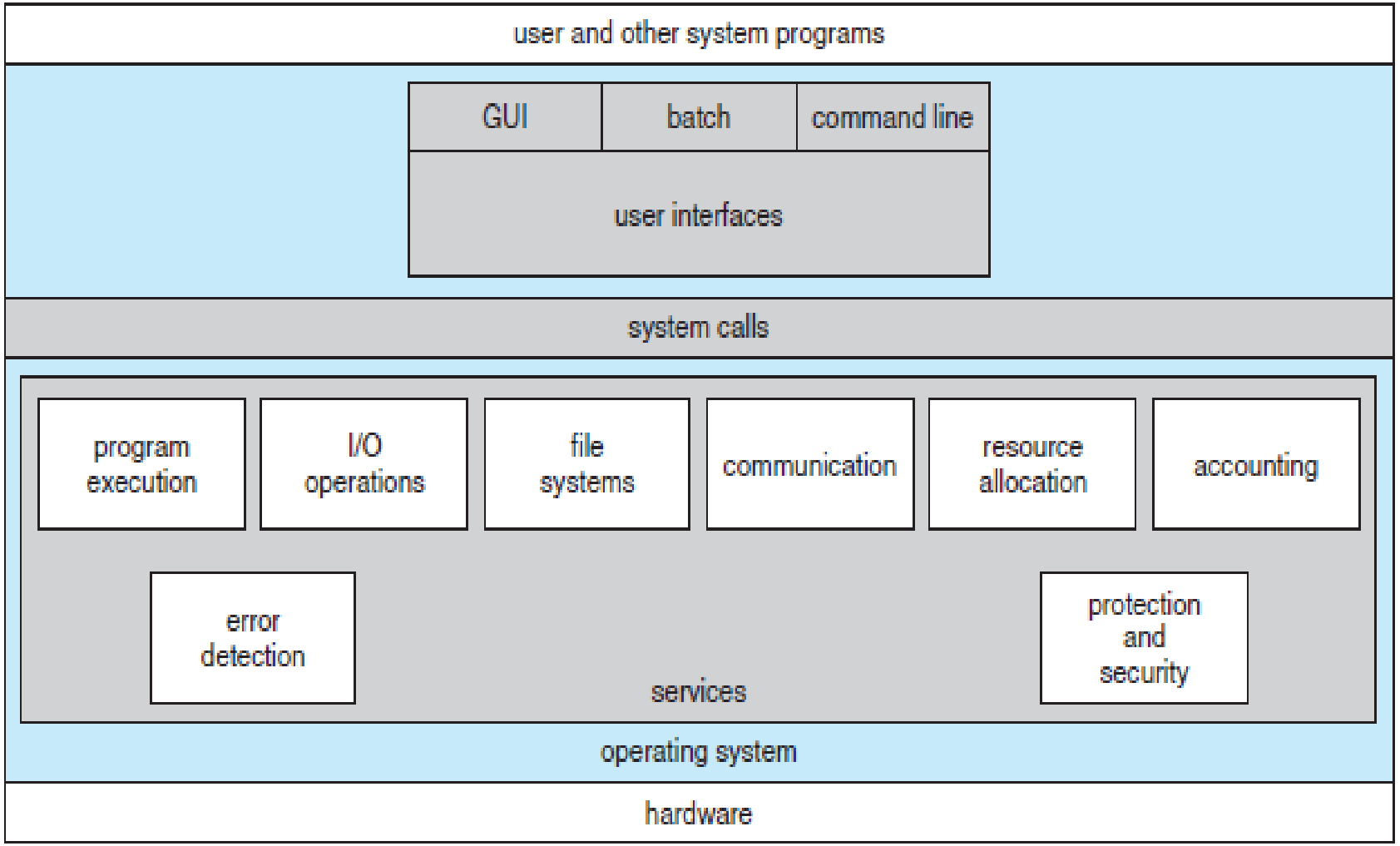
Example :

Mach, L4, AmigaOS, Minix, K42 etc.

Difference between Monolithic and Microkernel

Basis for Comparison	Microkernel	Monolithic Kernel
Size	Microkernel is smaller in size	It is larger than microkernel
Execution	Slow Execution	Fast Execution
Extendible	It is easily extendible	It is hard to extend
Security	If a service crashes, it does effects on working on the microkernel	If a service crashes, the whole system crashes in monolithic kernel.
Code	To write a microkernel more code is required	To write a monolithic kernel less code is required
Example	QNX, Symbian, L4Linux etc.	Linux,BSDs(FreeBSD,OpenBSD,NetBSD)etc.

Operating System Architecture



Operating System Architecture

- An operating system provides an environment for the execution of programs.
- It provides certain services to programs. These operating system services are provided for the convenience of the programmer.
- Figure shows one view of the various operating-system services and how they interrelate.
- **User interface.** Almost all operating systems have a user interface (UI).
- This interface can take several forms.

Operating System Architecture

Command-line interface (CLI):

- Which uses text commands and a method for entering them (say, a keyboard for typing in commands in a specific format with specific options.

Batch interface :

- Batch interfaces are non-interactive user interfaces, where the user specifies all the details of the batch job in advance to batch processing, and receives the output when all the processing is done.
- **Graphical user interface (GUI) :**
- Here, the interface is a window system with a pointing device to direct I/O, choose from menus, and make selections and a keyboard to enter text.

Operating System Architecture

System Call

- When a program in user mode requires access to RAM or Hardware a resource, it must ask the kernel to provide access to that resource.
- This is done via something called a system call.

Context switch

- When a program makes a system call, the mode is switched from user mode to kernel mode.
- This is called a **context switch.**

Operating System Architecture

Program Execution

The system must be able to load a program into memory and to run that program.

The program must be able to end its execution, either normally or abnormally (indicating error).

I/O operations

A running program may require I/O, which may involve a file or an I/O device.

For specific devices, special functions may be desired

- (such as recording to a CD or DVD drive or blinking a display screen).

Operating System Architecture

- **File-system Manipulation:**
- Programs need to read and write files and directories.
- They also need to create and delete them by name, search for a given file, and list file information.
- Finally, some operating systems include permissions-management to allow or deny access to files or directories based on file ownership.

Operating System Architecture

Communications:

- There are many circumstances in which one process needs to exchange information with another process.
- Such communication may occur between processes that are executing on the same computer or between processes that are executing on different computer systems tied together by a computer network.
- Communications may be implemented via **shared memory**, in which two or more processes read and write to a shared section of memory

Operating System Architecture

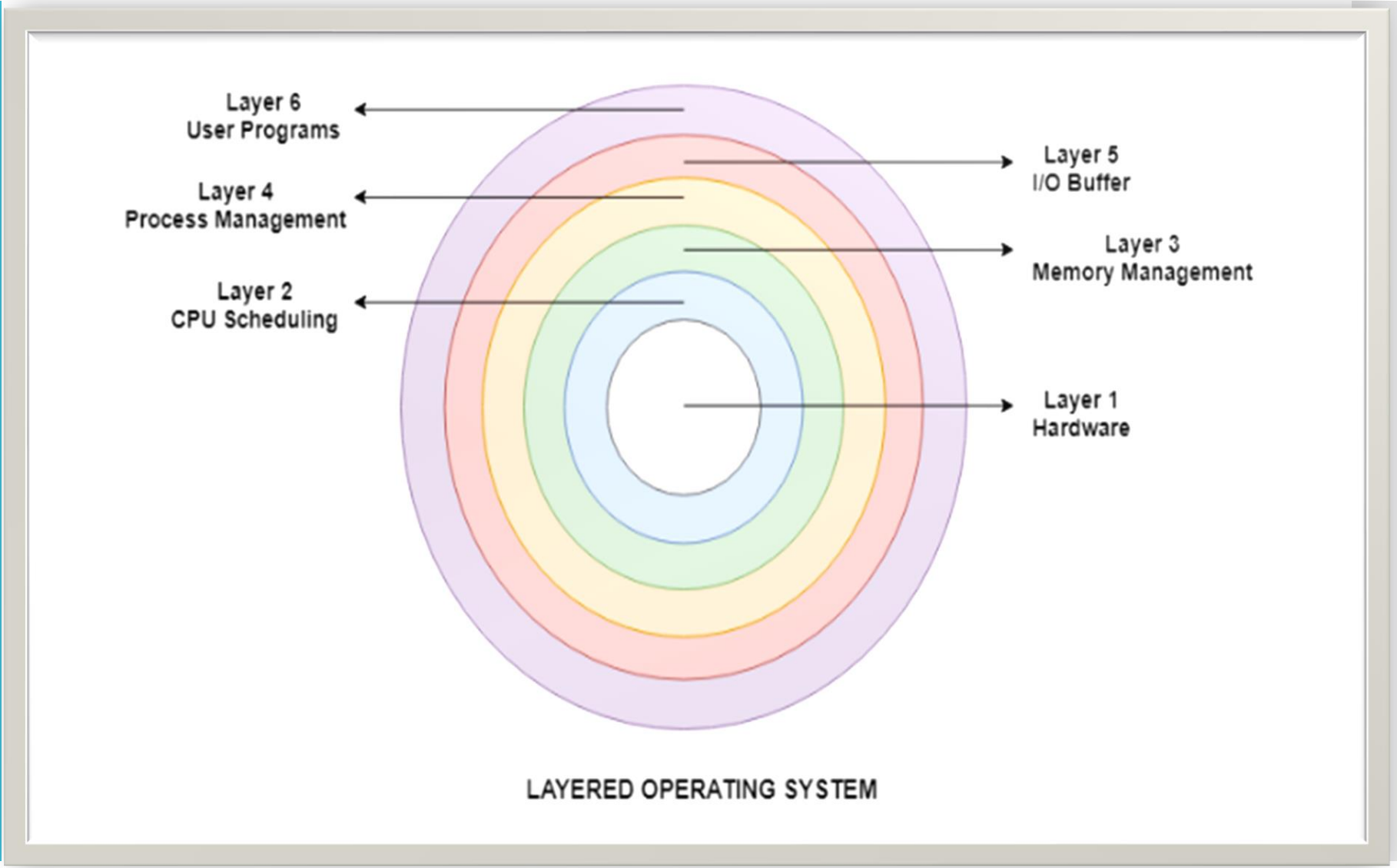
- **Resource Allocation**
- When there are multiple users or multiple jobs running at the same time, resources must be allocated to each of them.
- The operating system manages many different types of resources. Some (such as CPU cycles, main memory, and file storage) may have special allocation code, whereas others (such as I/O devices) may have much more general request and release code.
- **Accounting.**
- We want to keep track of which users use how much and what kinds of computer resources.
- This record keeping may be used for accounting (so that users can be billed) or simply for accumulating usage statistics.

Operating System Architecture

Error Detection

- The operating system needs to be detecting and correcting errors constantly.
- Errors may occur in the CPU and memory hardware (such as a memory error or a power failure) , a connection failure on a network, in an I/O devices.
- For each type of error, the operating system should take the appropriate action to ensure correct and consistent computing. Sometimes, it has no choice but to halt the system

Operating System Architecture Layered Approach



Operating System Architecture Layered Approach

There are six layers in the layered operating system.

- A diagram demonstrating these layers is as follows:

1) Hardware (Layer 1)
2) CPU Scheduling (Layer 2)
3) Memory Management (Layer 3)
4) Process Management (Layer 4)
5) I/O Buffer (Layer 5)
6) User Programs (Layer 6)

Operating System Architecture Layered Approach

(1) Hardware

- This layer interacts with the system hardware and coordinates with all the peripheral devices used such as printer, mouse, keyboard, scanner etc. The hardware layer is the lowest layer in the layered operating system architecture.

(2) CPU Scheduling

- This layer deals with scheduling the processes for the CPU. There are many scheduling queues that are used to handle processes. When the processes enter the system, they are put into the job queue. The processes that are ready to execute in the main memory are kept in the ready queue.

Operating System Architecture Layered Approach

(3) Memory Management

- Memory management deals with memory and the moving of processes from disk to primary memory for execution and back again. This is handled by the third layer of the operating system.

(4) Process Management

- This layer is responsible for managing the processes i.e assigning the processor to a process at a time. This is known as process scheduling. The different algorithms used for process scheduling are FCFS (first come first served), SJF (shortest job first), priority scheduling, round-robin scheduling etc.

Operating System Architecture Layered Approach

(5) I/O Buffer

- I/O devices are very important in the computer systems. They provide users with the means of interacting with the system. This layer handles the buffers for the I/O devices and makes sure that they work correctly.

(6) User Programs

- This is the highest layer in the layered operating system. This layer deals with the many user programs and applications that run in an operating system such as word processors, games, browsers etc.

Thank you…