Operating System

Unit-I

By
Balaram Das
School of Engineering & Technology
GIET University, Gunupur

Unit-01	Unit-02
• Introduction to operating system • Evolution of operating system	• Process & CPU scheduling
Unit-03	**Unit-04**
• Dead locks • Memory management and virtual memory	• File system interface • File system implementation • Mass storage structure • Protection

UNIT:1 **(10 Hours)**

Operating System Introduction: Operating Systems Objectives and functions, Computer System Architecture, OS Structure, OS Operations, Evolution of Operating Systems - Simple Batch, Multi programmed, time shared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, Special - Purpose Systems, Operating System services, user OS
Interface, System Calls, Types of System Calls, System Programs, Opening System Design and Implementation, OS Structure, Virtual machines.

UNIT:2 **(12 Hours)**

Process and CPU Scheduling - Process concepts - The Process, Process State, Process Control Block, Threads, Process Scheduling - Scheduling Queues, Schedulers, Context Switch, Pre-emptive Scheduling, Dispatcher, Scheduling Criteria, Scheduling algorithms,Multiple-Processor Scheduling, Real-Time Scheduling, Thread scheduling, Case studies: Linux, Windows. Process Coordination - Process Synchronization, The Critical section Problem, Peterson's solution, Synchronization Hardware, Semaphores, and Classic Problems of Synchronization, Monitors, Case Studies: Linux, Windows.

UNIT:3 **(14 Hours)**

Memory Management and Virtual Memory - Logical & physical Address Space, Swapping, Contiguous Allocation, Paging, Structure of Page Table. Segmentation, Segmentation with Paging, Virtual Memory, Demand Paging, Performance of Demanding Paging, Page Replacement Algorithms, Allocation of Frames, Thrashing.

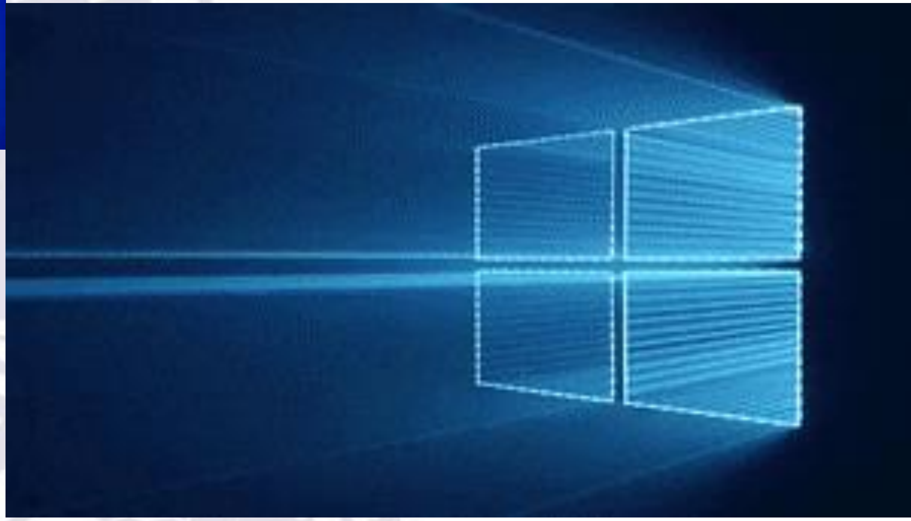
Deadlocks - System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection and Recovery from
Deadlock.

UNIT:4 **(14 Hours)**

File System Interface - The Concept of a File, Access methods, Directory Structure, File System Mounting, File Sharing, Protection, File System Implementation - File System Structure, File System Implementation, Allocation methods, Free-space Management, Directory Implementation, Efficiency and Performance. Mass Storage Structure - Overview of Mass Storage Structure, Disk Structure, Disk Scheduling, Disk Management, Swap space Management.

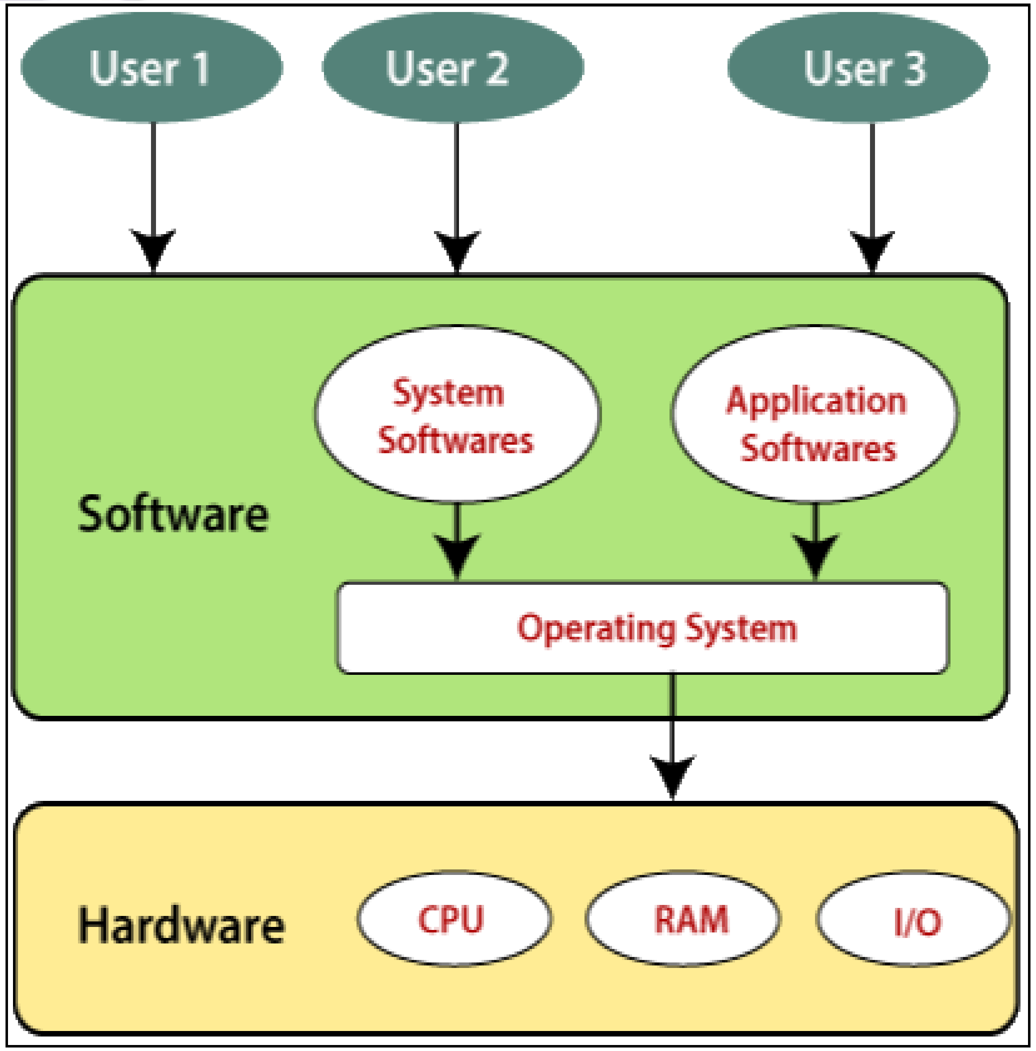
Protection - System Protection, Goals of Protection, Principles of Protection, Domain of Protection, Access Matrix, Implementation of Access Matrix, Access Control, Revocation of Access Rights, Capability-Based Systems, Language-Based Protection.

Different Operating Systems



Introduction

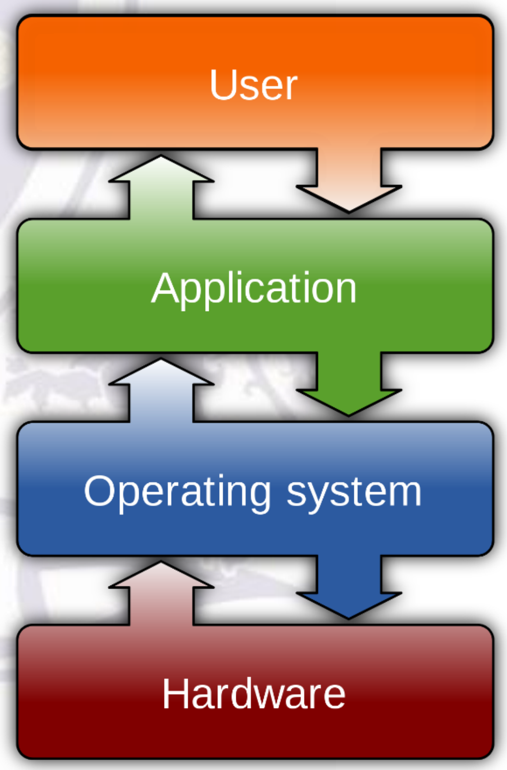
- **Operating System is a system software.**
- **Software is a set of instructions, data or programs used to operate computers and execute specific tasks.**
- **There are two types of software.**
- **(a) Application software**
- **(b) System software**



- **Application software are designed for some specific task.**
- **Ex: Media player, MS Word, Excel, PPT etc.**
- **System software are designed to handle the hardware of the system and give platform for application software so that they can execute their tasks.**
- **Ex: MS Windows, Apple iOS, Apple macOS, Google's Android, Linux OS etc.**

System software	Application software
System software is meant to manage the system resources. It serves as the platform to run application software.	Application software helps perform a specific set of functions for which they have been designed.
Developed in a low-level language (assembly language for example)	Developed in a high-level language such as Java, C++, .net and VB.
Automatically starts running once the system is turned on and stops when the system is shut down.	Runs as and when the user requests it.
A system cannot even start without system software	Application software is user specific and it is not needed to run the system on the whole.
Ex: Windows Operating System	Ex: MS Office, Photoshop and CorelDraw

- ***"Operating system acts as a system software that creates an interface between the user of the computer and computer hardware."***
- **Purpose of the operating system is to provide an environment where a user can execute programs in a convenient & efficient manner.**
- **It consists of applications and system software.**



Objective of the Operating System

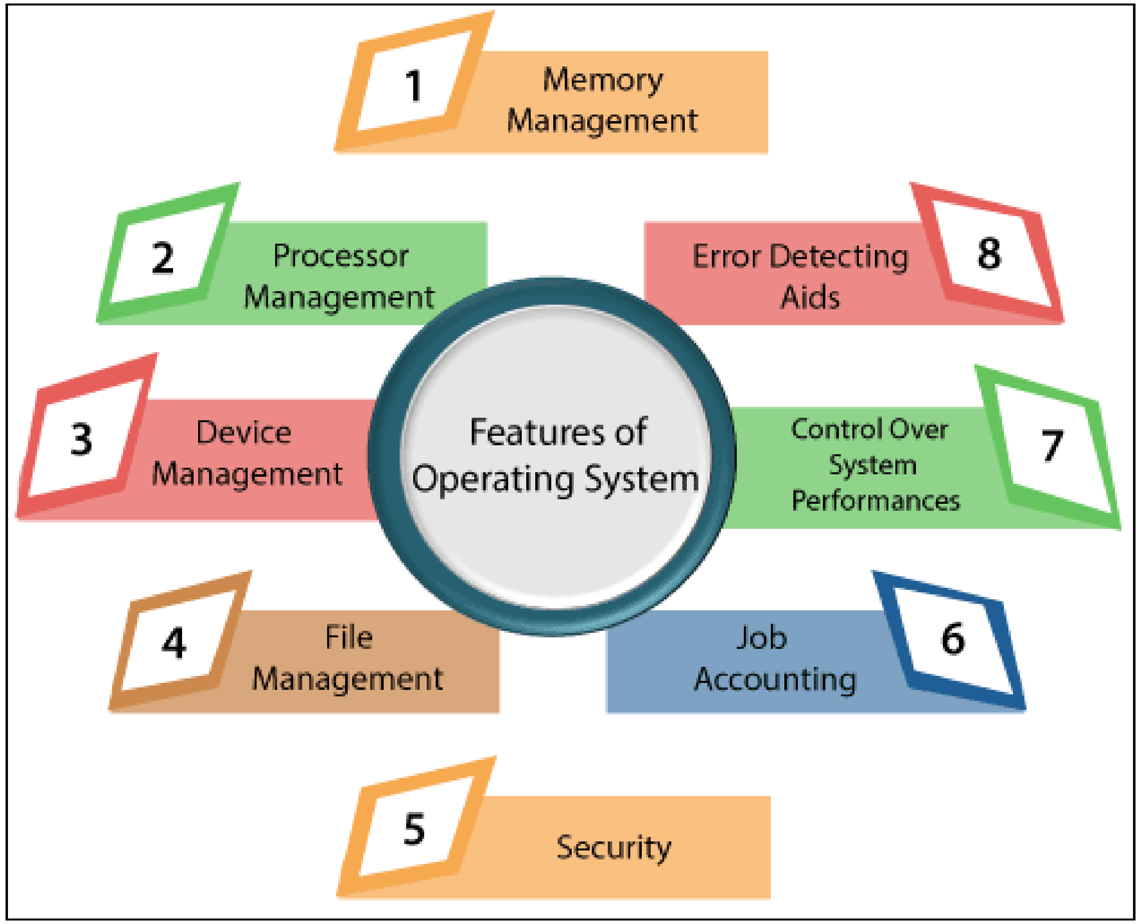
Objective of the Operating System

- Offers users an appropriate interface to use the computer system.
- Helps us to hide the information related to the hardware resources from the users.
- Provides the sharing of resources among users and programs in an efficient way.
- Helps us to manage the computer system resources.
- Offers convenient and impartial sharing of resources between the users and programs.

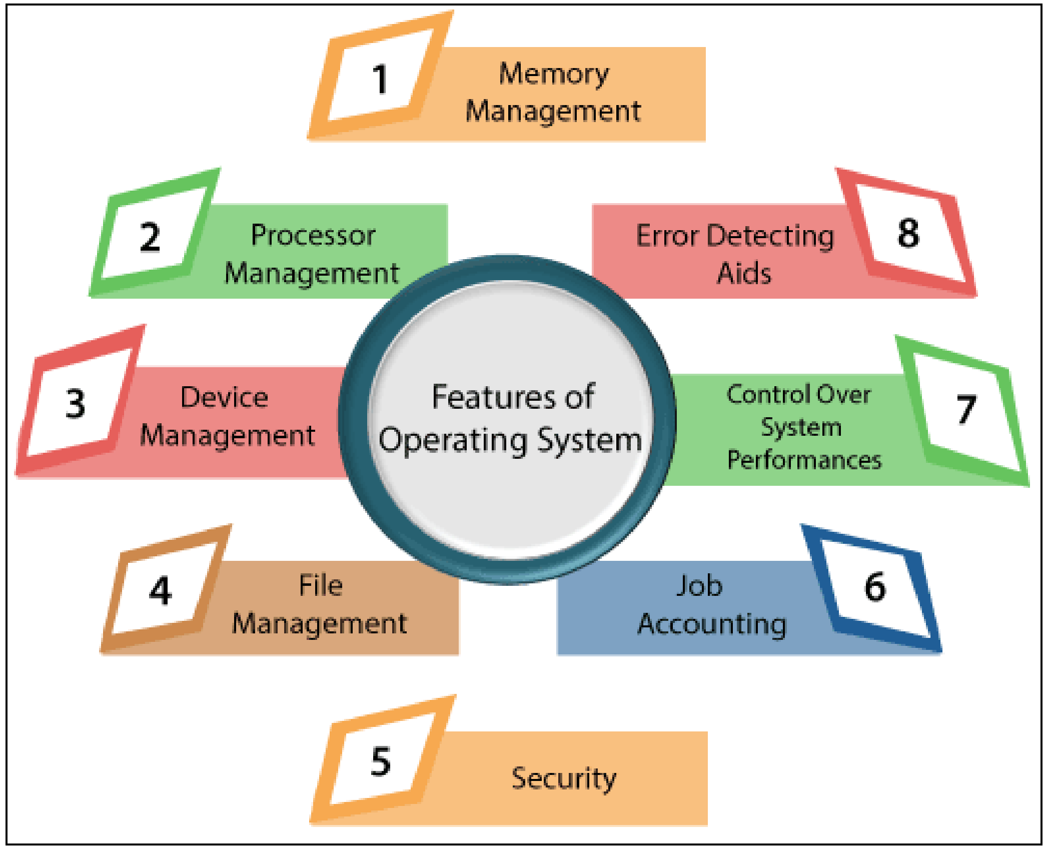
Functions of Operating System

Functions of Operating System

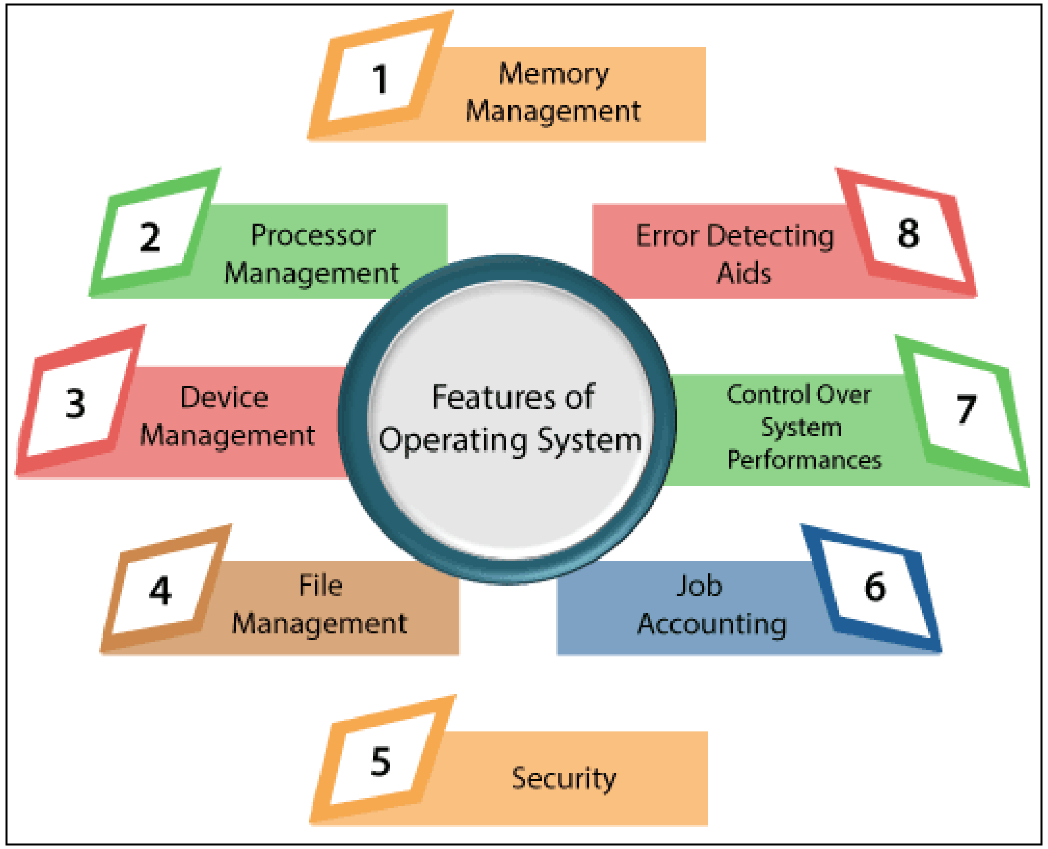
- **Memory management: –**
- **Keep track of the primary memory, means it keeps track of what part of memory is used, what part is not in use.**
- **Helps to allocate the memory when a process requests it.**
- **Processor management: –**
- **Allocate processors to the process when the process needs and deallocates the processor when the process does not require the processor.**



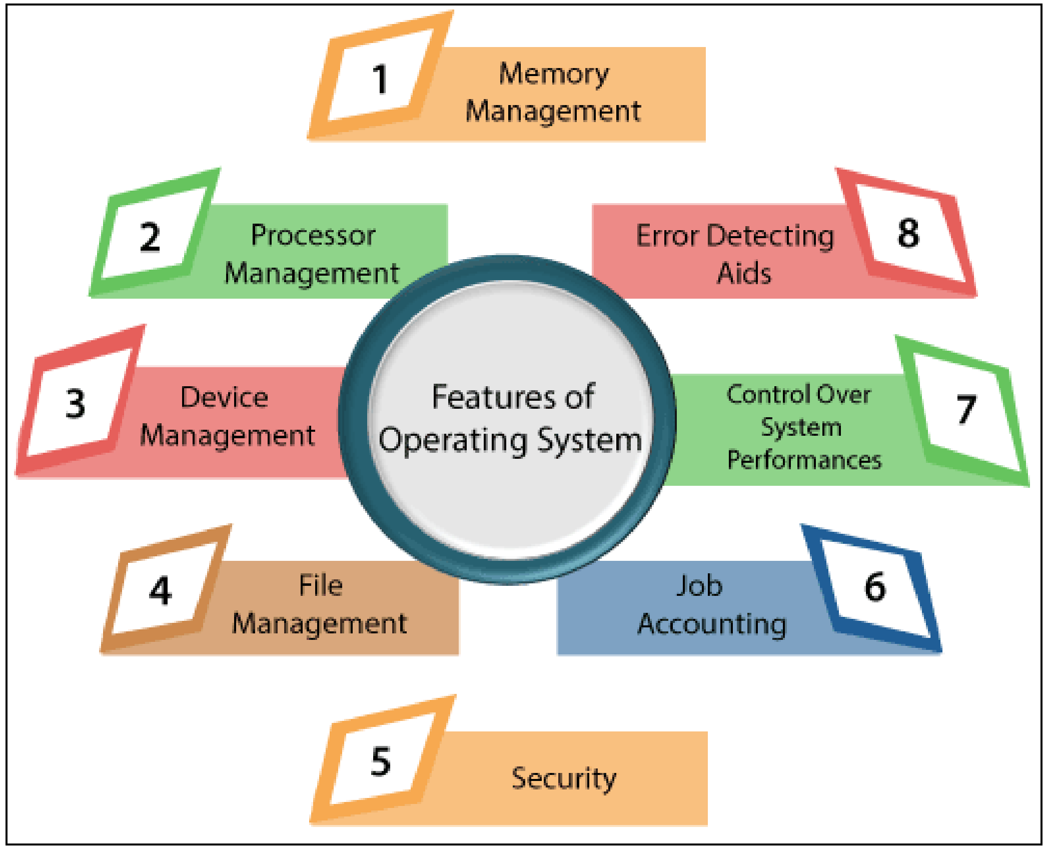
- **Device management: –**
- **The operating system also maintains the track of all the devices, which means an operating system decides which process should get the device, when and for how much time.**
- **File management: – The operating system provides the facility of file management.**



- **Security: –**
- **It prevents programs and data from unauthorized access by using passwords and some other techniques.**
- **Job accounting: –**
- **It also keeps track of resources and time used by different jobs.**



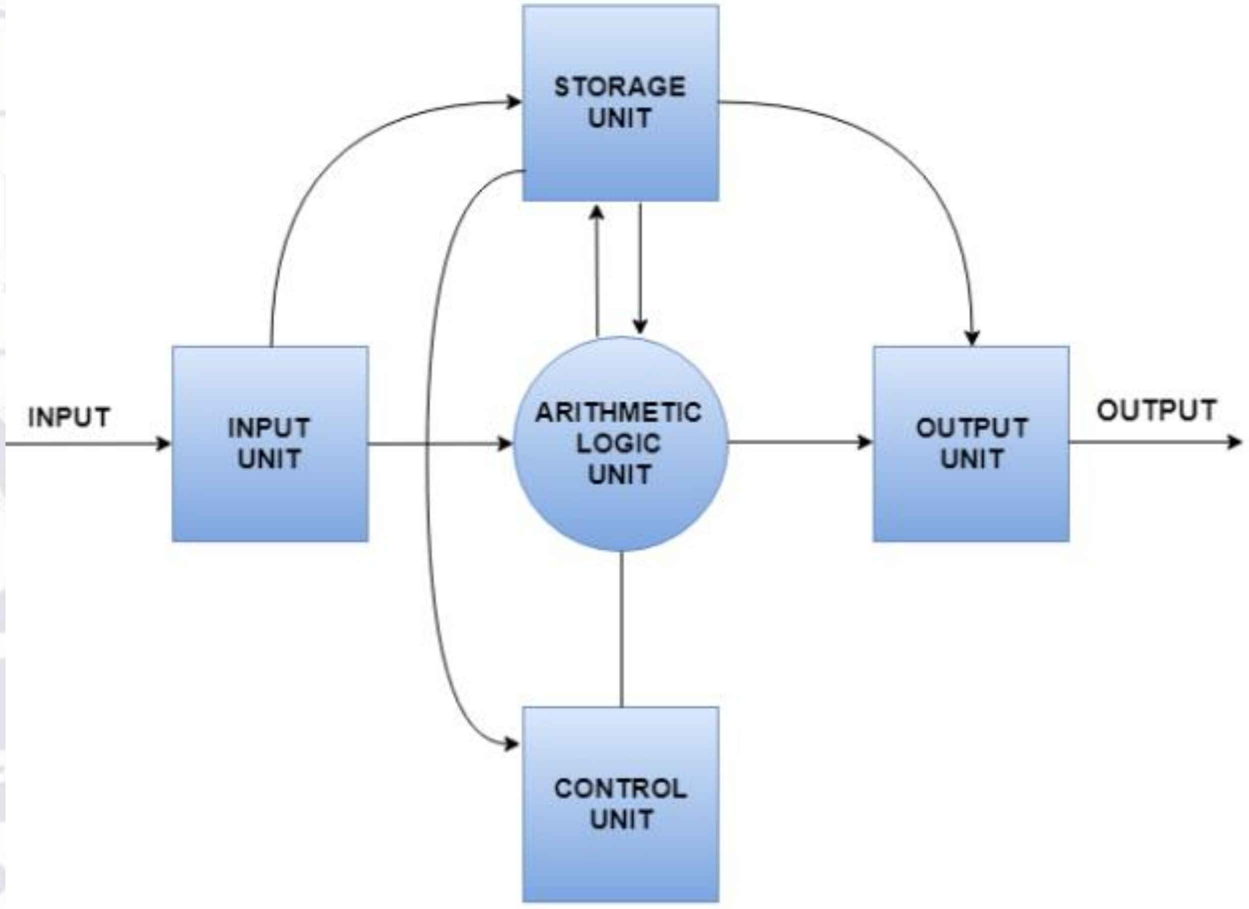
- **Control over system performance:-**
- **Monitors overall system health to help improve performance.**
- **Records the response time between service requests and system response to having a complete view of the system health.**
- **Error-detecting Aids: –**
- **It provides the facility of error detection by the production of dumps, traces, error messages, and by using different debugging and error detecting techniques.**



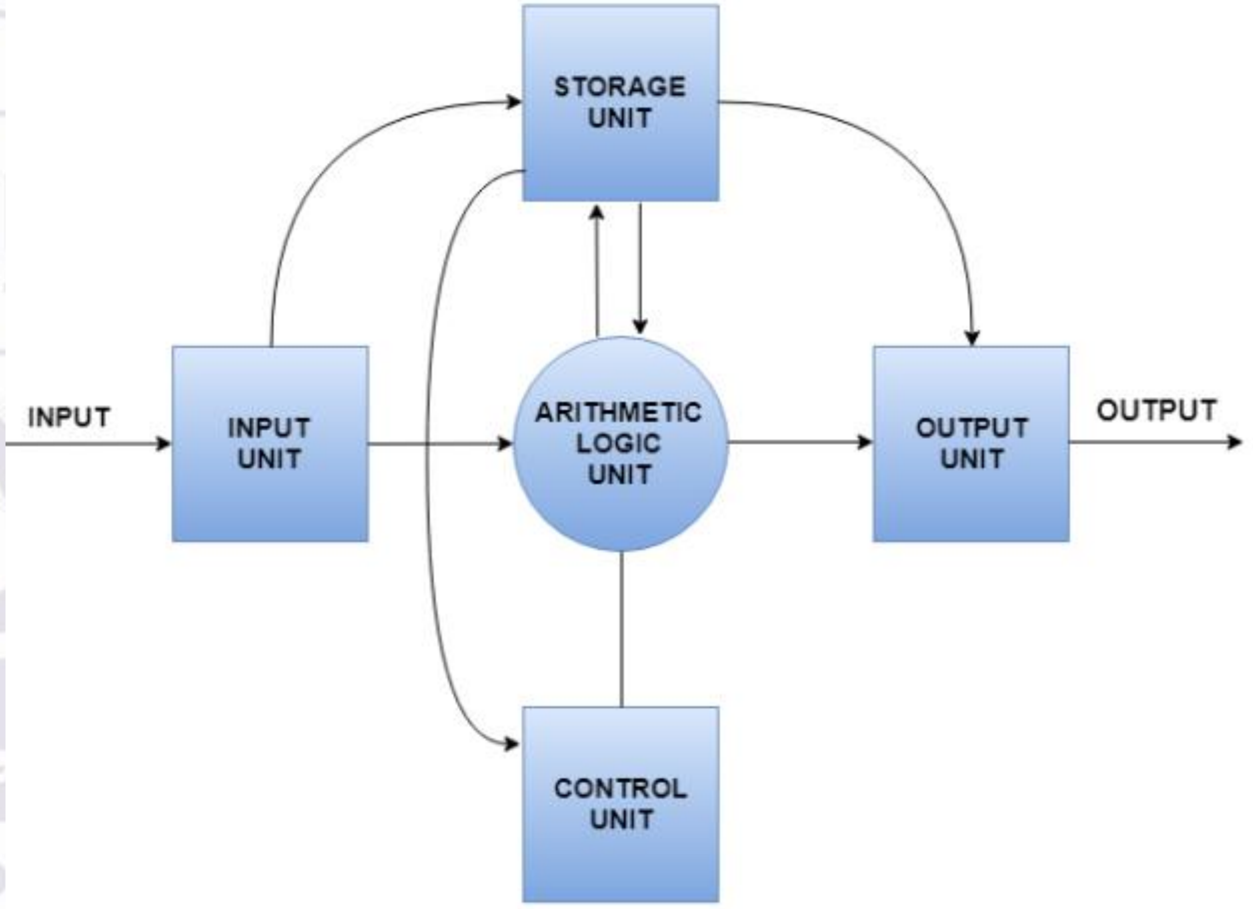
Computer System Architecture

Computer System Architecture

- A computer system is basically a machine that **simplifies complicated tasks**.
- It should **maximize performance** and **reduce costs** as well as **power consumption**.
- The different components in the Computer System Architecture are:
- **Input Unit,**
- **Output Unit,**
- **Storage Unit,**
- **Arithmetic Logic Unit,**
- **Control Unit** etc.



- The **input data travels** from **input unit to ALU.**
- Similarly, the **computed data travels** from **ALU to output unit**.
- The **data constantly moves from storage unit to ALU and back again**. This is because stored data is computed on before being stored again.
- The **control unit controls all the other units** as well as their **data**.



- Input Unit
- Connects the external environment with internal computer system.
- Provides data and instructions to the computer system.
- Commonly used input devices are *keyboard, mouse, magnetic tape* etc.
- Input unit performs following tasks:
  - ➢ Accept the data and instructions from the outside environment.
  - ➢ Convert it into machine language.
  - ➢ Supply the converted data to computer system.

- **Output Unit:**
- **It connects the internal system of a computer to the external environment.**
- **It provides the results of any computation, or instructions to the outside world.**
- **Some output devices are *printers, monitor* etc.**

- **Storage Unit:**
  - ➢ **This unit holds the data and instructions.**
  - ➢ **It stores the intermediate results before these are sent to the output devices.**
  - ➢ **It also stores the data for later use.**

- **The storage unit of a computer system can be divided into two categories:**
- **Primary Storage:**
  - ➢ **This memory is used to store the data which is being currently executed.**
  - ➢ **It is used for temporary storage of data.**
  - ➢ **The data is lost, when the computer is switched off.**
  - ➢ **RAM is used as primary storage memory.**
- **Secondary Storage:**
  - ➢ **The secondary memory is slower and cheaper than primary memory.**
  - ➢ **It is used for permanent storage of data.**
  - ➢ **Commonly used secondary memory devices are hard disk, CD etc.**

- **Arithmetic Logical Unit (ALU):**
- **All the calculations are performed in ALU of the computer system.**
- **The ALU can perform basic operations such as addition, subtraction, division, multiplication etc.**
- **Whenever calculations are required, the control unit transfers the data from storage unit to ALU.**
- **When the operations are done, the result is transferred back to the storage unit.**

- **Control Unit:**

- **It controls all other units of the computer.**
- **It controls the flow of data and instructions to and from the storage unit to ALU.**
- **Thus it is also known as central nervous system of the computer.**

- CPU
- It is Central Processing Unit of the computer.
- The control unit and ALU are together known as CPU. CPU is the brain of computer system.
- It performs following tasks:
  - ➢ It performs all operations.
  - ➢ It takes all decisions.
  - ➢ It controls all the units of computer.

- Computer system architecture also means design or construction of a computer.
- A computer system may be organized in different ways.
- Some computer systems have single processor and other have multiprocessors.
- So computer systems categorized in three different ways.

1. Single Processor Systems
2. Multiprocessor Systems
3. Clustered Systems

Single-Processor Systems

- On a single processor system, there is **only one CPU** that perform all the activities in the computer system.
- These processors **execute only a limited system programs** and **do not run the user program.**
- Ex: Micro-computer

Multiprocessor Systems

- Some systems have two or more processors.
- These systems are also known as parallel systems or tightly coupled systems.
- Mostly the processors of these systems share the common system bus (electronic path) memory and peripheral (input/output) devices.
- These systems are fast in data processing and have capability to execute more than one program simultaneously on different processors.

- **Multiple processors further divided in two types.**
- **Asymmetric Multiprocessing Systems (AMS)**
- **Symmetric Multiprocessing Systems (SMS)**

Asymmetric multiprocessing systems

- The multiprocessing system, in which each processor is assigned a specific task, is known as Asymmetric Multiprocessing Systems.
- In this system there exists master slave relationship like one processor defined as master and others are slave.
- The master processor controls the system and other processor executes predefined tasks.

Symmetric multiprocessing system

- The multiprocessing system in which each processor performs all types of task within the operating system is known as Symmetric multiprocessing system.
- All processors are peers and no master slave relationship exits.
- In SMP systems many programs can run simultaneously.

Clustered systems

- This system also contains multiple processors.
- The clustered system is composed of multiple individual systems that are connected together.
- In this system, individual systems or computers share the same storage and linked together via local area network.
- Other form of clustered system includes parallel clusters and clustering over a wide area network.
- In parallel cluster multiple hosts can access the same data on the shared storage.

Components of an Operating Systems

- There are two basic components of an Operating System.
- Shell
- Kernel

- **Shell:** Shell is the outermost layer of the Operating System, and it handles the interaction with the user. The main task of the Shell is the management of interaction between the User and OS.
- **Kernel:** The kernel is one of the components of the Operating System which works as a core component. The rest of the components depends on Kernel for the supply of the important services that are provided by the Operating System. The kernel is the primary interface between the Operating system and Hardware.

- **Functions of Kernel:** The following functions are to be performed by the Kernel.
- It helps in controlling the System Calls.
- It helps in I/O Management.
- It helps in the management of applications, memory, etc.

- **Types of Kernel**
- There are four types of Kernel that are mentioned below.

a. Monolithic Kernel

b. Microkernel

c. Hybrid Kernel

d. Exokernel

OS Structure

Operating-System Structure

- A modern operating system is large and complex.
- For efficient performance and implementation, an OS should be partitioned into separate subsystems/ components/ modules.
- Each of the sub systems/parts should be well defined with clear inputs, outputs, and functions.
- These subsystems can then be arranged in various architectural configurations:

1. Simple Structure

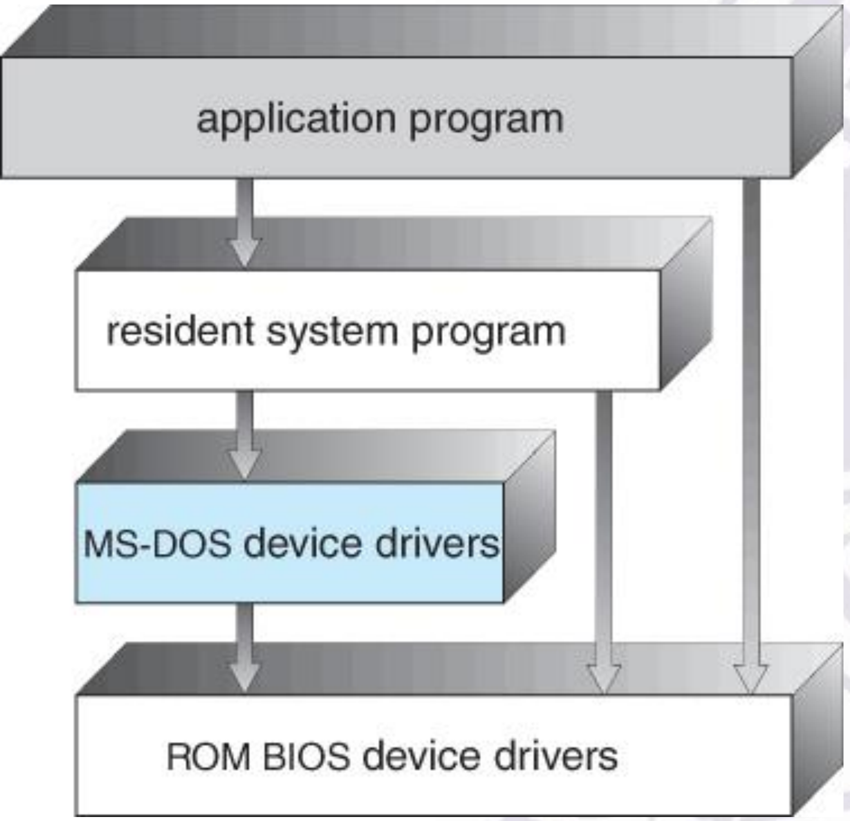


Fig.1 - MS-DOS layer structure

- **Such operating systems designed at the very beginning.**
- **Base hardware can be accessed by all three layers above it.**
- **So in this structure the interfaces and levels of functionalities are not well separated.**
- **Hence we see that application programs are able to access the basic I/O routines and write directly to the display and disk drivers.**
- **MS-DOS is an example of such operating system.**
- **In MS-DOS application programs are able to access the basic I/O routines.**

- **Advantages of Simple structure:**
- **It delivers better application performance because of the few interfaces between the application program and the hardware.**
- **Easy for kernel developers to develop such an operating system.**

- **Disadvantages of Simple structure:**
- **Do not have well defined structure.**
- **The interfaces and levels of functionality are not well separated.**
- **It does not break the system into subsystems,**
- **These types of operating system cause the entire system to crash if one of the user programs fails.**
- **It allows all programs direct access to the underlying hardware.**
- **It has no distinction between user and kernel modes.**

- **The original UNIX OS used a simple layered approach, but almost all the OS was in one big layer, hence we call it as monolithic structure.**

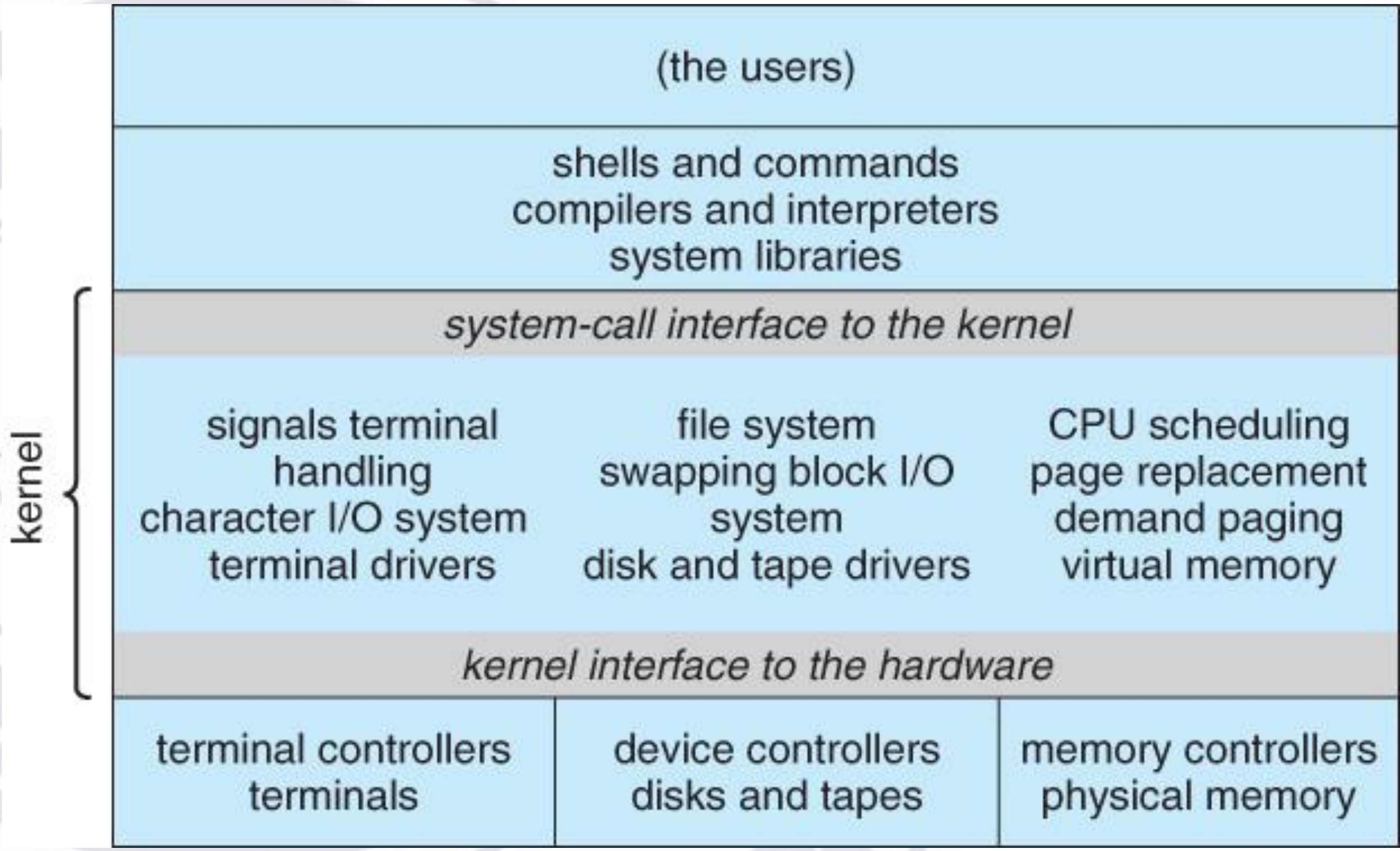


Fig.2 - Traditional UNIX system structure

Layered Approach

- In this structure the OS is broken into number of layers (levels).
- The bottom layer (layer 0) is the hardware and the topmost layer (layer N) is the user interface.
- These layers are so designed that each layer uses the functions of the lower level layers only.

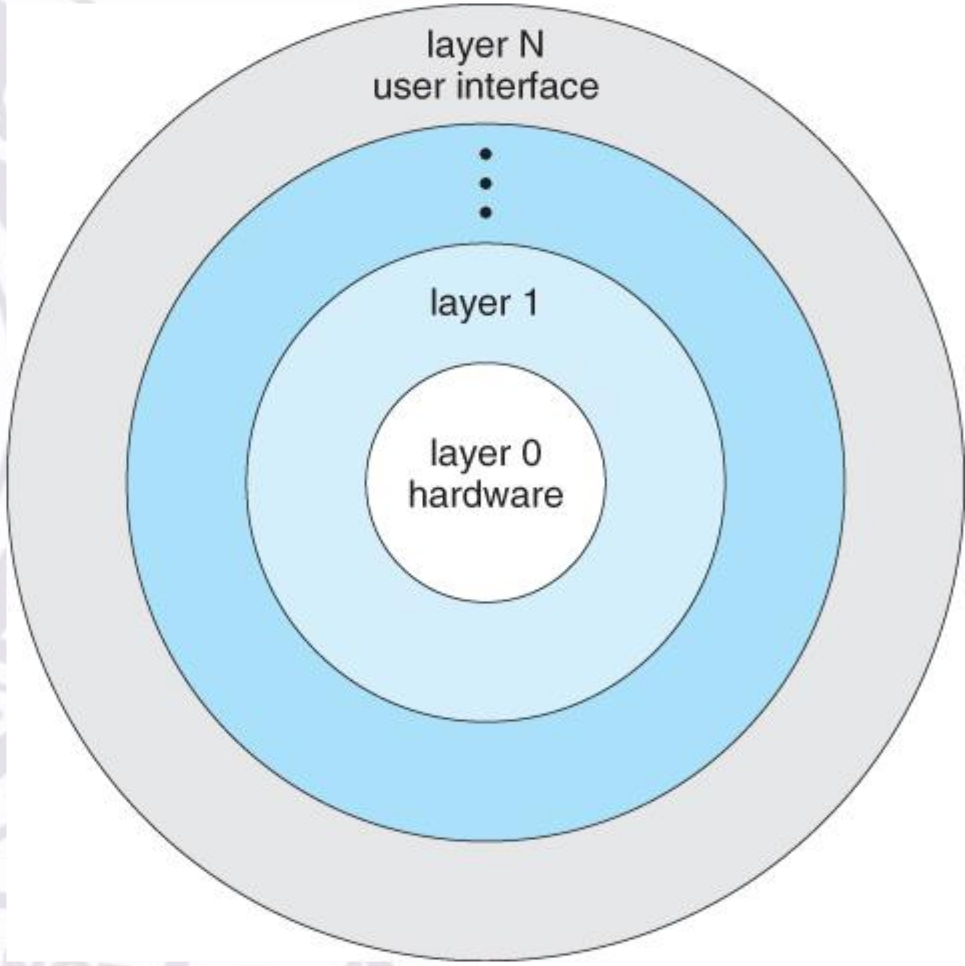


Figure 2.13 - A layered operating system

- **This simplifies the debugging process as if lower level layers are debugged and an error occurs during debugging then the error must be on that layer only as the lower level layers have already been debugged.**
- **The main disadvantage of this structure is that at each layer, the data needs to be modified and passed on which adds overhead to the system.**
- **Moreover careful planning of the layers is necessary as a layer can use only lower level layers. UNIX is an example of this structure.**

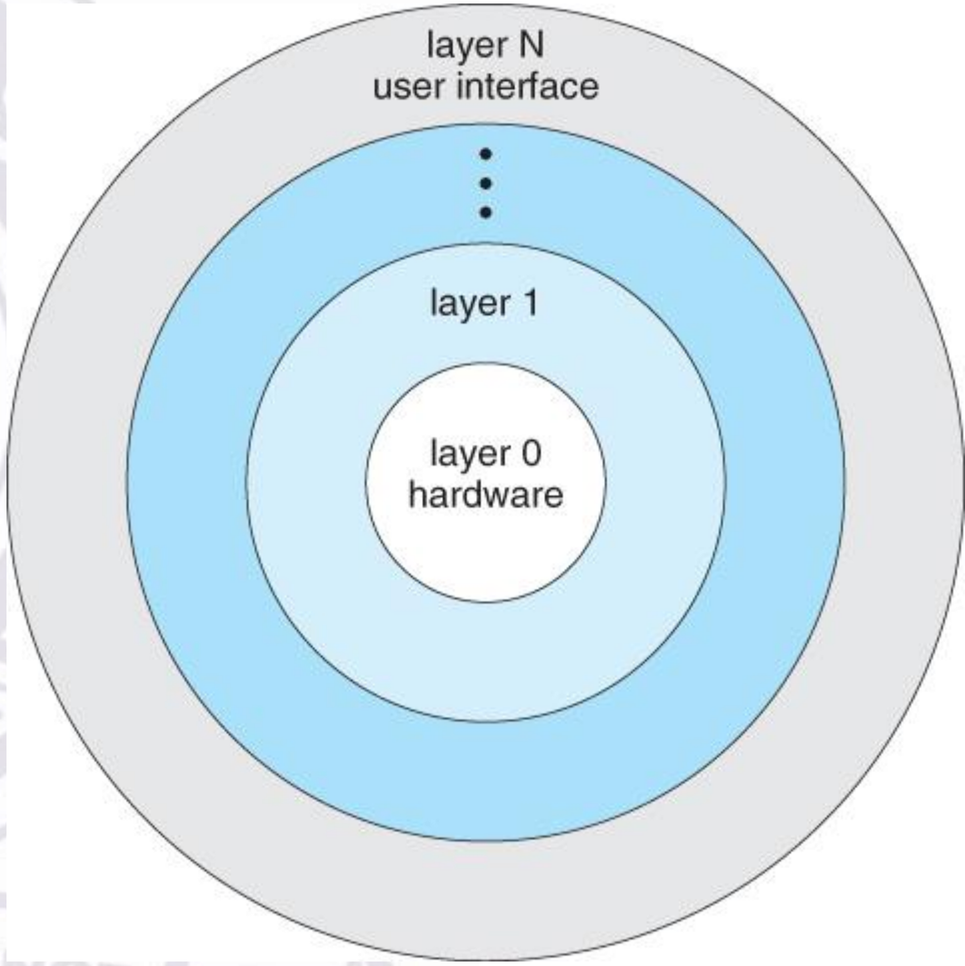


Figure 2.13 - A layered operating system

- **Advantages of Layered structure:**
- **Layering makes it easier to enhance the operating system as implementation of a layer can be changed easily without affecting the other layers.**
- **It is very easy to perform debugging and system verification.**
- **Disadvantages of Layered structure:**
- **In this structure the application performance is degraded as compared to simple structure.**
- **It requires careful planning for designing the layers as higher layers use the functionalities of only the lower layers.**

Microkernels

- This structure designs the operating system by removing all non-essential services from the kernel, and implement them as system and user-level programs.
- This result in a smaller and efficient kernel called the micro-kernel.
- The main function of the microkernel is to provide communication between the client program and the various services that are also running in user space.
- Security and protection can be enhanced, as most services are performed in user mode, not kernel mode.
- System expansion can also be easier, because it only involves adding more system applications, not rebuilding a new kernel.

- **Example: Mach was the first and most widely known microkernel, and now forms a major component of Mac OS.**
- **Windows NT was originally microkernel, but suffered from performance problems relative to Windows 95.**
- **Another microkernel example is QNX, a real-time OS for embedded systems.**

- **Advantages of Micro-kernel structure:**
- All new services need to be added to user space and does not require the kernel to be modified.
- It is more secure and reliable as if a service fails then rest of the operating system remains untouched.
- It makes the operating system portable to various platforms.
- As microkernels are small so these can be tested effectively.
- **Disadvantages of Micro-kernel structure:**
- Increased level of inter module communication degrades system performance.

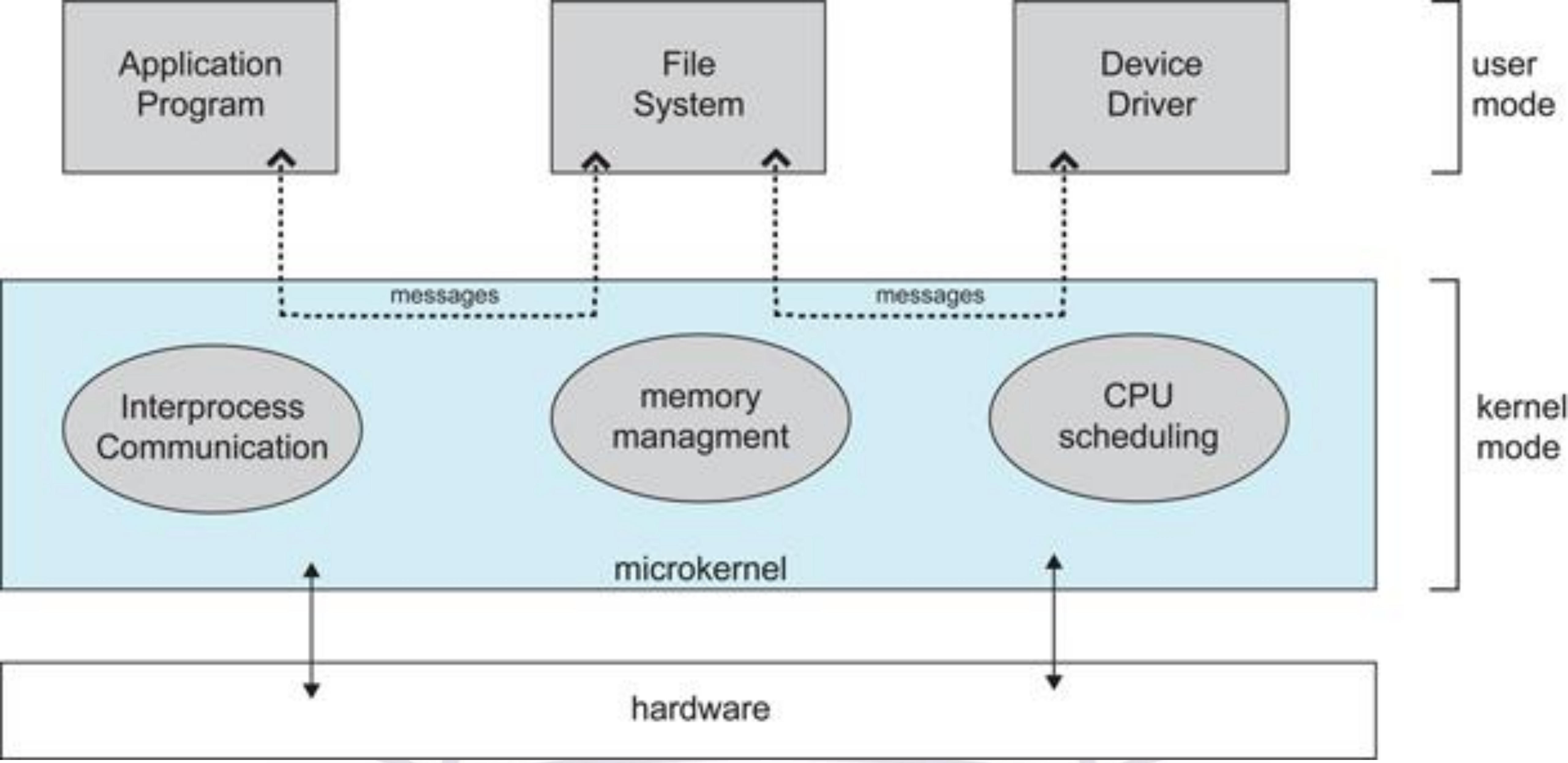
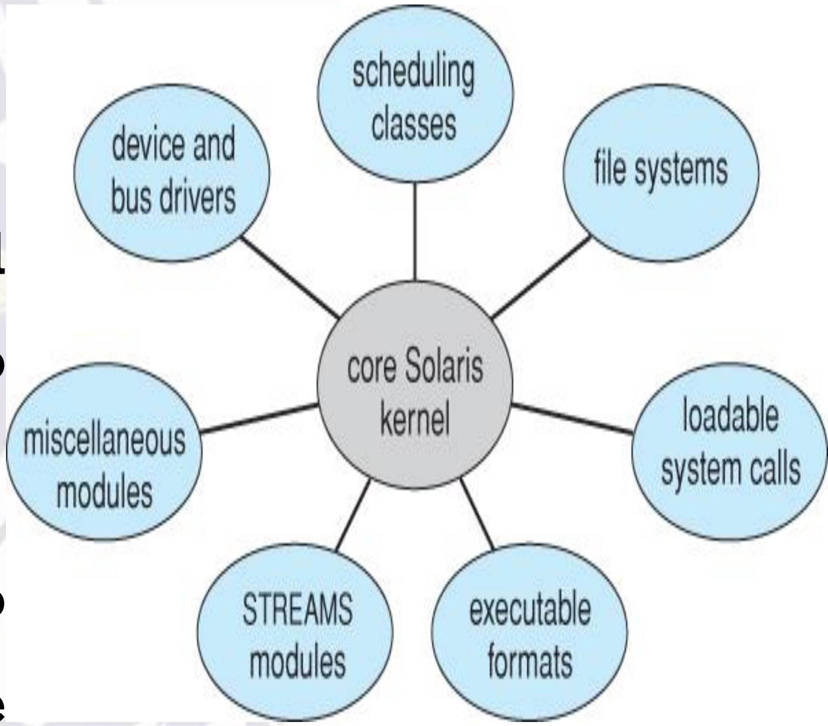


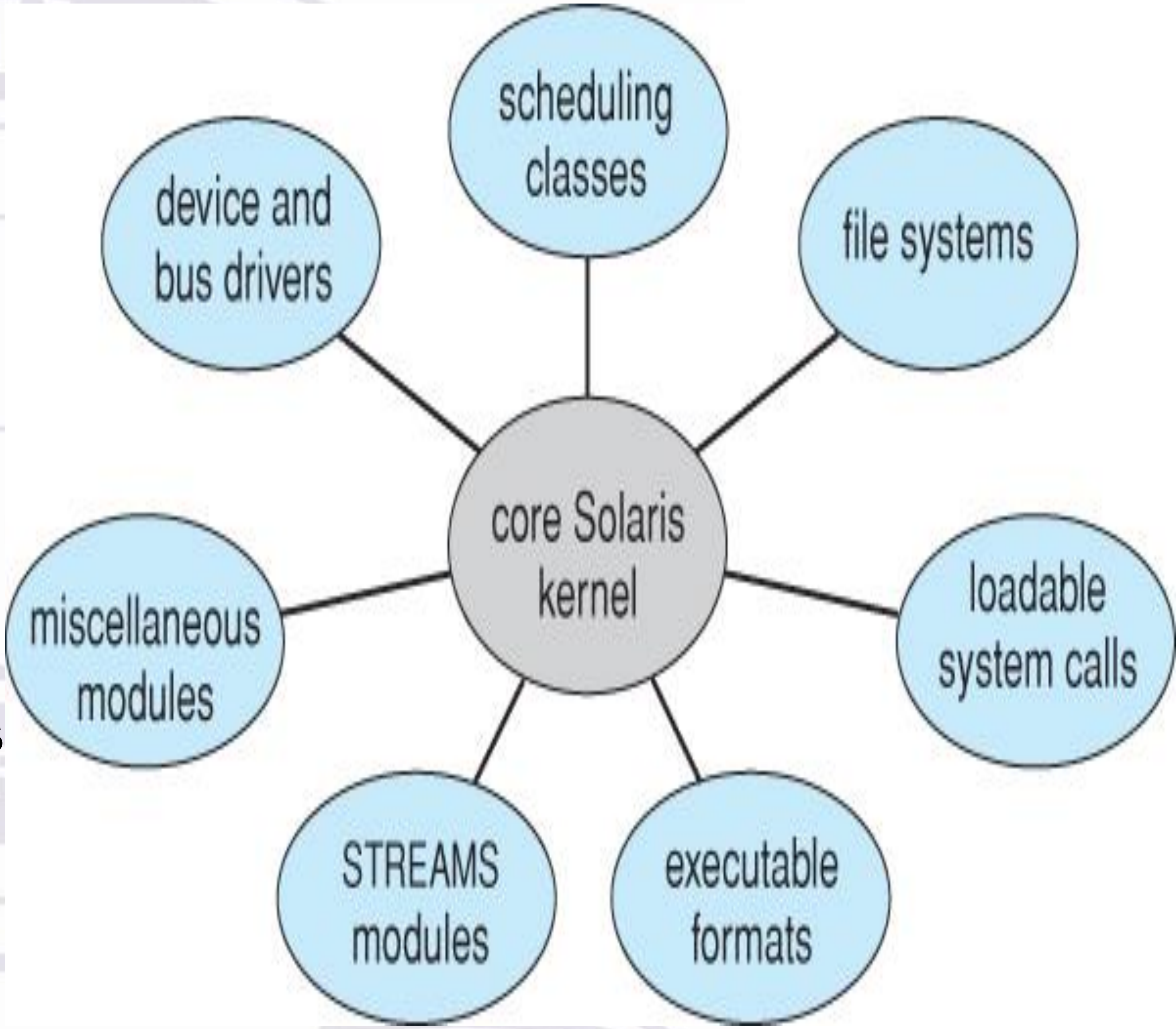
Figure 2.14 - Architecture of a typical microkernel

Modules/Modular structure or approach:

- **Modern OS development is object-oriented, with a relatively small core kernel and a set of *modules*.**
- **Ex: the Solaris structure.**
- **Each subsystem in a module has clearly defined tasks and interfaces, but any module is free to contact any other module.**
- **The kernel is relatively small does not have to implement message passing since modules are free to contact each other directly.**



- **1. Scheduling classes**
- **2. File systems**
- **3. Loadable system calls**
- **4. Executable formats**
- **5. STREAMS modules**
- **6. Miscellaneous modules**
- **7. Device and bus drivers**



- **These modules will be dynamically loaded to the core kernel as and when required.**
- **Modular approach looks some thing like layered approach as well as it looks little bit like microkernel approach.**
- **Each of the module can communicate any other module through core kernel. Hence it is more flexible than layered structure.**

- **Advantages as compared to microkernel approach is that in microkernel, we need to have message passing in order to communicate between the modules, because they are implemented as system level programs above the kernels.**
- **But in this case the modules are connected with core kernel directly dynamically. Hence message passing is not required/ hence system overhead is not there.**

Hybrid Systems

- In practice, very few operating systems adopt a single, strictly defined structure.
- Many OS combine different structures, resulting in hybrid systems that address performance, security, and usability issues.
- In this section, we explore the structure of three hybrid systems: the Apple Mac OS X operating system and the two most prominent mobile operating systems—iOS and Android.

- **Mac OS X:**
- **The Max OSX architecture relies on the Mach microkernel for basic system management services, and the BSD (*Berkeley Software Distribution or Berkeley Standard Distribution*) kernel for additional services.**

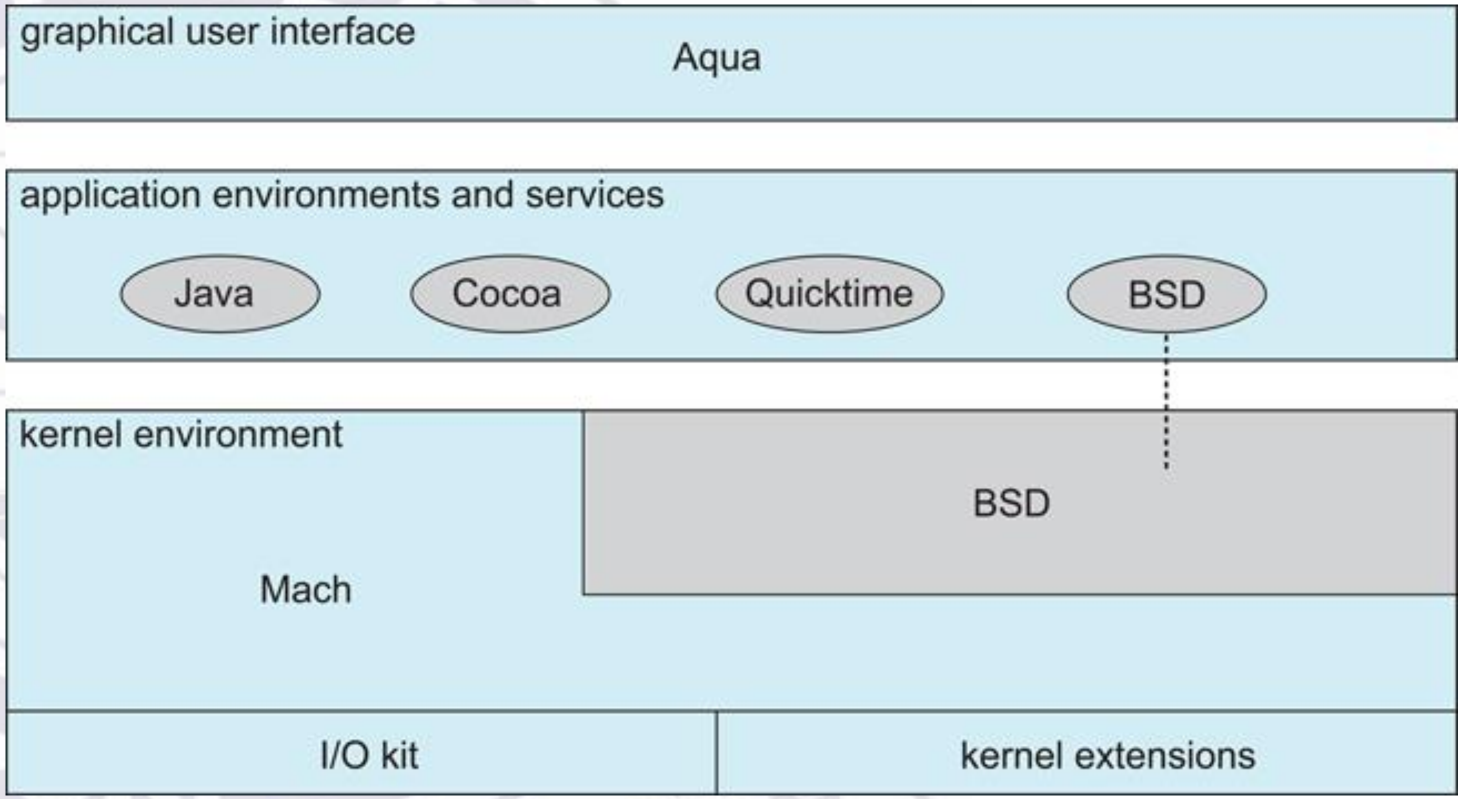


Figure - The Mac OS X structure

- iOS
- The iOS operating system was developed by Apple for iPhones and iPads.
- It runs with less memory and computing power needs than Max OS X, and supports touchscreen interface and graphics for small screens:

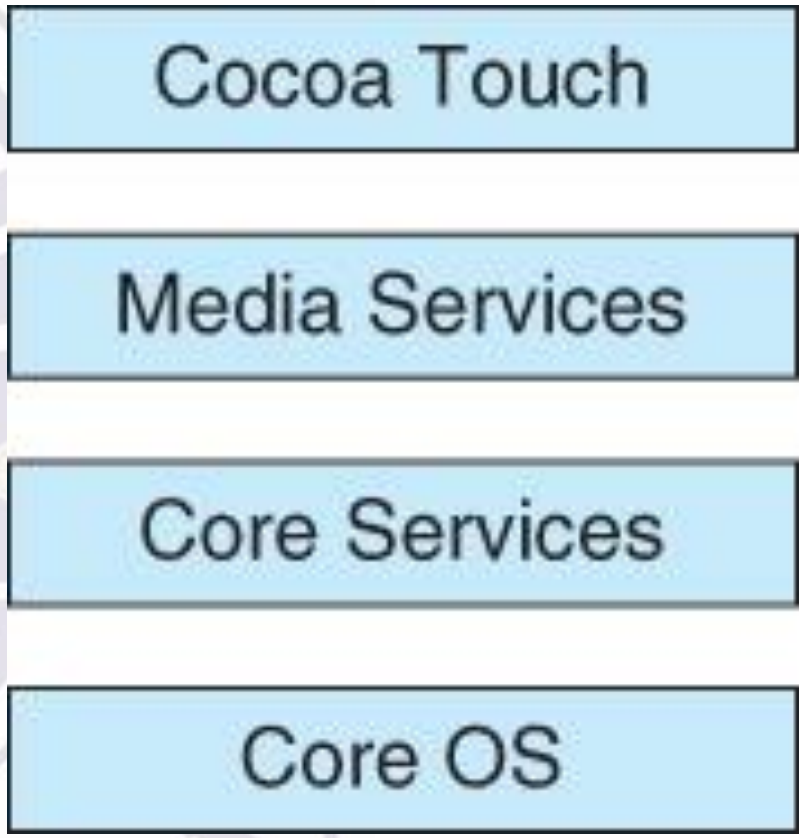


Figure 2.17 - Architecture of Apple's iOS.

- Android
- The Android OS was developed for Android smartphones and tablets by the Open Handset Alliance, primarily Google.
- Android is an open-source OS, as opposed to iOS, which has lead to its popularity.
- Android includes versions of Linux and a Java virtual machine both optimized for small platforms.
- Android apps are developed using a special Java-for-Android development environment.

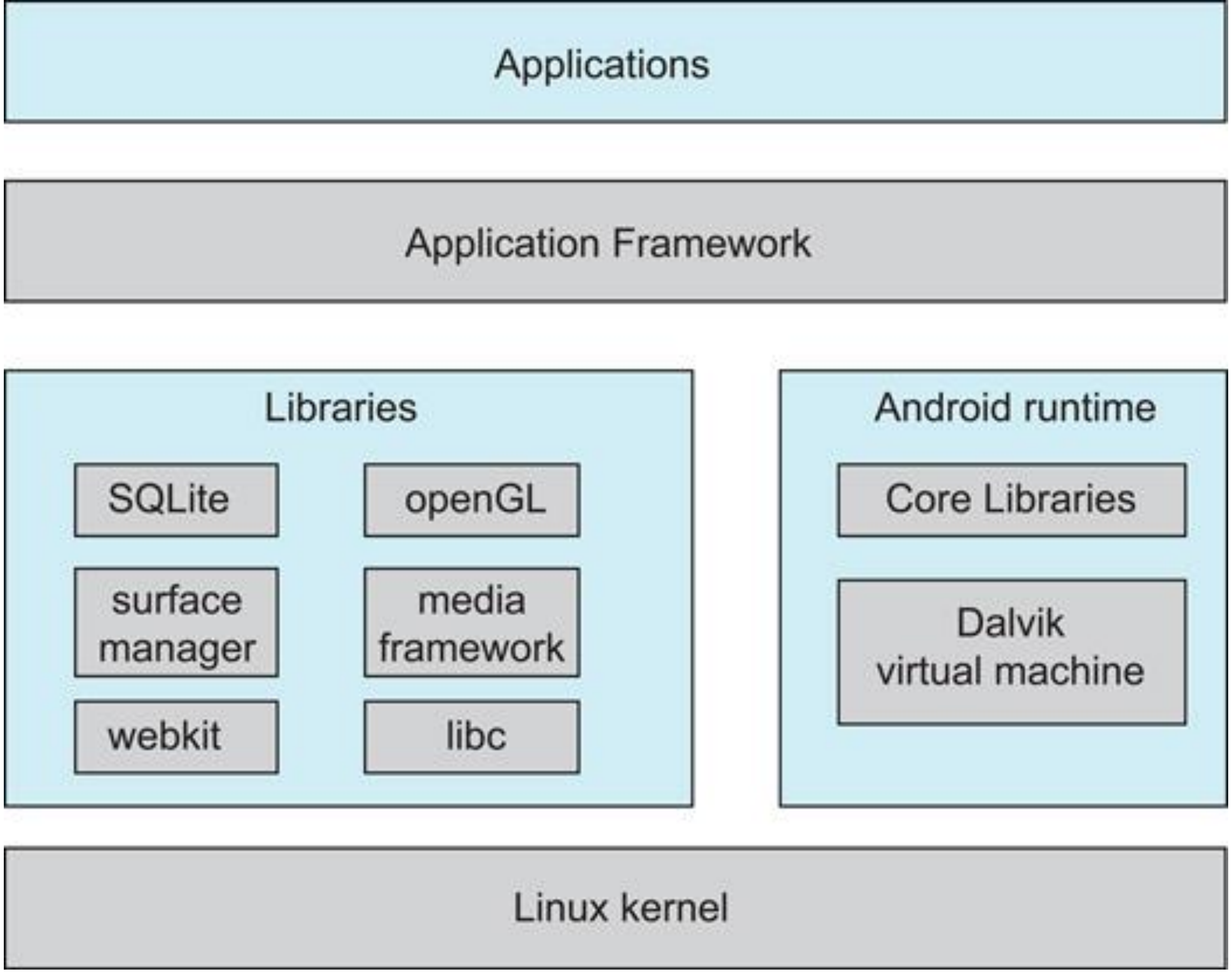


Figure 2.18 - Architecture of Google's Android

Operation of the Operating System

Introduction

- Modern operating systems are interrupt driven.
- If

➢ there are no processes to execute,

➢ no I/O devices to service, and

➢ no users to whom to respond,

➢ an operating system will sit ideally (quietly), And waiting for something to happen.

- **Events are almost always signalled by the occurrence of an interrupt or a trap.**
- **Interrupts are important because they give the user better control over the computer.**
- **Interrupts are signals sent to the CPU by external devices, normally I/O devices.**

- They tell the CPU to stop its current activities and execute the appropriate part of the operating system.
- There are three types of interrupts:
- Hardware Interrupts are generated by hardware devices to signal that they need some attention from the OS.
- They may have just received some data; or they have just completed a task which the operating system previous requested, such as transferring data between the hard drive and memory.

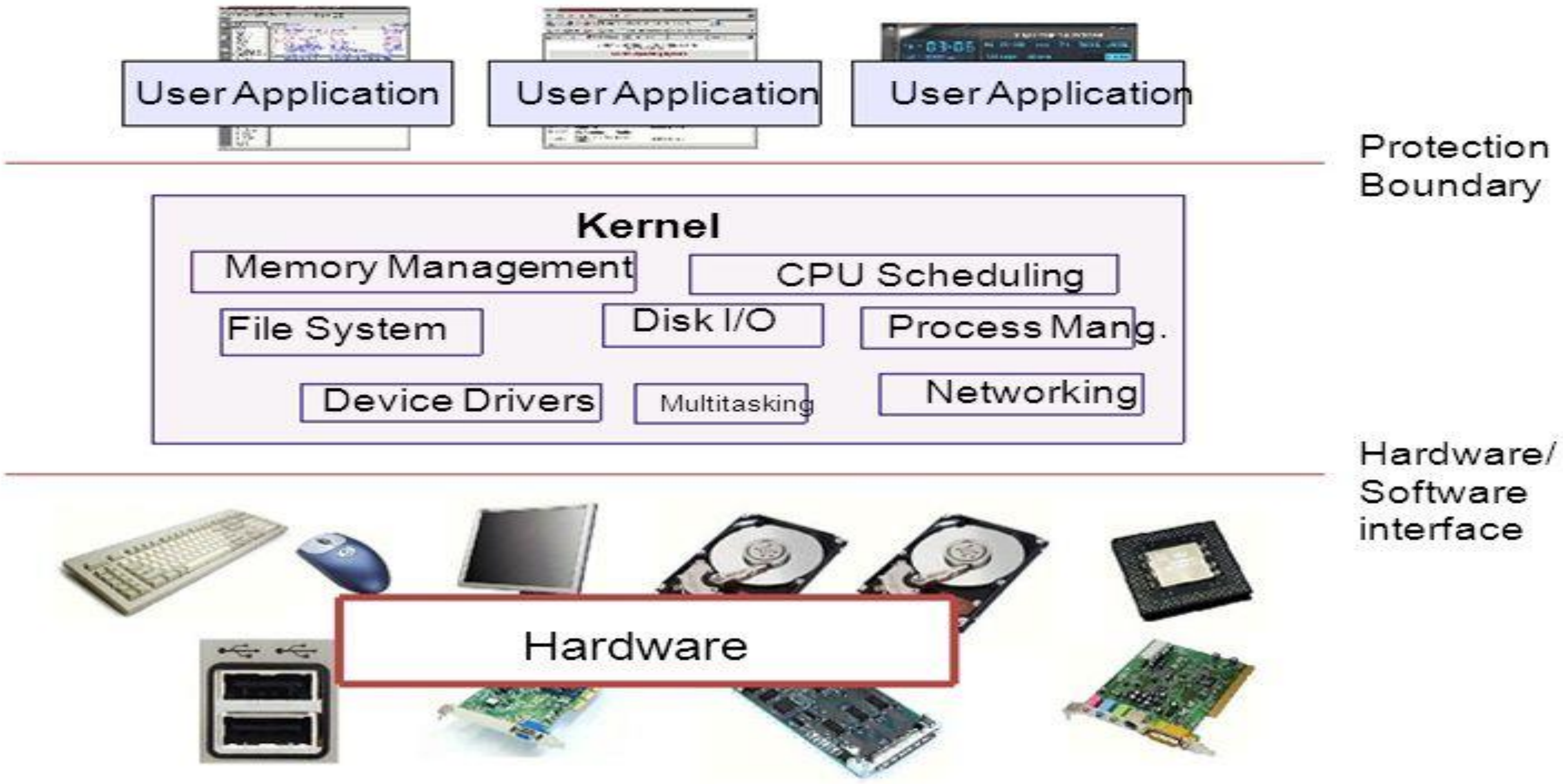
- **Software Interrupts** are generated by programs when they want to request a **system call** to be performed by the operating system.
- **Traps** are generated by the CPU itself to indicate that some error or condition occurred for which assistance from the operating system is needed.
- In computing, a **system call** (**syscall**) is the programmatic way in which a computer program requests a service from the kernel of the operating system on which it is executed.

- **For each type of interrupt, separate segment of code called interrupt service routines are available that determines what action should be taken.**
- **Since the operating system and the users share the hardware and software resources of the computer system, we need to make sure that an error in a user program could cause problems only for the one program that was running.**

- **With sharing, many processes could be adversely affected by a bug in one program.**
- *(A bug is referred to as* ***a failure in the software program****. It produces an incorrect or undesired result that deviates from the expected result.)*
- **Ex: if a process gets stuck in an infinite loop, this loop could prevent the correct operation of many other processes.** ***(It might modify another program or the data of another program, or even the operating system itself).***

- **Without protection against these sorts of errors, either the computer must execute only one process at a time or all output must be suspect.**
- **Hence a properly designed operating system must ensure that an incorrect (or malicious) program cannot cause other programs to execute incorrectly.**
- **(a) Dual mode operation**
- **(b) Timer**

OS Operations a snapshot



Dual-Mode Operation

- In order to protect OS & System programs from malfunctioning programs, two modes of operations were evolved.
- (a) user mode and
- (b) kernel mode (Also known as supervisor mode or system mode or privileged mode).

- Dual mode of operation is used to provide protection & security to user program & also OS and other system components from event users.
- When the system is executing user instructions, the system is in user mode.
- However, when a user application requests a service from the operating system, it must transition from user to kernel mode to fulfil the request.

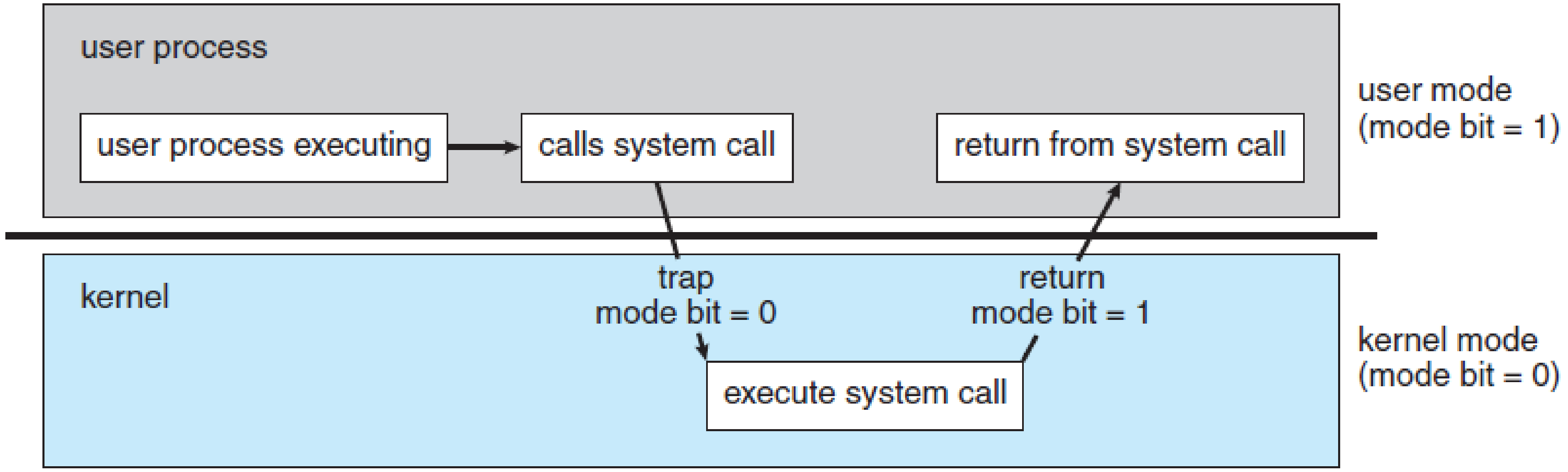


Figure : Transition from user to kernel mode.

- **A mode bit is added to the hardware of computer to indicate the current mode: kernel (0) or user (1).**
- **With the mode bit, we are able to distinguish between a task that is executed on behalf of the operating system and one that is executed on behalf of the user.**

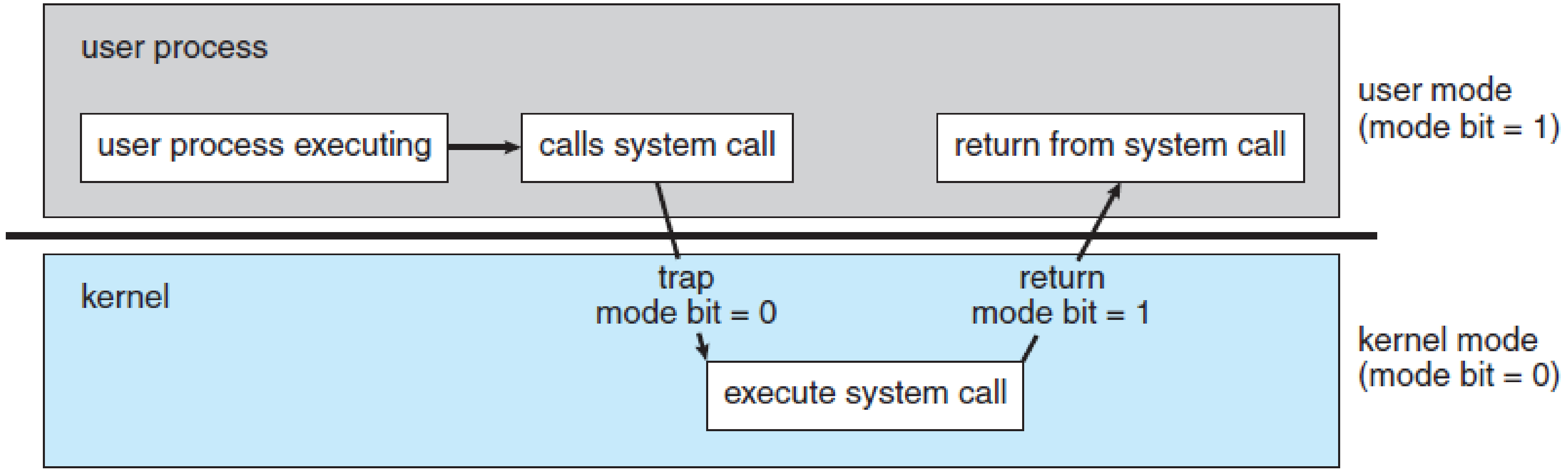


Figure : Transition from user to kernel mode.

- **At system boot time, the hardware starts in kernel mode.**
- **The OS is then loaded and starts user applications in user mode.**
- **Whenever a trap or interrupt occurs, the hardware switches from user mode to kernel mode (that is, changes the state of the mode bit to 0).**

- **Thus, whenever the OS gains control of the computer, it is in kernel mode.**
- **The system always switches to user mode (by setting the mode bit to 1) before passing control to a user program.**
- **The dual mode of operation provides us with the means for protecting the operating system from errant users—and errant users from one another.**
- **This protection is achieved by special machine instructions called as privileged instructions.**

- The **hardware allows privileged instructions** to be **executed** only in **kernel mode**.
- If an **attempt** is made to **execute** a privileged instruction in user mode, the hardware **does not execute** the instruction but rather **treats it** as **illegal** and **traps** it to the operating system and **execution mode changes** from **user to kernel mode**.

- **The instruction to switch to user mode is an example of a privileged instruction.**
- **Some other examples include I/O control, timer management, and interrupt management.**
-

Timer

- We must ensure that the operating system maintains control over the CPU.
- We must prevent a user program from getting stuck in an infinite loop or not calling system services and never returning control to the operating system.
- To accomplish this goal, we can use a timer.
- A timer can be set to interrupt the computer after a specified period. The period may be fixed (for example, 1/60 second) or variable (for example, from 1 millisecond to 1 second).

- A variable timer is generally implemented by a fixed-rate clock and a counter.
- The operating system sets the counter.
- Every time the clock ticks, the counter is decremented.
- When the counter reaches 0, an interrupt occurs.
- Before turning over control to the user, the operating system ensures that the timer is set to interrupt.

- **If the timer interrupts, control transfers automatically to the operating system, which may treat the interrupt as a fatal error or may give the program more time.**
- **Clearly, instructions that modify the content of the timer are privileged.**
- **Thus, we can use the timer to prevent a user program from running too long.**

- **A program with a 7-minute time limit, for example, would have its counter initialized to 420.**
- **Every second, the timer interrupts and the counter is decremented by 1.**
- **As long as the counter is positive, control is returned to the user program.**
- **When the counter becomes negative, the operating system terminates the program for exceeding the assigned time limit.**

Evolution of Operating Systems

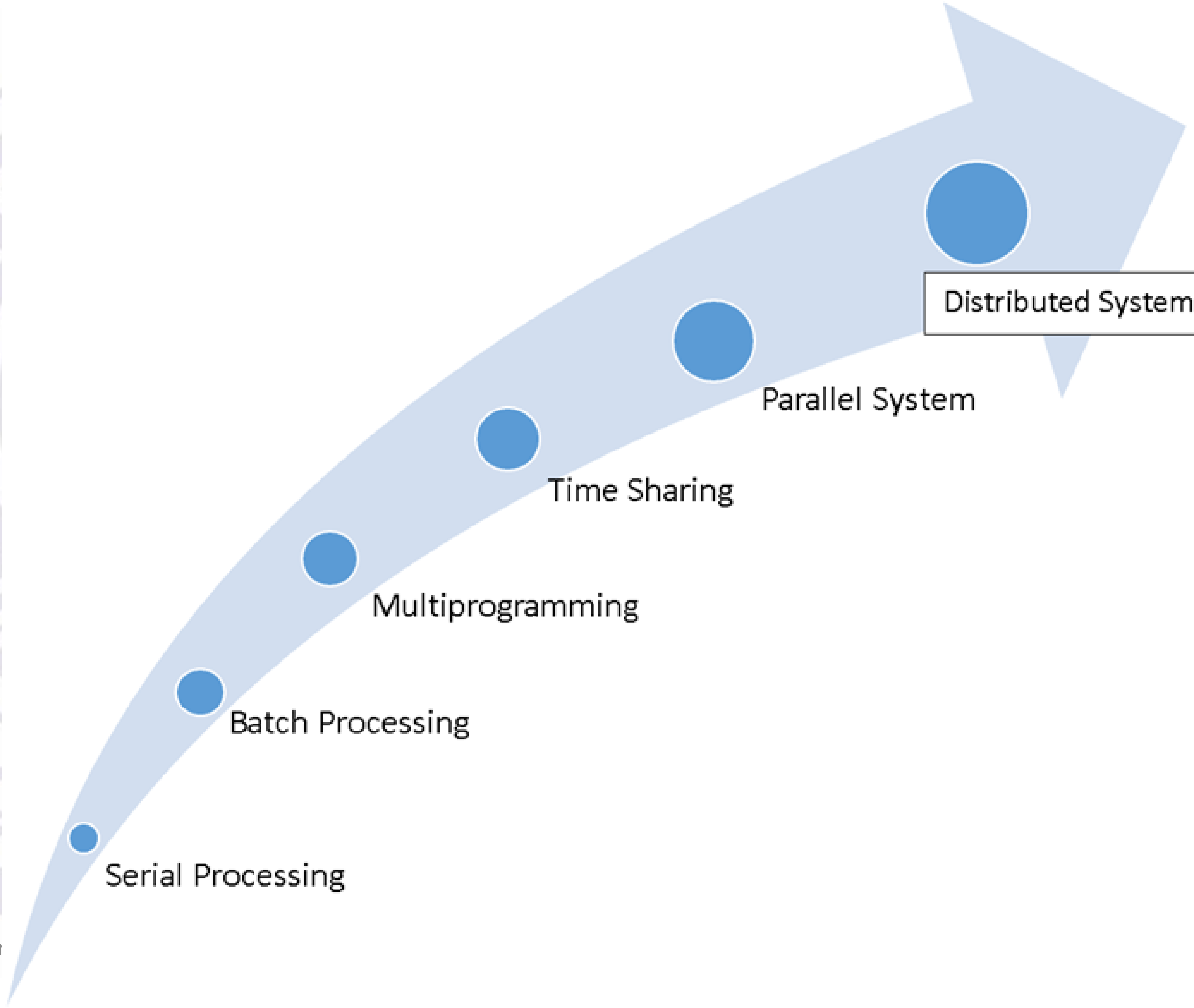
Introduction

- Operating systems have evolved from slow and expensive systems to present-day technology where computing power has reached exponential speeds and relatively inexpensive costs.
- In the beginning, computers were manually loaded with program code to control computer functions.

- **As more users demanded increased computer time and resources, computer scientists determined they needed a system to improve convenience, efficiency, and growth.**
- **As a result, they created an operating system (OS) to process jobs in batches.**
- **Later they created Multitasking and Time-Sharing to run multiple jobs and allow user interaction to improve efficiency.**

Evolution of OS

- The evolution of operating systems is directly dependent on the development of computer systems and how users use them.



- **Serial Processing**
- **Simple Batch,**
- **Multi programmed,**
- **time shared,**
- **Personal Computer,**
- **Parallel,**
- **Distributed Systems,**
- **Real-Time Systems,**
- **Special - Purpose Systems,**

Serial Processing

- **Before 1950, there were no operating systems, and programmers had to directly communicate with the hardware.**
- **And if a programmer wants to execute the programs, the programmer had to follow the following steps:**

- First, type a program or punched card.
- Translate the punch card into a card reader.
- The translated card reader is submitted to the computer, and if any error has occurred, the error was indicated by the lights.
- The Programmer examined at the main memory and register to verify the reason behind the error.
- Then outputs are taken from the printers.
- Then the program is ready to execute another program.

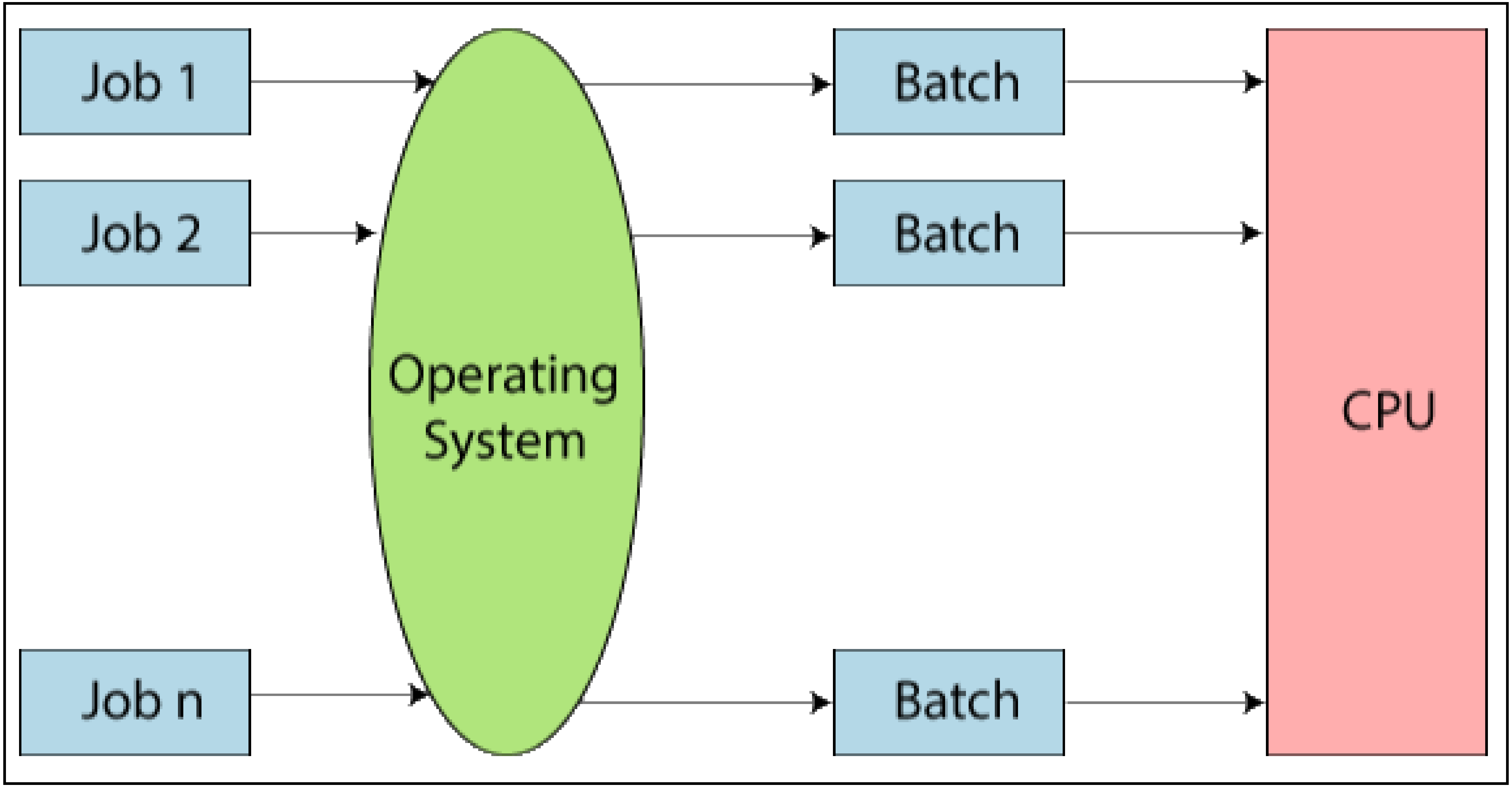
Drawback of Serial Processing

- This type of processing is difficult for users because, in serial processing, more time is required.
- The user cannot start executing another program if the previous program does not complete its execution.
- In serial processing, the programs are submitted to the computer one-by-one. That's the reason behind the name of serial processing.

Simple Batch Processing

- **After Serial Processing, the mid1950s introduced Simple Batch Processing, the first operating system.**
- **In Batch Operating System, there is no direct interaction between user and computer.**
- **Therefore, the user needs to prepare jobs and save offline mode to punch card or paper tape or magnetic tape.**
- **After creating the jobs, hand it over to the computer operator; then the operator sort or creates the similar types of batches like B2, B3, and B4.**

- Now, the computer operator submits batches into the CPU to execute the jobs one by one.
- After that, CPUs start executing jobs, and when all jobs are finished, the computer operator provides the output to the user.
- Despite an improvement over Serial Processing, Simple Batch Processing was slow and consumed large amounts of processing time.


Job 1
Job 2
Job n
Operating System
Batch
Batch
Batch
CPU

Advantages of Simple Batch Operating System

- There is no mechanism to prioritize the processes.
- There is no communication between the user and the computer.
- The ideal time is very less for a batch operating system.

Disadvantages of a Simple Batch Operating System

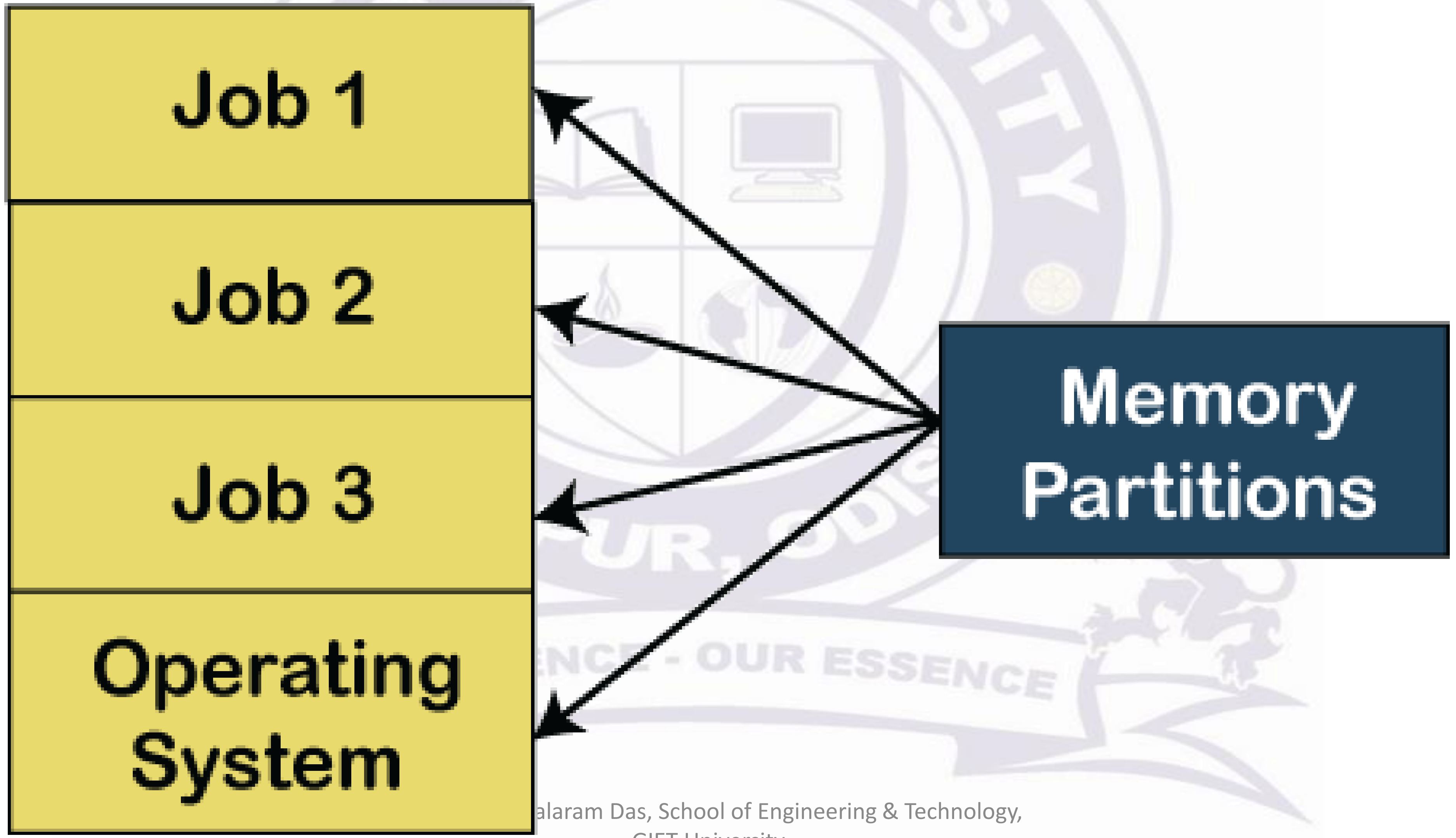
- It is **hard to debug**.
- The Batch operating systems are **costly**.

Multiprogramming

- Multiprogramming means executing multiple programs at the same time with the help of a single processor.
- It is an improved form of Simple Batch Processing.
- In multiprogramming, a number of processes reside in main memory at a time.
- The OS picks and begins to execute one of the jobs in main memory.

- It took advantage of processor idle time by loading the processor with multiple user jobs.
- When one program completes processing, the results transfer to an I/O device, and the processor executed another job waiting in memory.
- Multiprogramming utilizes computer resources efficiently as it switches between jobs until each one completes.
- Ex: Operating systems like Microsoft Windows 7 still use multitasking today.

Multiprogramming



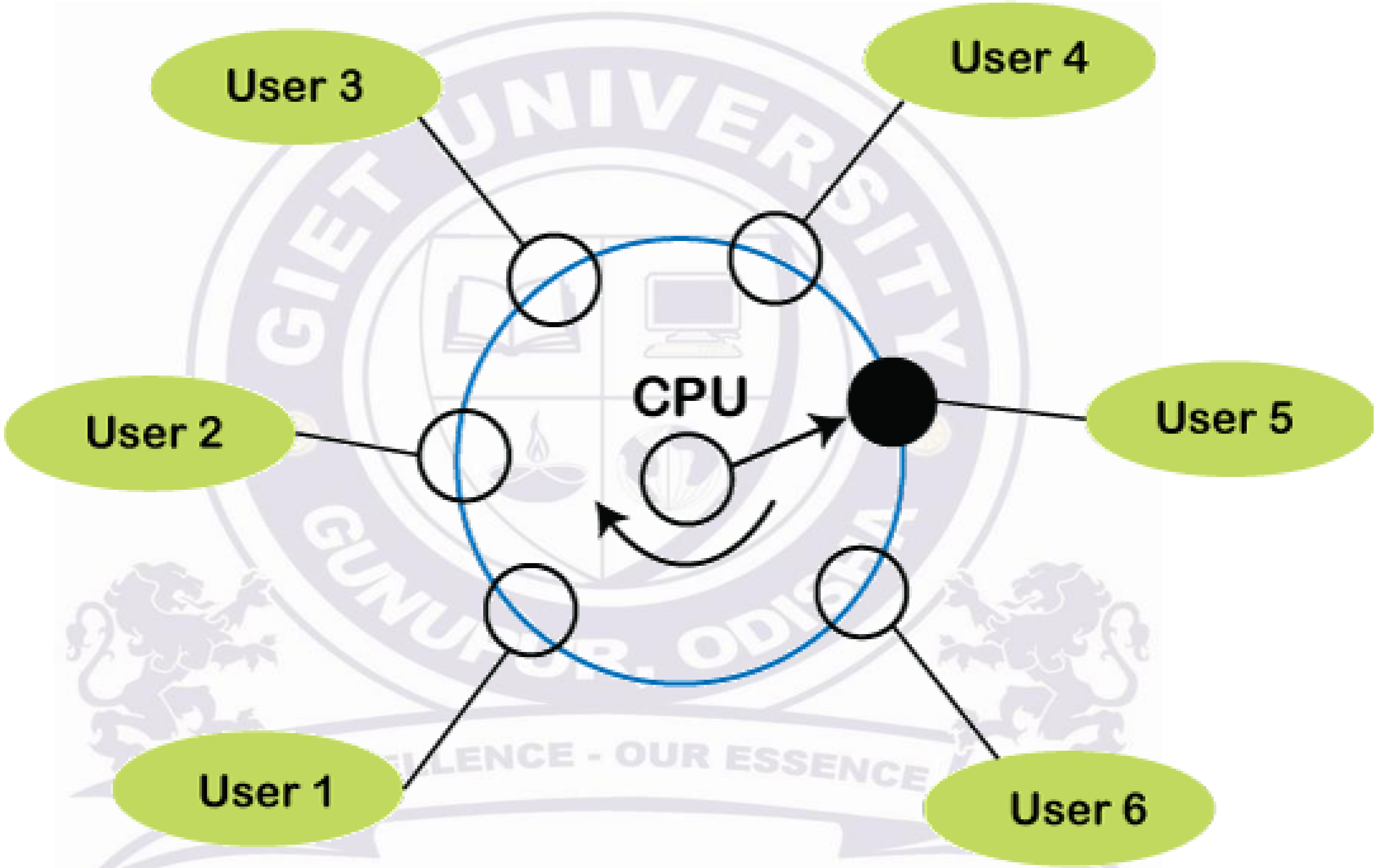
Advantages:

- The CPU will never sit ideal so the performance of CPU will increase.
- Multiprogramming provides effective memory utilization.
- The throughput (*the amount of work completed in a unit of time*) of CPU may also increase.

Time Sharing System

- Time-sharing is an extension of multiprogramming.
- Time-sharing is a technique for multiple users to share system resources simultaneously.
- This offers the users an opportunity to interact directly with the computer.

- Using a terminal and keyboard, each user submits a job request by pressing a transmit key and wait their turn for a response from the processor.
- The intention of Time-sharing is to minimized response time back to the user, reduce idle time, and still maximize processor usage.
- Today, UNIX systems still use Time-sharing.


User 3
User 4
User 2
CPU
User 5
User 1
User 6

Advantages of Time-Sharing System

- Offers the benefit of quick response.
- Minimizes the idle time of the CPU.
- Avoids the duplication of the software.
- Provides efficient utilization of CPU.

Disadvantages of Time-Sharing System

- There is a **problem of reliability**.
- Data **communication problem**.

Personal Computer/Desktop Systems

- **Earlier, CPUs and PCs lacked the features needed to protect an operating system from user programs.**
- **PC operating systems therefore were neither multiuser nor multitasking.**
- **However, the goals of these operating systems have changed with time; instead of maximizing CPU and peripheral utilization, the systems opt for maximizing user convenience and responsiveness.**
- **These systems are called Desktop Systems and include PCs running Microsoft Windows.**

Parallel System

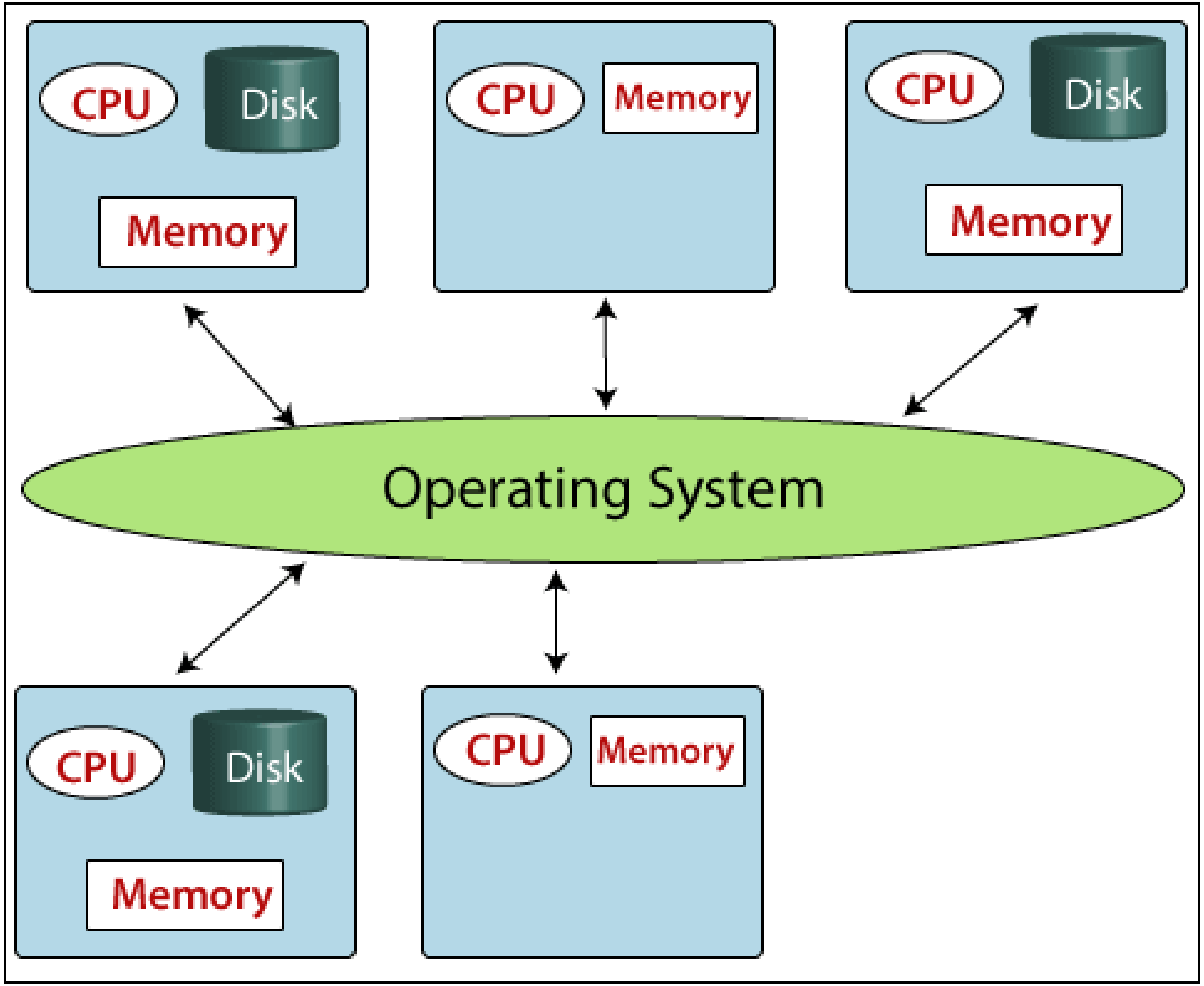
- In the parallel processing system, there are multiple processors, and, in this system, all the processors work concurrently.
- In this type of system, the job is divided into several sub-jobs, and then these sub-jobs are distributed among the processors that are present in the system.
- Parallel processing finishes the job in less time. (multiple processors execute the job in a parallel manner).

Advantages of Parallel Processing System

- **In Parallel processing, throughput is increased.**
- **Multiple jobs are executed in less time.**

Distributed System

- In this system, the processors cannot share a memory or a clock, each processor has its own local memory.
- The processor communicates with one another through various communication lines, such as high-speed buses.
- These systems are referred to as "Loosely Coupled" systems.


CPU
Disk
Memory
CPU
Memory
CPU
Disk
Memory
Operating System
CPU
Disk
Memory
CPU
Memory

Advantages of Distributed System

- Load on the host computer can be reduced.
- This system reduces the delay in the processing of data.
- In a distributed system, the failure of one node will never affect the other node communication because each node is independent of each other.
- It increases the speed of the data exchange through electronic mail.

Disadvantages of Distributed System

- The disadvantages of a distributed system are:
- In a distributed system, if the main network fails, the whole communication will be stopped.
- The language which we are used to create distributed systems is not well defined.
- Distributed systems are costly.

Real-Time Operating System

- It is a special-purpose operating system.
- Used in real-time applications where the data processing must be done in a fixed interval of time.
- It gives the response very fast and quick.
- It is used when a large number of events are processed in a short interval of time.

- It is different from another OS because, in this, time concept is the most crucial part.
- In the real-time system, the process is executed on the basis of priority. The high priority process always executes first.

- **The task of synchronizing the process is done by the real-time operating system so that the process can interact with each other efficiently.**
- **In this way, resources are used effectively without time-wasting.**

Applications of Real-time operating system (RTOS):

- **used inside the Radar gadget.**
- **utilized in Missile guidance.**
- **utilized in on line inventory trading.**
- **used inside the cell phone switching gadget.**
- **used in Medical Imaging Systems.**
- **used inside the Fuel injection gadget.**
- **used inside the Traffic manipulate gadget.**
- **utilized in Autopilot travel simulators.**
- ***Industrial system,***
- ***Nuclear reactors control scientific experiments,***
- ***Military software system,***
- ***Airline resolution system,***
- ***Networked multimedia systems,***
- ***Internet telephony, etc.***

- **There are three kinds of the Real-Time operating system:**
- **Firm Real-time Operating System**
- **Hard Real-time Operating System**
- **Soft Real-time Operating System**

- **Hard-Real time: – In Hard-Real time system, there is some deadline for executing the task, which means that the task must start its execution on the particular scheduled time, and should complete within the assigned duration of time.**
- ***Example: – Aircraft systems, Medical critical care System, etc.***

- **Soft-Real time: – In the Soft-Real time system also, we assign a time to each process, but some delaying in time is acceptable. So, in Soft-real time, deadlines are handled softly. That's why it is called Soft-Real time. *Example:* – Live stock price and Online Transaction System.**

- **Firm-Real time: – In the Firm-Real time system, there is also a deadline for every task to execute. But in this, due to missing deadlines, there may be no big impact, but there can be chances of undesired effects such as problems in the quality of a product.**
- ***Example:* – Multimedia Applications.**

Advantages of Real-Time Operating System

- Real-time operating systems are error-free.
- It offer the facility of memory allocation management.
- It offers better utilization of devices and systems and produces more output from all the resources.
- The real-time operating system more focuses on running the applications and give less importance to those applications which are present in the queue.

Disadvantages of a Real-Time Operating System

- In RTOS, the task of writing the algorithm is very challenging and complex.
- It is expensive because it is using heavy system resources.
- In RTOS, there is less switching of tasks.
- In RTOS, complicated algorithms are used that are tough to understand.

Special - Purpose Systems

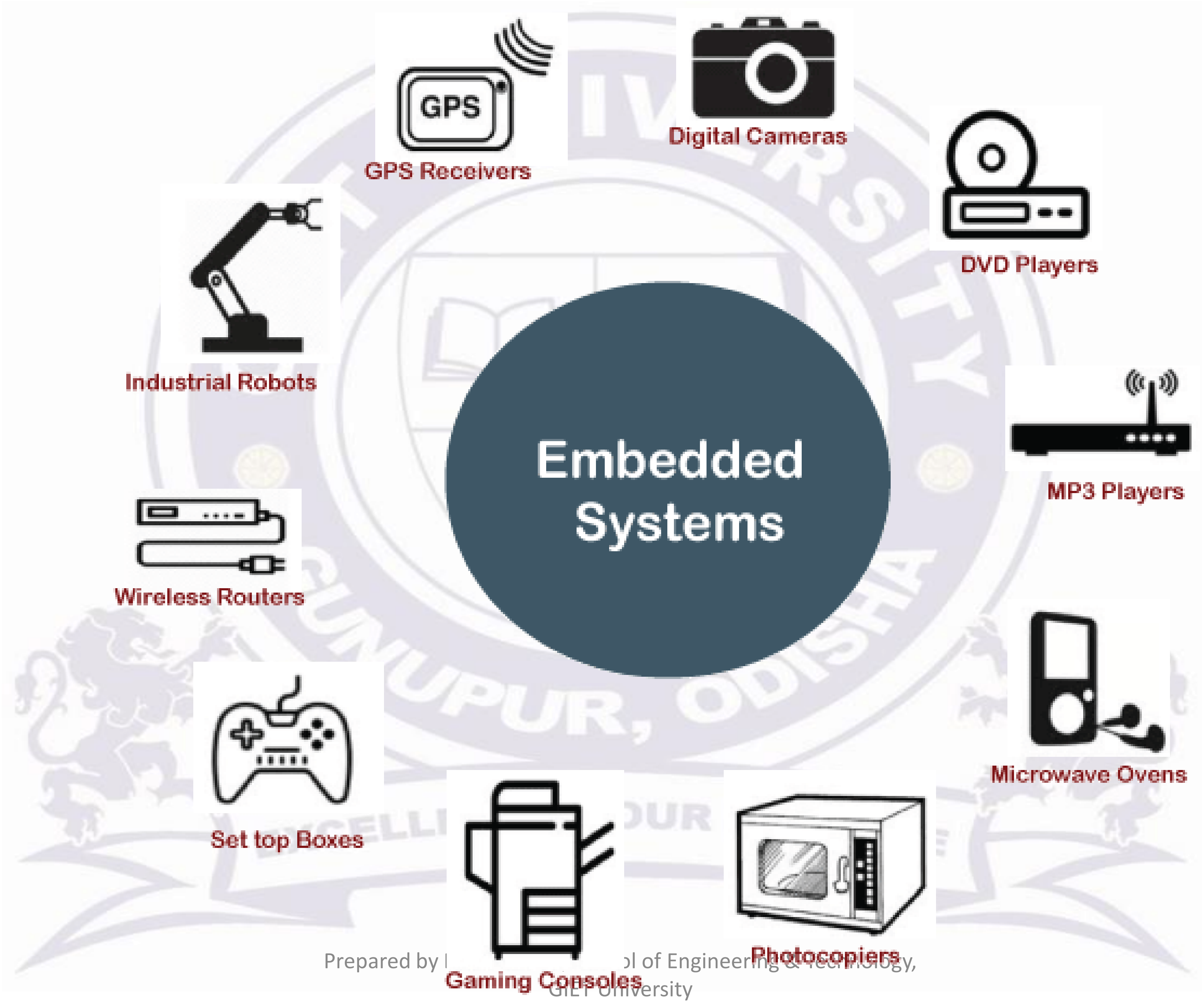
- Systems whose functions are limited and their objective is to deal with limited computational domains, are called special purpose systems.
- There are three types of special purpose systems:
- Real time embedded systems
- Multimedia Systems
- Handheld systems

Real-Time Embedded Systems

- **We use the microwave oven in our kitchen everyday to heat, cook etc.**
- **The display and buttons of a microwave oven depend on an OS to run and such OSs are called real time embedded systems.**

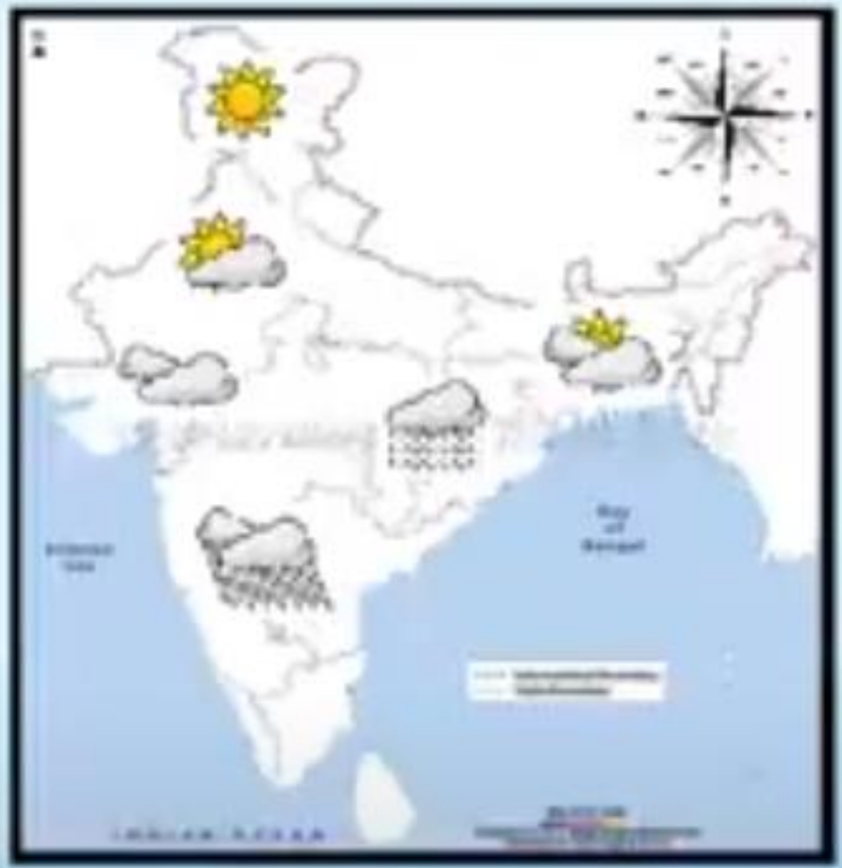
- **They provide computing environment that reacts to input within a specific time period.**
- **The response time in these systems is very small, so result appears instantaneously.**
- **These systems do not interface using keyboard or mouse like PC, but use sensors, actuators etc to interface as they are part of a larger system.**
- **These devices are found everywhere, from car engines and manufacturing robots to VCRs and microwave ovens.**
- **They tend to have very specific tasks.**
- **Ex: Washing Machines, ATM, Set top boxes etc.**

- **The computer must analyze the data and possibly adjust controls to modify the sensor inputs.**
- **Systems that control scientific experiments, medical imaging systems, industrial control systems, and certain display systems are real-time systems.**



Embedded Systems
GPS Receivers
Digital Cameras
DVD Players
MP3 Players
Microwave Ovens
Photocopiers
Gaming Consoles
Set top Boxes
Wireless Routers
Industrial Robots

Multimedia Systems



- Have you seen the weather forecast report in news on T.V?
- weather information comes from a satellite is what we all know
- Do you know how that information is taken, stored and then broadcast on T.V?

- The content which uses different forms of media is called as multimedia.
- The systems which process multimedia data and its applications are called multimedia systems.
- multimedia includes text, images, video, audio, animation and interactive media forms
- Multimedia applications use multiple media sources such as images, audio, video, animation etc.

Examples are : Video games, Video conferencing, Home shopping, Air traffic control, Weather forecast

Multimedia Systems

- Most operating systems are designed to handle conventional data such as text files, programs, word-processing documents, and spreadsheets.
- However, a recent trend in technology is the incorporation of multimedia data into computer systems.
- Multimedia data consist of audio and video files as well as conventional files.
- These data differ from conventional data in that multimedia data—such as frames of video—must be delivered (streamed) according to certain time restrictions (for example, 30 frames per second).
- Multimedia describes a wide range of applications that are in popular use today.

- These include audio files such as MP3 DVD movies, video conferencing, and short video clips of movie previews or news stories downloaded over the Internet.
- Multimedia applications may also include live webcasts (broadcasting over the World Wide Web) of speeches or sporting events and even live webcams that allow a viewer in Manhattan to observe customers at a cafe in Paris.

- **Multimedia applications need not be either audio or video; rather, a multimedia application often includes a combination of both.**
- **For example, a movie may consist of separate audio and video tracks.**
- **Nor must multimedia applications be delivered only to desktop personal computers.**
- **Increasingly, they are being directed toward smaller devices, including PDAs and cellular telephones.**
- **For example, a stock trader may have stock quotes delivered wirelessly and in real time to his PDA.**

Handheld Systems

- Handheld systems include personal digital assistants (PDAs), such as Palm and Pocket-PCs, and cellular telephones, many of which use special-purpose embedded operating systems.
- Developers of handheld systems and applications face many challenges, most of which are due to the limited size of such devices.
- For example, a PDA is typically about 5 inches in height and 3 inches in width, and it weighs less than one-half pound.

3 Distributed and Special Purpose Systems

Press Esc to exit full screen

Special Purpose Systems

Handheld Systems

CLOSE

- Your smart phone is nothing but an handheld system.
- Handheld systems provide some features of a computer with small size, less weight and portability.
- PDA's, Pocket PC's are some of the examples of handheld systems
- Memory should be managed efficiently due to limited size and memory.
- **Web Clipping** technique is used to display subset of a webpage content on the handheld device.

Exit full screen (f)

7:25 / 8:20 • Special Purpose Systems ›

- **Because of their size, most handheld devices have a small amount of memory, slow processors, and small display screens.**
- **We will take a look now at each of these limitations.**
- **The amount of physical memory in a handheld depends upon the device, but typically is is somewhere between 512 KB and 128 MB. (Contrast this with a typical PC or workstation, which may have several gigabytes of memory!)**
- **As a result, the operating system and applications must manage memory efficiently. This includes returning all allocated memory back to the memory manager when the memory is not being used.**
- **Virtual memory allows developers to write programs that behave as if the system has more memory than is physically available.**

- **Currently, not many handheld devices use virtual memory techniques, so program developers must work within the confines of limited physical memory.**
- **A second issue of concern to developers of handheld devices is the speed of the processor used in the devices.**
- **Processors for most handheld devices run at a fraction of the speed of a processor in a PC.**
- **Faster processors require more power.**
- **To include a faster processor in a handheld device would require a larger battery, which would take up more space and would have to be replaced (or recharged) more frequently.**
- **Most handheld devices use smaller, slower processors that consume less power.**
- **Therefore, the operating system and applications must be designed not to tax the processor.**

- **The last issue confronting program designers for handheld devices is I/O.**
- **A lack of physical space limits input methods to small keyboards, handwriting recognition, or small screen-based keyboards.**
- **The small display screens limit output options.**
- **Whereas a monitor for a home computer may measure up to 30 inches, the display for a handheld device is often no more than 3 inches square.**
- **Familiar tasks, such as reading e-mail and browsing web pages, must be condensed into smaller displays.**

- **One approach for displaying the content in web pages is web clipping, where only a small subset of a web page is delivered and displayed on the handheld device.**
- **Some handheld devices use wireless technology, such as Bluetooth or 802.11, allowing remote access to e-mail and web browsing.**
- **Cellular telephones with connectivity to the Internet fall into this category.**
- **However, for PDAs that do not provide wireless access, downloading data typically requires the user to first download the data to a PC or workstation and then download the data to the PDA.**

- Some PDAs allow data to be directly copied from one device to another using an infrared link.
- Generally, the limitations in the functionality of PDAs are balanced by their convenience and portability.
- Their use continues to expand as network connections become more available and other options, such as digital cameras and MP3 players, expand their utility.

- Alternatively referred to as an HPC, a handheld, handheld PC, or handheld computer is a computer device that can be held in the palm of one's hand.
- These computers are commonly used to store appointments, phone numbers, tasks, and other data commonly needed while away from home or office.
- In the picture is an example of a Palm Pilot handheld computer.
- The operating systems used for handheld computers differ from those used with a typical computer.
- Popular early handheld operating systems include Palm OS, Microsoft Windows CE, and other custom Linux-based operating systems.

- **Handheld systems are usually cellular phones with connection to a network like the Internet, bluetooth with limited size, small physical memory, slow processors, and small display screens.**
- **The memory size usually being between 512 KB and 8 MB, the OS needs to manage the memory efficiently and return the allocated memory back to the memory manager when not in use.**
- **These are usually used for sending emails and web browsing.**

Operating System services

Introduction

- An operating system provides certain services to programs and to the users of those programs.
- It provides an environment for the execution of programs.
- The services provided by one OS is different than other OS.
- These operating system services are provided for the convenience of the programmer, to make the programming task easier.

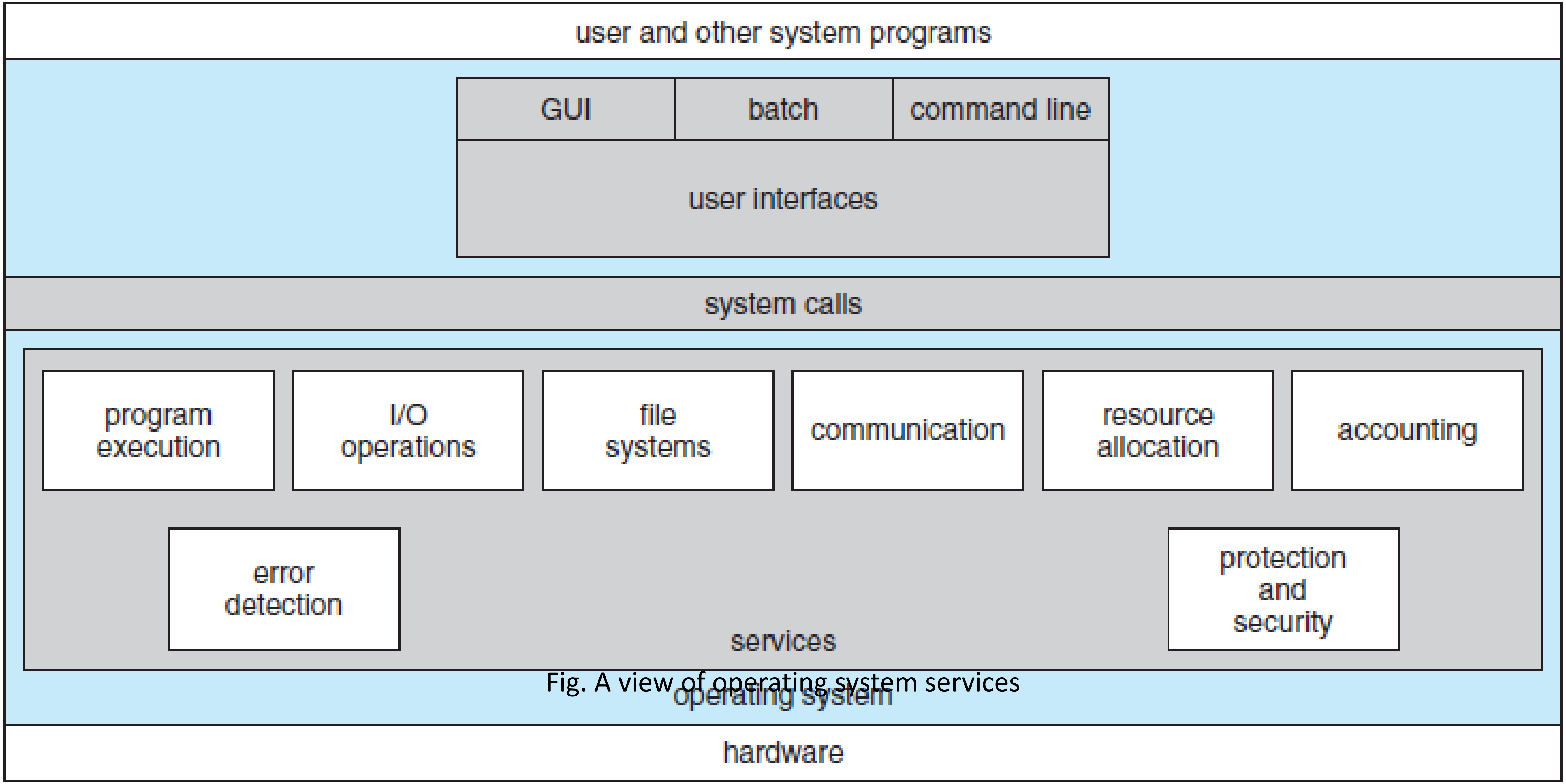


Fig. A view of operating system services

- **The common services provided by the operating system is listed below.**
- **A. User interface**
- **B. Program execution**
- **C. I/O operation**
- **D. File system manipulation**
- **E. Communications**
- **F. Error detection.**

- **Operating system with multiple users provide following services.**
- **G. Resource allocation**
- **H. Accounting**
- **I. Protection**

User interface

- the user interface is something that allows the user to interact with the operating system or to interact with the computer.
- Almost all operating systems have a user interface (UI).
- This interface can take several forms.
- One is a command-line interface (CLI), which uses text commands and a method for entering them.

- Another is a batch interface, in which commands and directives to control those commands are entered into files, and those files are executed.
- Most commonly, a graphical user interface (GUI) is used.
- 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.

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 an I/O device (or file).**
- **For efficiency and protection, users usually cannot control I/O devices directly.**
- **Therefore, the operating system must provide the required I/O.**

File-system manipulation

- While working on the computer, generally a user is required to manipulate various types of files like as opening a file, saving a file and deleting a file from the storage disk. This is an important task that is also performed by the operating system.
- The operating system gives the permission to the program for operation on file.
- Thus operating system makes it easier for the user programs to accomplish their task by providing the file system manipulation service.

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, or message passing, in which packets of information in predefined formats are moved between processes by the operating system.

Error detection

- **The operating system needs to be detecting and correcting errors constantly.**
- **Errors may occur in the CPU and memory hardware, in I/O devices, and in the user program.**
- **For each type of error, the operating system should take the appropriate action to ensure correct and consistent computing.**
- **Debugging facilities can greatly enhance the user's and programmer's abilities to use the system efficiently.**

- **Another set of operating system functions exists not for helping the user but rather for ensuring the efficient operation of the system itself.**
- **Systems with multiple users can gain efficiency by sharing the computer resources among the users.**
- **Operating system with multiple users provide following services.**
- **Resource allocation**
- **Accounting**
- **Protection**

Resource allocation

- When there are multiple users or multiple jobs running at the same time, , it is the responsibility of an operating system to allocate the required resources (like as CPU, main memory, tape drive or secondary storage etc.) to each process for its better utilization.
- For this purpose various types of algorithms are implemented such as process scheduling, CPU scheduling, disk scheduling etc.
- There may also be routines to allocate printers, USB storage drives, and other peripheral devices.

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.
- Usage statistics may be a valuable tool for researchers who wish to reconfigure the system to improve computing services.

Protection and security

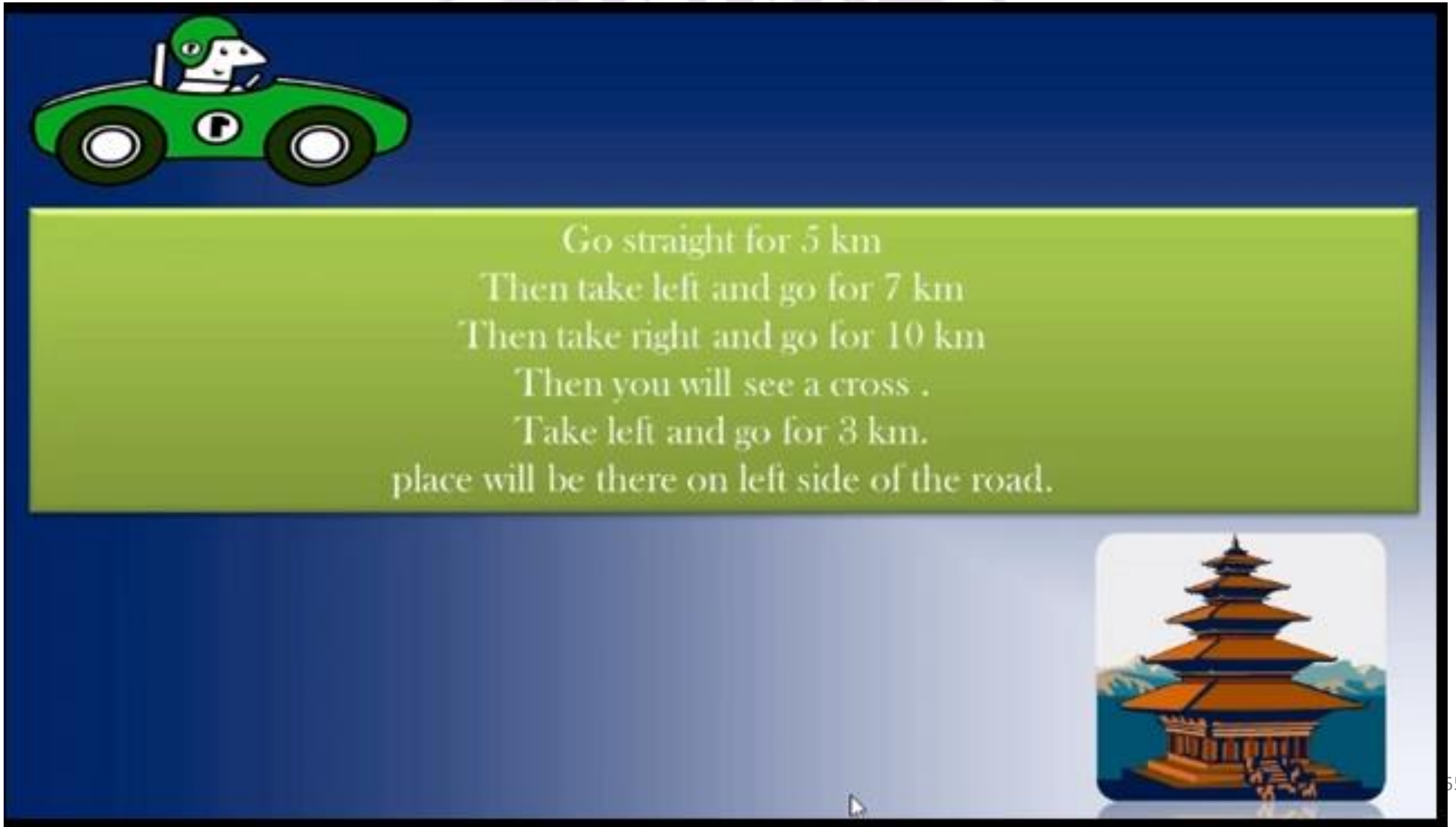
- This is to ensure the safety of the system. Thus, user authentication is required to access a system.
- It is also necessary to protect a process from another when multiple processes are running on a system at the same time.
- The OS controls the access to the resources, protects the I/O devices from invalid access, and provides authentication through passwords.
- Security of the system from outsiders is also important.
- Such security starts with requiring each user to authenticate himself or herself to the system, usually by means of a password, to gain access to system resources.

User OS Interface

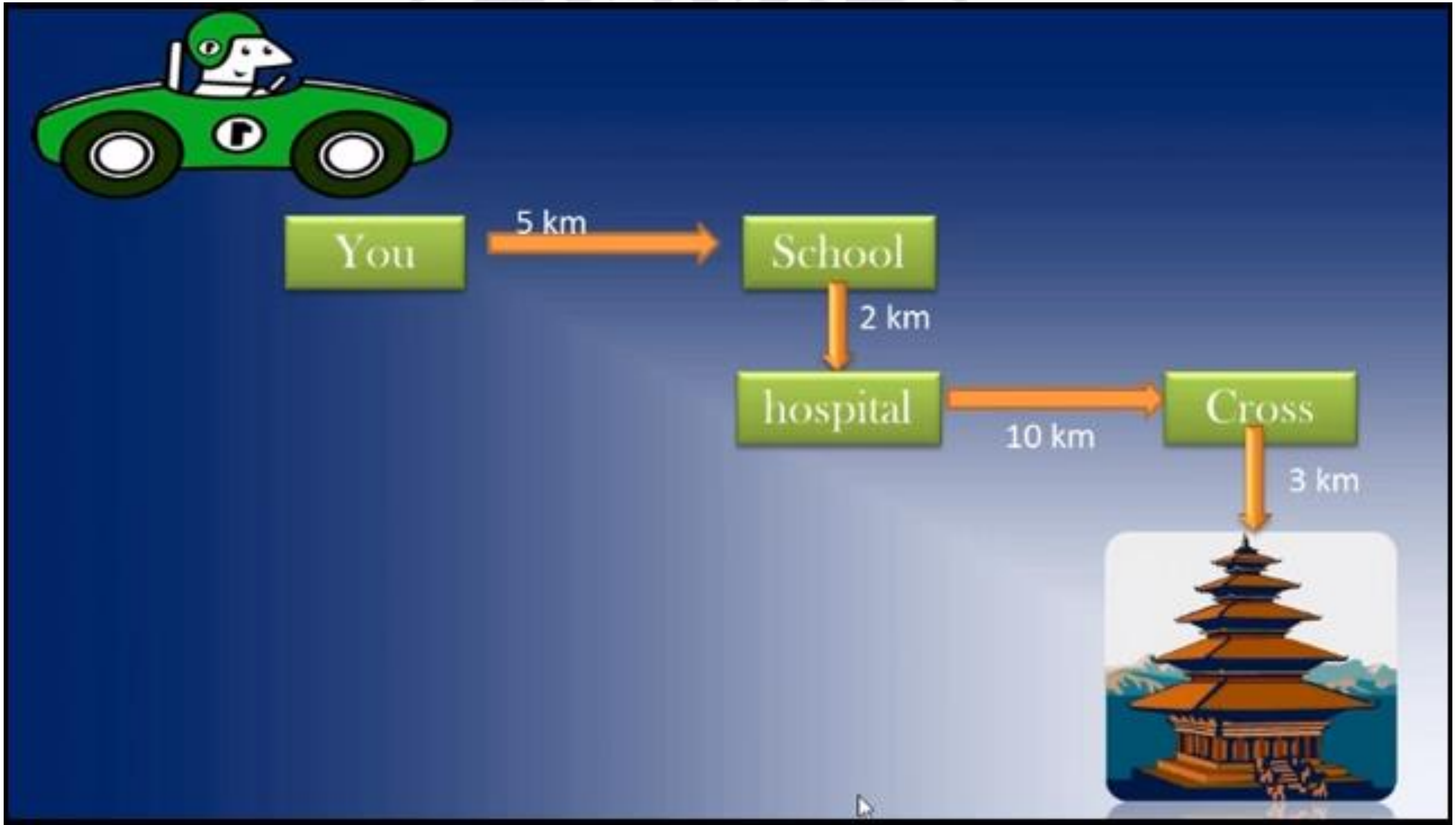
Introduction

- There are two fundamental approaches for users to interface with the operating system.
- One technique is to provide a command-line interface (CLI) or command interpreter (CI) or Character user interface (CUI) that allows users to directly enter commands that are to be performed by the operating system.
- The second approach allows the user to interface with the operating system via a graphical user interface(GUI).

Command-line interface (CLI)


Go straight for 5 km
Then take left and go for 7 km
Then take right and go for 10 km
Then you will see a cross .
Take left and go for 3 km.
place will be there on left side of the road.

Graphical user interface(GUI)



Command-line interface (CLI)/ Command Interpreter (CI)/ Character User Interface (CUI)

- **When we complete our works in a computer using only characters or commands then it is called command-line interface CLI or Character user interface (CUI).**
- **Ex: DOS, LINUX, UNIX-CUI**
- **Some operating systems include the command interpreter in the kernel.**
- **Others, such as Windows XP and UNIX, treat the command interpreter as a special program.**
- **On systems with multiple command interpreters to choose from, the interpreters are known as shells.**
- **Example: Bourne shell, C shell, Bourne-Again shell (BASH), the Korn shell, etc.**

- The **main function** of the command interpreter is to **get and execute the next user-specified command**.
- Many of the commands given at this level manipulate files: **create, delete, list, print, copy, execute**, and so on.
- The **MS-DOS and UNIX** shells operate in this way.
- There are **two general ways** in which these commands can be implemented.

- **In one approach, the command interpreter itself contains the code to execute the command.**
- **For example, a command to delete a file may cause the command interpreter to jump to a section of its code that sets up the parameters and makes the appropriate system call.**
- **In second approach, most commands are implemented through system programs. (used by UNIX)**

- In this case, the command interpreter does not understand the command in any way; it merely uses the command to identify a file to be loaded into memory and executed.
- Thus, the UNIX command to delete a file rm file.txt would search for a file called rm, load the file into memory, and execute it with the parameter file.txt.

- **The function associated with the rm command would be defined completely by the code in the file rm.**
- **In this way, programmers can add new commands to the system easily by creating new files with the proper names.**
- **The command-interpreter program, which can be small, does not have to be changed for new commands to be added.**

- ***Advantages:***
- **Simple structure**
- **Minimal memory usage**
- **Great for slow-running computers, or those low on memory**
- **An expert CLI user can give commands and perform tasks much faster than when using an alternative UI type**
- ***Disadvantages:***
- **Difficult to learn command language**
- **Complex for novice users**
- **Minimal error message information**
- **User needs to remember a lot to interact with the operating system.**

Graphical User Interfaces

- **A second strategy for interfacing with the operating system is graphical user interface or GUI.**
- **Rather than having users directly enter commands via a command-line interface, a GUI provides a mouse-based window-and-menu system as an interface.**
- **GUI contains several icons representing pictorial representation of the variables such as a file, directory, and device.**

- **The graphical icon provided in the UI can be manipulated by the users using a suitable pointing device such as a mouse, trackball, touch screen and light pen.**
- **Depending on the mouse pointer's location, clicking a button on the mouse can invoke a program, select a file or directory—known as a folder— or pull down a menu that contains commands.**
- **The first GUI appeared on the Xerox Alto computer in 1973.**

- **However, graphical interfaces became more widespread with the advent of Apple Macintosh computers in the 1980s.**
- **Microsoft's first version of Windows—version 1.0—was based upon a GUI interface to the MS-DOS operating system.**
- **Traditionally, UNIX systems have been dominated by command-line interfaces, although there are various GUI interfaces available, including the Common Desktop Environment (CDE) and X-Windows systems that are common on commercial versions of UNIX such as Solaris and IBM's AIX system.**

- **However, there has been significant development in GUI designs from various open source projects such as K Desktop Environment (or KDE) and the GNOME desktop by the GNU project.**
- **As a very general rule, many UNIX users prefer a command-line interface as they often provide powerful shell interfaces.**
- **Alternatively, most Windows users are pleased to use the Windows GUI environment and almost never use the MS-DOS shell interface.**

- **The user interface can vary from system to system and even from user to user within a system.**
- **GUIs are considered to be very user- friendly interface because each object is represented with a corresponding icon. Unlike the other UIs the users need not provide text command for executing tasks.**

Advantages

- **The GUI interface is easy to understand and even the new users can operate on them on their own.**
- **The GUI interface visually acknowledges and confirms each type of activities performed by the users.**
- **Example: when the user deletes a file in the Windows operating system, then the operating system asks for the confirmation before deleting it.**
- **The GUI interface enables the users to perform a number of tasks at the same time. This features of the operating system are also known as multitasking.**
- **Memorizing command lists is not necessary**
- **Allows for running multiple applications, programs, and tasks simultaneously**

Disadvantages:

- **Uses large amounts of memory – although this is less of a concern as computers get more powerful**
- **Ex: WINDOWS, MAC OS-GUI**

Menu-Driven Interface

- **The menu-driven user interface provides you with a range of commands or options in the form of a list or menu displayed in full-screen, pop-up, pull-down, or drop-down.**
- **An ATM is an example of a menu-driven interface.**

- ***Advantages:***
- It is not necessary to remember a long list of manual commands
- Simple interface for novices
- Self-explanatory menu options
- ***Disadvantages:***
- Slower for experienced users
- Limited menu options
- Often requires you to access multiple menu screens to perform simple functions.

Touchscreen Graphical User Interface

- The touchscreen GUI is very similar to the regular GUI, except that you use your fingers or a stylus to select icons and perform tasks, rather than a mouse or trackpad.
- Touchscreen GUIs are commonly found on tablets, smartphones, and medical devices, like the t:slim insulin pump.
- The touchscreen GUI has the same benefits and disadvantages as standard GUIs, but also offers a more intimate method of interaction.
- The lack of peripherals makes touchscreen GUIs very convenient.

- **Of the four types of user interface, the graphical user interface is by far the most common, followed by the touchscreen variation.**
- **Despite the alternative technologies that already exist and continue to emerge, the GUI remains the preferred standard.**
- **This is largely due to the simplicity and ease of use.**

- **Graphical user interfaces are easier for most end users to understand as the icons and menus are generally self-explanatory and the GUI does not require the user to remember or input complex commands.**
- **While they do take up considerable memory space compared to other UIs, this is a secondary concern as devices continue to have larger, more efficient storage than their predecessors.**

Command Line User Interface (CUI)	Graphical User Interface (GUI)
Difficult to use	**Easy to use**
Needs memorization of commands	**No need**
Limitations for multimedia components	**Suitable for multimedia**
Working in CUI is boring	**Interesting and eco friendly**
More secure	**Less secure**
Faster	**Slower**
Efficient memory utilization	**Less**

System Calls

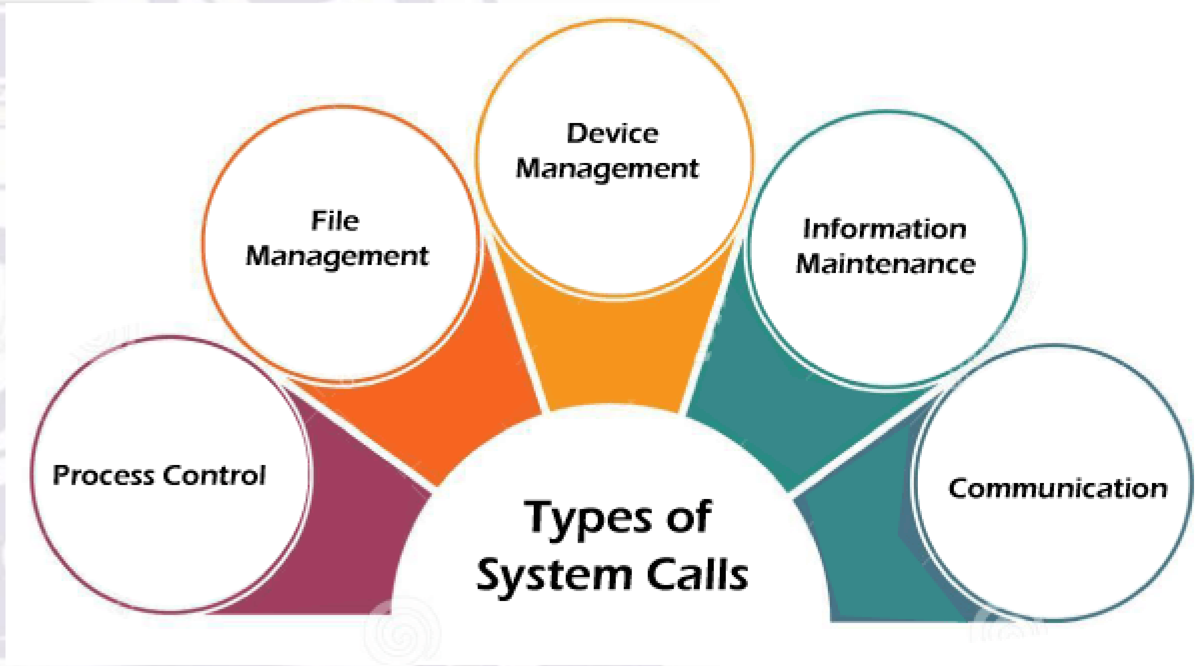
Introduction

- When a program in user mode requires access to RAM or a hardware resource, it must ask the kernel to provide access to that resource. This is done via something called a system call.
- When a program makes a system call, the mode is switched from user mode to kernel mode. This is called a context switch.
- Then the kernel provides the resource which the program requested. After that, another context switch happens which results in change of mode from kernel mode back to user mode.

- **Generally, system calls are made by the user level programs in the following situations:**
- **Creating, opening, closing and deleting files in the file system.**
- **Creating and managing new processes.**
- **Requesting access to a hardware device, like a mouse or a printer.**
- **In a typical UNIX system, there are around 300 system calls.**

Types of System Calls

- There are 5 different categories of system calls:
- process control,
- file manipulation,
- device manipulation,
- information maintenance, and
- communication.



Process Control

- **A running program needs to be able to stop execution either normally or abnormally.**
- **When execution is stopped abnormally, often a dump of memory is taken and can be examined with a debugger.**

File Management

- **Some common system calls are *create*, *delete*, *read*, *write*, *reposition*, or *close*.**
- **Many times the OS provides an API to make these system calls.**

Device Management

- **Process usually require several resources to execute, if these resources are available, they will be granted and control returned to the user process.**
- **User programs *request* the device, and when finished they *release* the device.**
- **Similar to files, we can *read*, *write*, and *reposition* the device.**

Information Management

- **Some system calls exist purely for transferring information between the user program and the operating system.**
- **An example of this is *time*, or *date*.**

Communication

- **There are two models of inter-process communication, the message-passing model and the shared memory model.**
- **Message-passing uses a common mailbox to pass messages between processes.**
- **Shared memory use certain system calls to create and gain access to regions of memory owned by other processes.**

System Programs

Introduction

- System programs provide an environment for program development and execution.
- It also provide a bridge between the user interface and system calls.
- Some of the System Programs are simply user interfaces, others are complex.

- **According to Computer Hierarchy, one which comes at last is Hardware. Then it is Operating System, System Programs, and finally Application Programs.**
- **Ex: a compiler is a complex system program.**

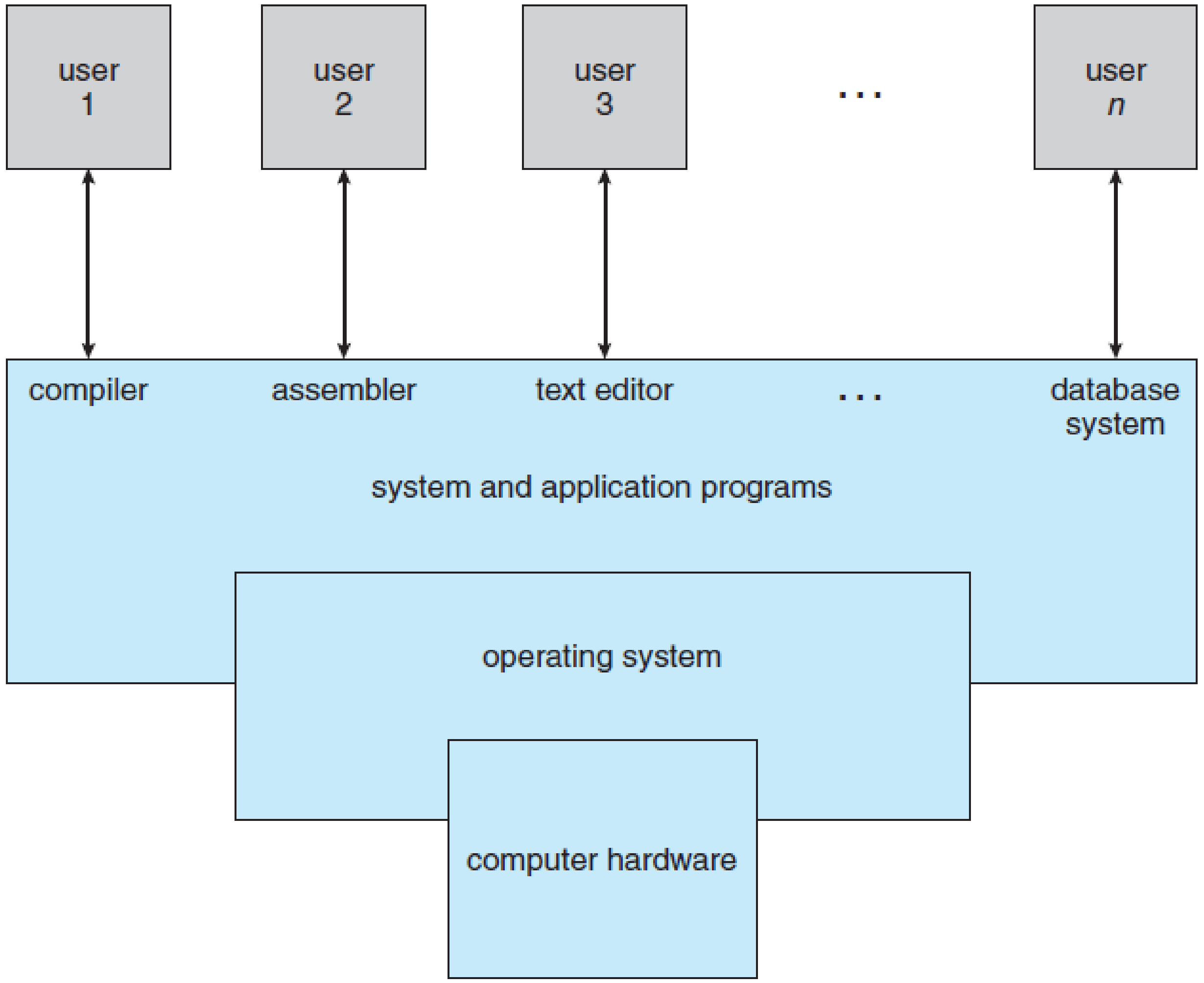
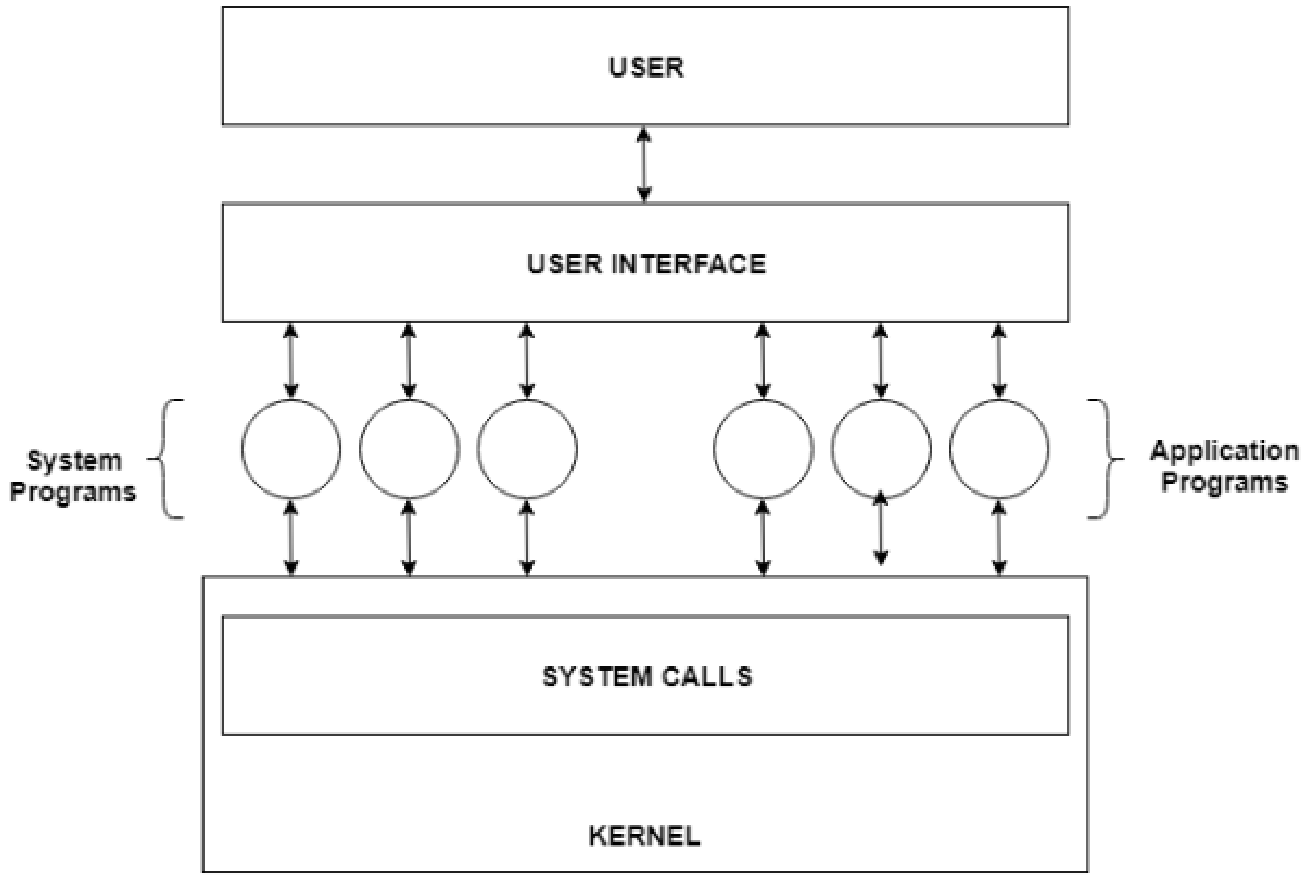


Figure 1.1 Abstract view of the components of a computer system.

System Programs Purpose

- The system program serves as a part of the operating system.
- It lies between the user interface and the system calls.
- The **user view of the system is actually defined by system programs.**
- System Programs can be divided in to different categories.

- **In the above image, system programs as well as application programs form a bridge between the user interface and the system calls.**
- **So, from the user view the operating system observed is actually the system programs and not the system calls.**



Operating System Design and Implementation

Introduction

- An operating system is a construct that allows the user application programs to interact with the system hardware.
- Operating system by itself does not provide any function but it provides an atmosphere in which different applications and programs can do useful work.
- There are many problems that can occur while designing and implementing an operating system.

- It is quite complicated to define all the goals and specifications of the operating system while designing it.
- The design changes depending on the type of the operating system.

- There are basically two types of goals while designing an operating system.
- User Goals
- The operating system should be convenient, easy to use, reliable, safe and fast according to the users.
- However, these specifications are not very useful as there is no set method to achieve these goals.

- System Goals
- The operating system should be easy to design, implement and maintain.
- These are the specifications required by those who create, maintain and operate the operating system.
- But there is not specific method to achieve these goals as well.

Operating System Mechanisms and Policies

- There is **no specific way to design an operating system** as it is a highly creative task.
- However, there are **general software principles that are applicable to all operating systems.**
- A subtle difference between **mechanism and policy** is that **mechanism shows how to do something** and **policy shows what to do.**
- **Policies may change over time and this would lead to changes in mechanism.**

- So, it is better to have a general mechanism that would require few changes even when a policy change occurs.
- For example - If the mechanism and policy are independent, then few changes are required in mechanism if policy changes.

Operating System Implementation

- The operating system needs to be implemented after it is designed.
- Earlier they were written in assembly language but now higher level languages are used.
- The first system not written in assembly language was the Master Control Program (MCP) for Burroughs Computers.

Advantages of Higher Level Language

- the code can be written faster,
- it is more compact
- It is easier to debug and understand.
- The operating system can be easily moved from one hardware to another if it is written in a high level language.

Disadvantages of Higher Level Language

- Using high level language for implementing an operating system leads to a loss in speed and increase in storage requirements.
- However in modern systems only a small amount of code is needed for high performance, such as the CPU scheduler and memory manager.

- It **takes additional translation** times to translate the source to machine code.
- High level programs are comparatively **slower than low level programs.**
- Compared to low level programs, they are generally **less memory efficient.**
- **Cannot communicate directly with the hardware.**

Virtual machines

Introduction

- A virtual machine (VM), is no different than any other physical computer like a laptop, smart phone or server.
- A virtual machine (VM) is a virtual environment that functions as a virtual computer system with its own CPU, memory, network interface, and storage, created on a physical hardware system (located off- or on-premises).

- **The fundamental idea behind a virtual machine is to abstract the hardware of a single computer into several different execution environments, thereby creating the illusion that each separate environment is running on its own private computer.**

- **Virtual machine implementations involve several components.**
- **The Virtual Machine is connected with the hardware by an intermediary software called Hypervisor.**
- **The machine, where Hypervisor is installed, is called the host machine/host; whereas, the machine where Virtual Machine utilizes the host machine’s resources and Hypervisor to connect to hardware is called guest machine/guest.**

- **Virtual Machine is wholly isolated from the host and operates as a completely independent entity.**
- **There are two types of Virtual Machines:**
- **System Virtual Machine: used to emulate the entire Operating System.**
- **Process Virtual Machine: used to emulate a process independent from the host environment.**

Hypervisor

- A Hypervisor acts as a bridge between the virtualized OS and the hardware.
- It pools the hardware resources and allocates them to the host and the guest.
- Some of the components that Hypervisor needs to do its job are memory manager, process scheduler, I/O stack, device drivers, security manager, and network stack.
- Hypervisor provides an abstraction that lets you run multiple Operating Systems or Processes on one machine.

Advantages

- **Allows multiple operating system environments on a single physical computer without any intervention.**
- **Virtual machines are widely available and are easy to manage and maintain.**
- **A VM can be created or replicated very quickly by cloning it with an OS already installed, rather than installing a new OS on a physical server.**
- **VMs offer high availability since they can be moved from one server to another for maintenance purposes, even whilst running.**
- **Fewer hardware components needed to operate multiple Operating Systems**

Disadvantages

- They are not as efficient as a physical computer because the hardware resources are distributed in an indirect way.
- Multiple VMs running on a single physical machine can deliver unstable performance.
- Running multiple Operating System on one computer/server imposes a toll on its components and uses up more memory and CPU.
- Since hardware is shared across all Virtual Machines, failure of a component will have an adverse effect on all the machines

Thank You