Operating Systems
BCSC 0004

Introduction to Operating Systems

Books

Operating System Concepts
Ninth Edition
Avi Silberschatz Peter Baer Galvin Greg Gagne

Operating Systems: Internals and Design Principles
William Stallings

BCSC0004: OPERATING SYSTEMS

Objective: *The objective of the course is to provide basic knowledge of computer operating system structures and functioning including CPU scheduling, memory management, concurrent processes, deadlocks, security, and integrity.*

Credits:03 **L-T-P-J:3-0-0-0**

Module No.	Content	Teaching Hours
I	**Introduction:** Operating System and its Classification - Batch, Interactive, Multiprogramming, Time sharing, Real Time System, Multiprocessor Systems, Multithreaded Systems, System Protection, System Calls, Reentrant Kernels, Operating System Structure- Layered structure, Monolithic and Microkernel Systems, Operating System Components, Operating System Functions and Services. **Processes:** Process Concept, Process States, Process State Transition Diagram, Process Control Block (PCB), Process Scheduling Concepts, Threads and their management. **CPU Scheduling:** Scheduling Concepts, Performance Criteria, Scheduling Algorithms, Multiprocessor Scheduling.	14
II	**Process Synchronization:** Principle of Concurrency, Implementation of concurrency through fork/join and parbegin/parend, Inter Process Communication models and Schemes, Producer / Consumer Problem, Critical Section Problem, Dekker's solution, Peterson's solution, Semaphores, Synchronization Hardware. **Classical Problem in Concurrency:** Dining Philosopher Problem, Readers Writers Problem. **Deadlock:** System model, Deadlock characterization, Prevention, Avoidance and detection, Recovery from deadlock, Combined Approach.	13
III	**Memory Management:** Multiprogramming with fixed partitions, Multiprogramming with variable partitions, Paging, Segmentation, Paged segmentation. **Virtual memory concepts:** Demand paging, Performance of demand paging, Page replacement algorithms, Thrashing, Locality of reference. **I/O Management and Disk Scheduling:** I/O devices, I/O subsystems, I/O buffering, Disk storage and disk scheduling. **File System:** File concept, File organization and access mechanism, File directories, File allocation methods, Free space management.	13

Text Books:

- Silberschatz, Galvin and Gagne (2012), *"Operating Systems Concepts"*,9th Edition, Wiley.

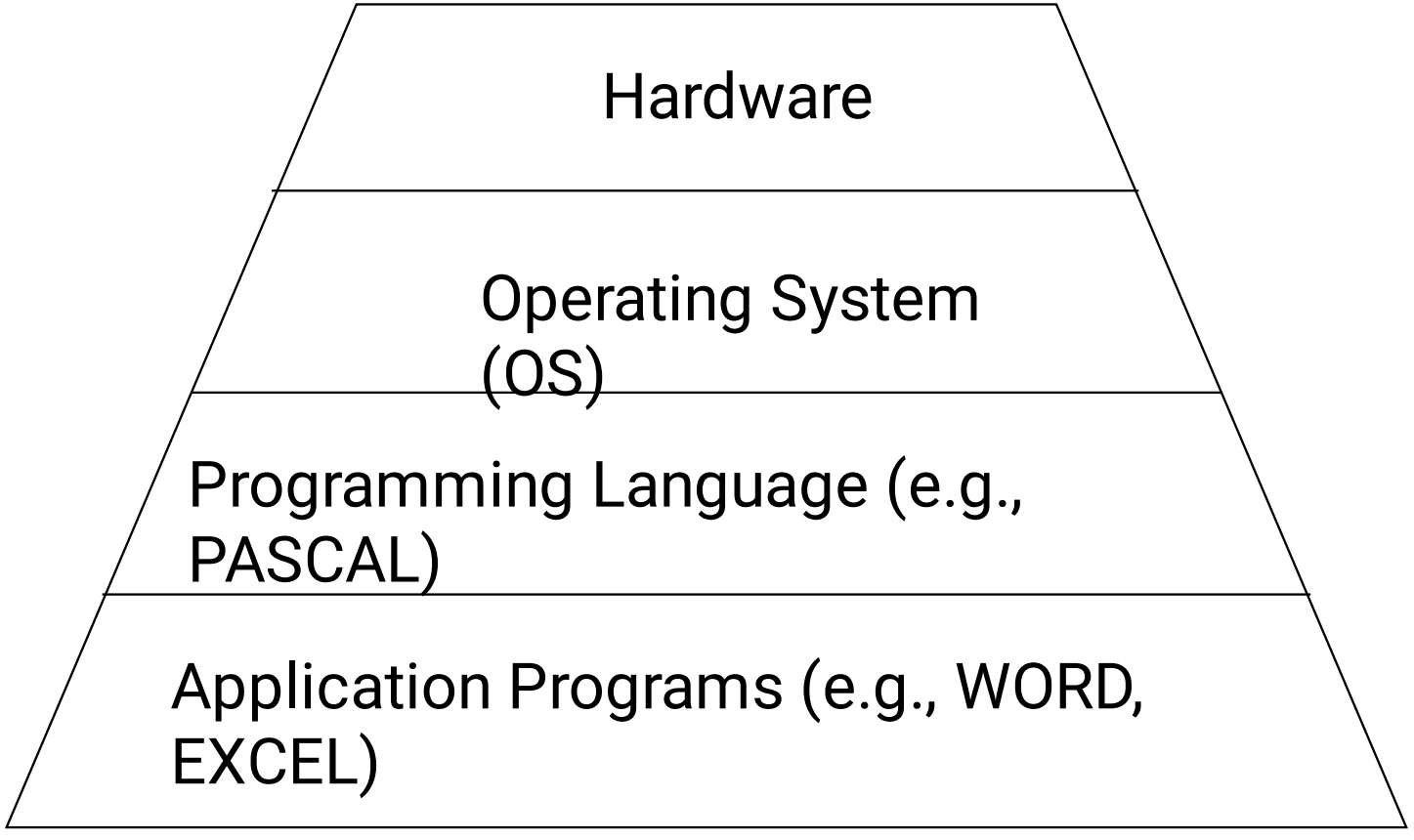
Reference Books:

- SibsankarHalder and Alex A Aravind (2009), *" Operating Systems"*, 6th Edition, Pearson Education.
- Harvey M Dietel (2002), *"An Introduction to Operating System"*, 2nd Edition, Pearson Education.
- D M Dhamdhere (2006), *"Operating Systems: A Concept Based Approach"*, 2nd Edition.
- M. J. Bach. (1986), *"Design of the Unix Operating System"*, PHI.

Outcome: After completion of course, the student will be able to:

- Identify the services provided by operating systems.
- Understand the internal structure of an operating system and be able to write programs using system calls.
- Understand and solve problems involving process control, mutual exclusion, deadlock and synchronization.

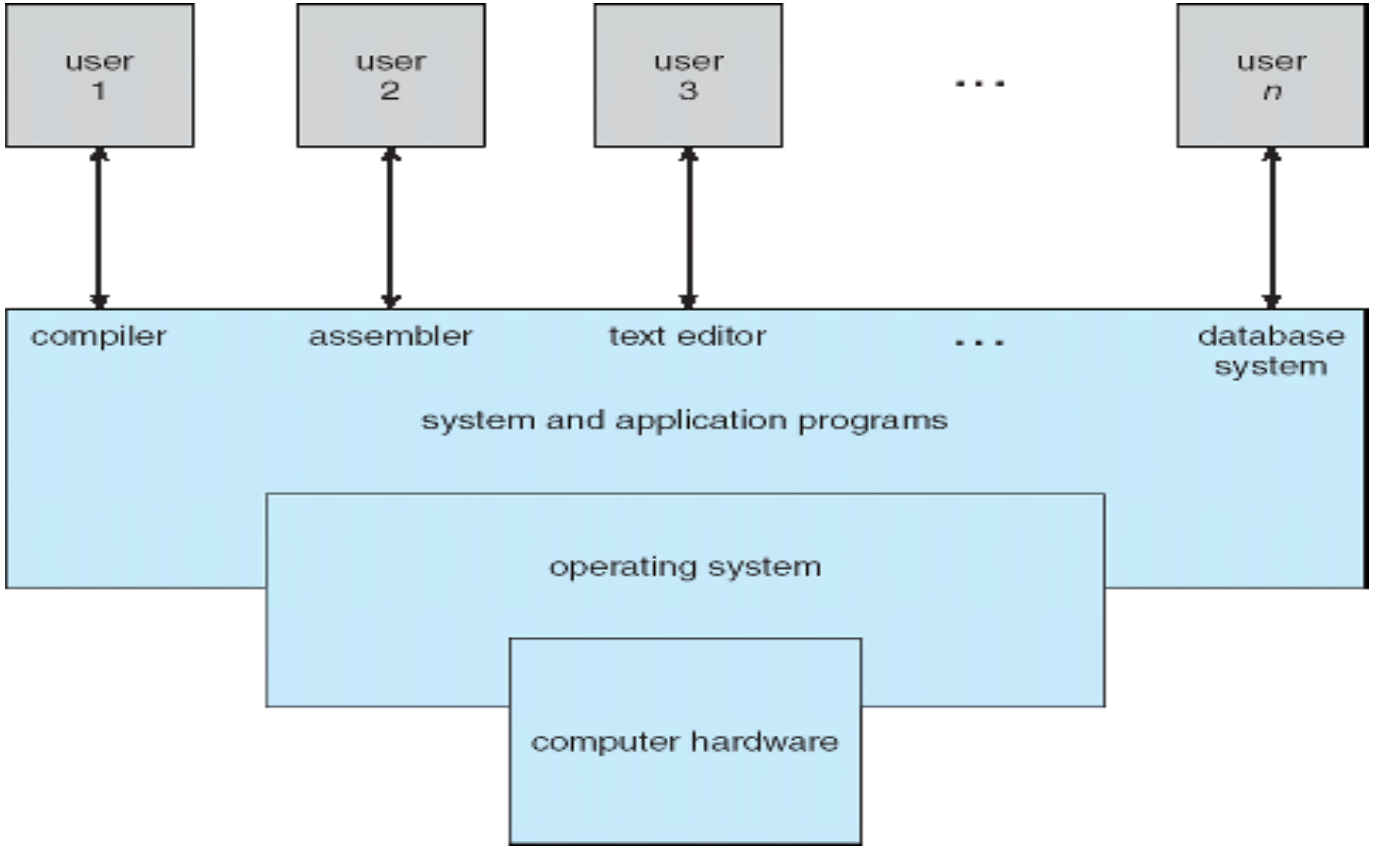
Architecture of Computer System



Computer System Structure

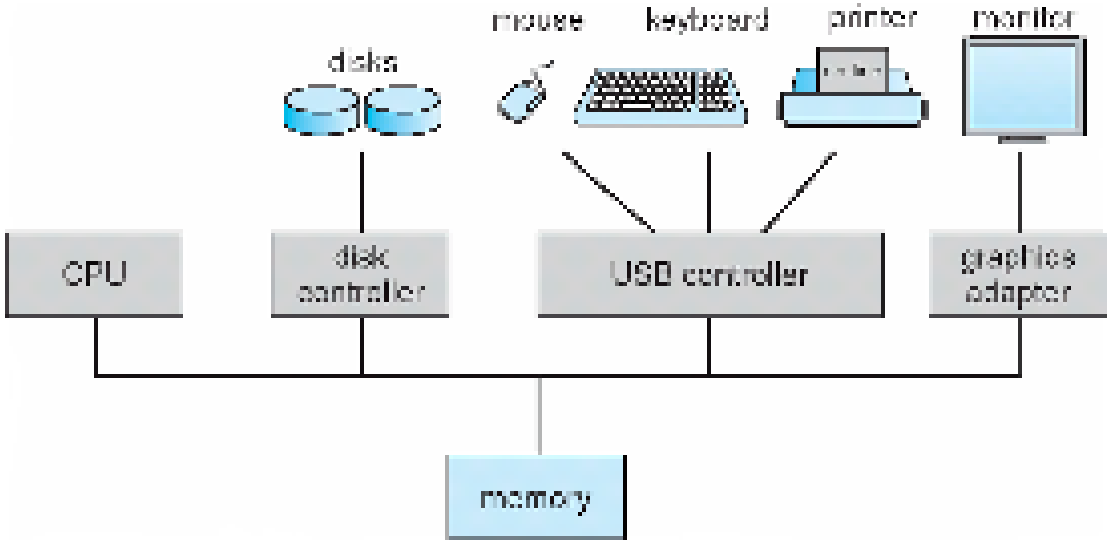
- Computer system can be divided into four components:
  - Hardware – provides basic computing resources
    - CPU, memory, I/O devices
  - Operating system- Controls and coordinates use of hardware among various applications and users
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users
    - Word processors, compilers, web browsers, database systems, video games
  - Users
    - People, machines, other computers

Four Components of a Computer System



Computer System Organization

- Computer-system operation
  - One or more CPUs, device controllers connect through common bus providing access to shared memory
  - Concurrent execution of CPUs and devices competing for memory cycles



History of operating system

- First generation system (1945-55)
  - Vacuum tubes & plug board, No operating system
- Second generation (1955-65)
  - Batch processing system, First OS was written for Intel
  - Used mostly for scientific & Engineering calculations.
- Third generation (1965-75)
  - Batch processing system, but several jobs could run at a time, multiprogramming.
- Fourth generation (1975-present)
  - PC's came to begin, Desktop system as powerful as mainframe of 1970's
  - Two OS are dominant MS-DOS, UNIX

Bootstrapping

- Booting is the process of loading operating system into main memory.
- For a computer to start running, it needs to have an initial program to load and execute the boot program, which in turn loads the operating system.
- The primitive loader program that can load and execute the Boot program is called Bootstrap Program.
- Boot strap program is generally stored in ROM.
- On- Start up, the computer automatically reads the **bootstrap program**.

Definition of OS

- A program that acts as an intermediary between a user of a computer and the computer hardware.
- Conveniently and efficiently used hardware .
- Manage resources in an unbiased fashion both hardware and software.
- Provide a platform on which other application programs run.

Goal and Functions

- Execute user programs and make solving user problems easier
- Make the computer system convenient to use
- Use the computer hardware in an efficient manner

Function of OS

- Process Management
- Memory Management
- I/O device Management
- Network Management
- Security and Protection

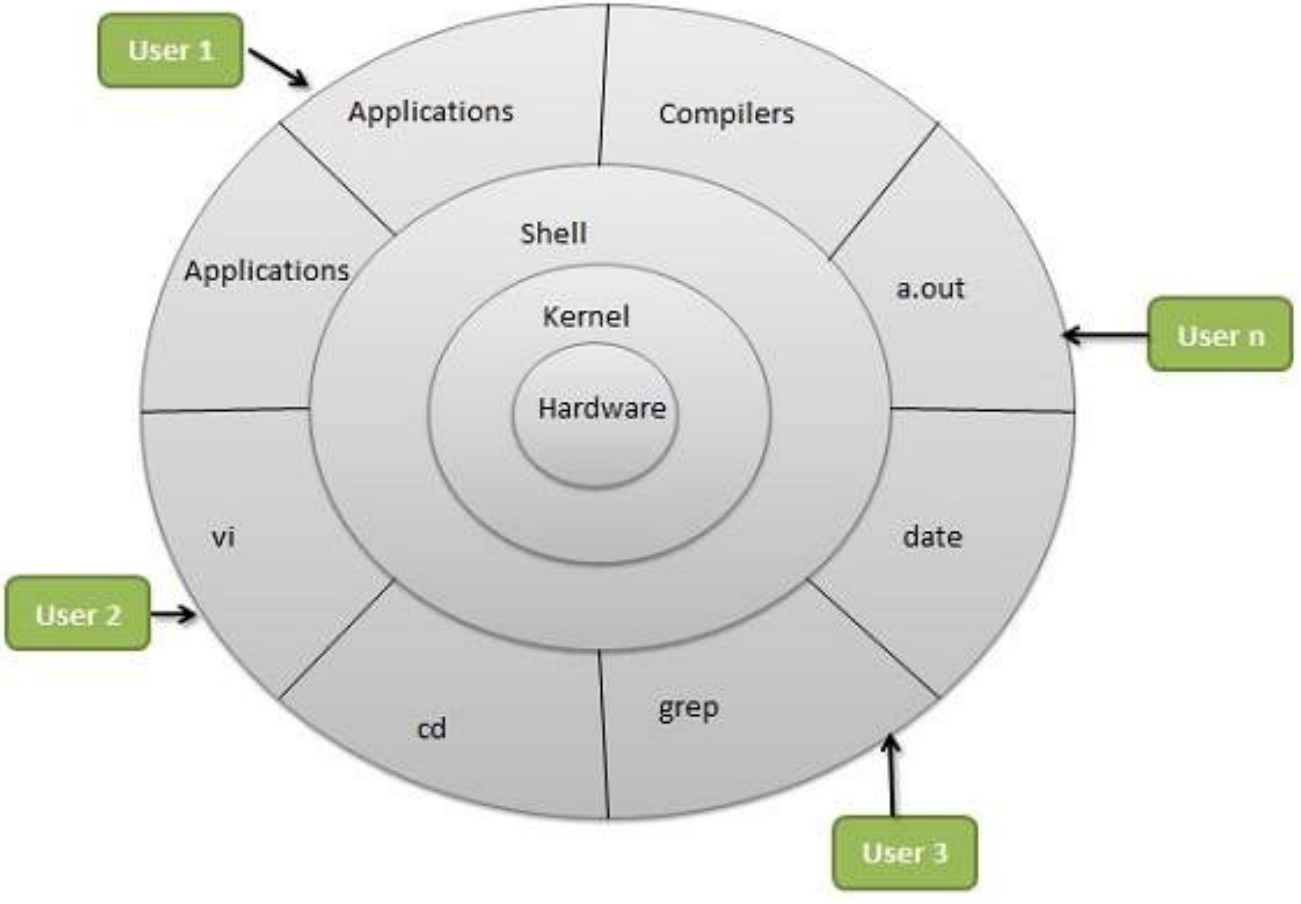
Function of OS

- Operating system are resource managers. The main resource in computer hardware in the form of
  - Processor Storage
  - I/O device Communication device

- **Some imp. OS function are**
  - Implementing user interface
  - Sharing hardware among users
  - Allowing users to share data among themselves.
  - Preventing users from interfering with one another.
  - Scheduling resources among users.

Kernel in Operating System

- Kernel is central component of an operating system that manages operations of computer and hardware.
- It basically manages operations of memory and CPU time.
- Kernel acts as a bridge between applications and data processing performed at hardware level using inter-process communication and system calls.
- It is responsible for various tasks such as disk management, task management, and memory management.


User 1
Applications
Compilers
Applications
Shell
Kernel
Hardware
a.out
User n
vi
date
User 2
cd
grep
User 3

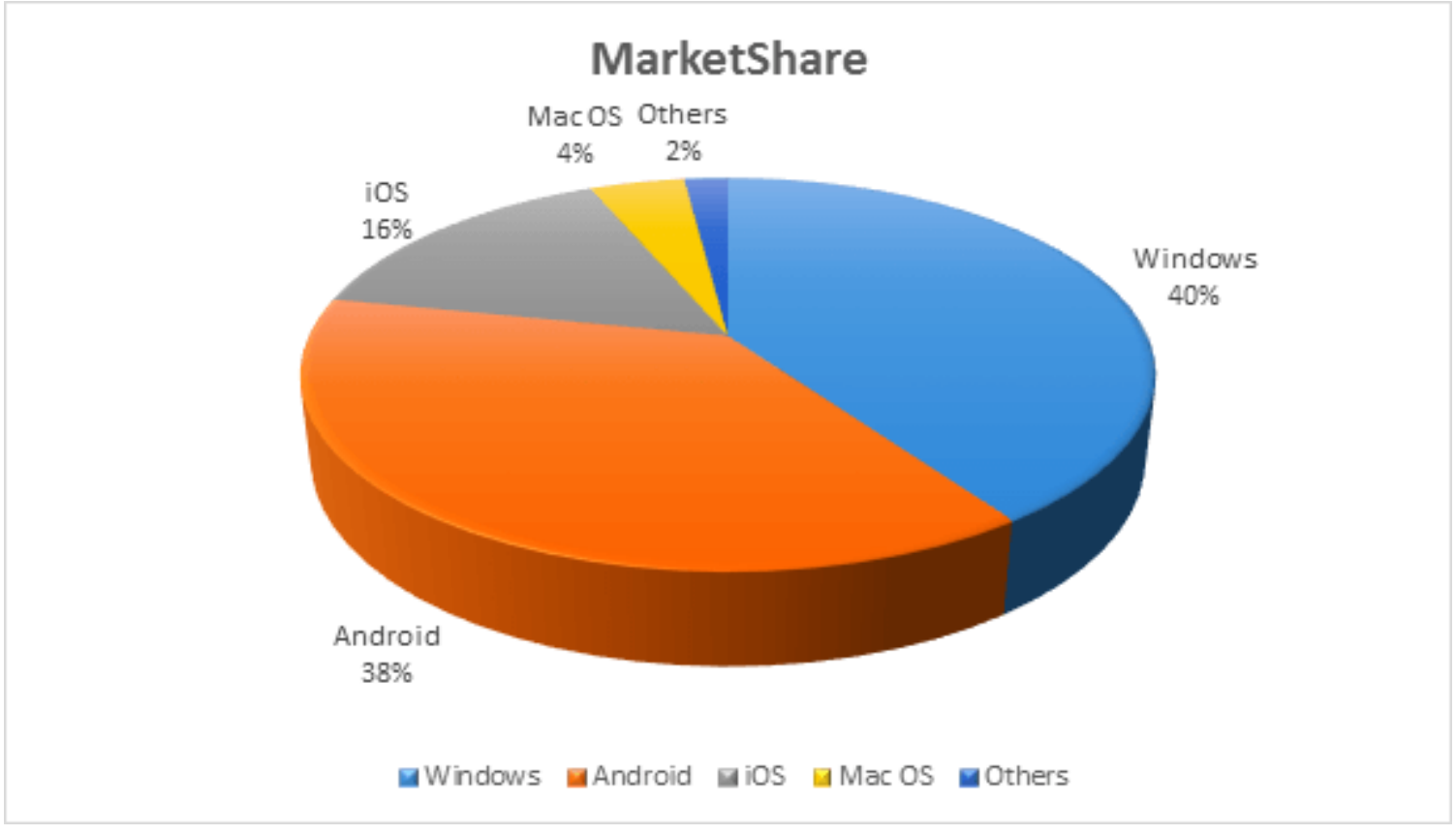
Types of Kernel

Monolithic kernel: It is one of types of kernel where all operating system services operate in kernel space. It has dependencies between systems components. It has huge lines of code which is complex. Ex. Unix, Linux

Micro Kernel: It has minimalist approach. It has virtual memory and thread scheduling. It is more stable with less services in kernel space. It puts rest in user space. Ex. Minix, AmigaOS

Hybrid Kernel: It is the combination of both monolithic kernel and mircro-kernel. It has speed and design of monolithic kernel and modularity and stability of microkernel. Ex Windows OS

Operating System with Market Share

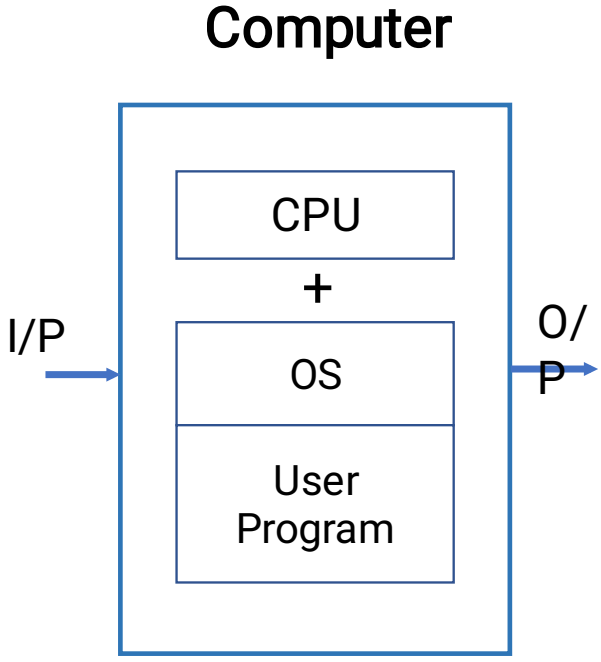


Classification of Operating System

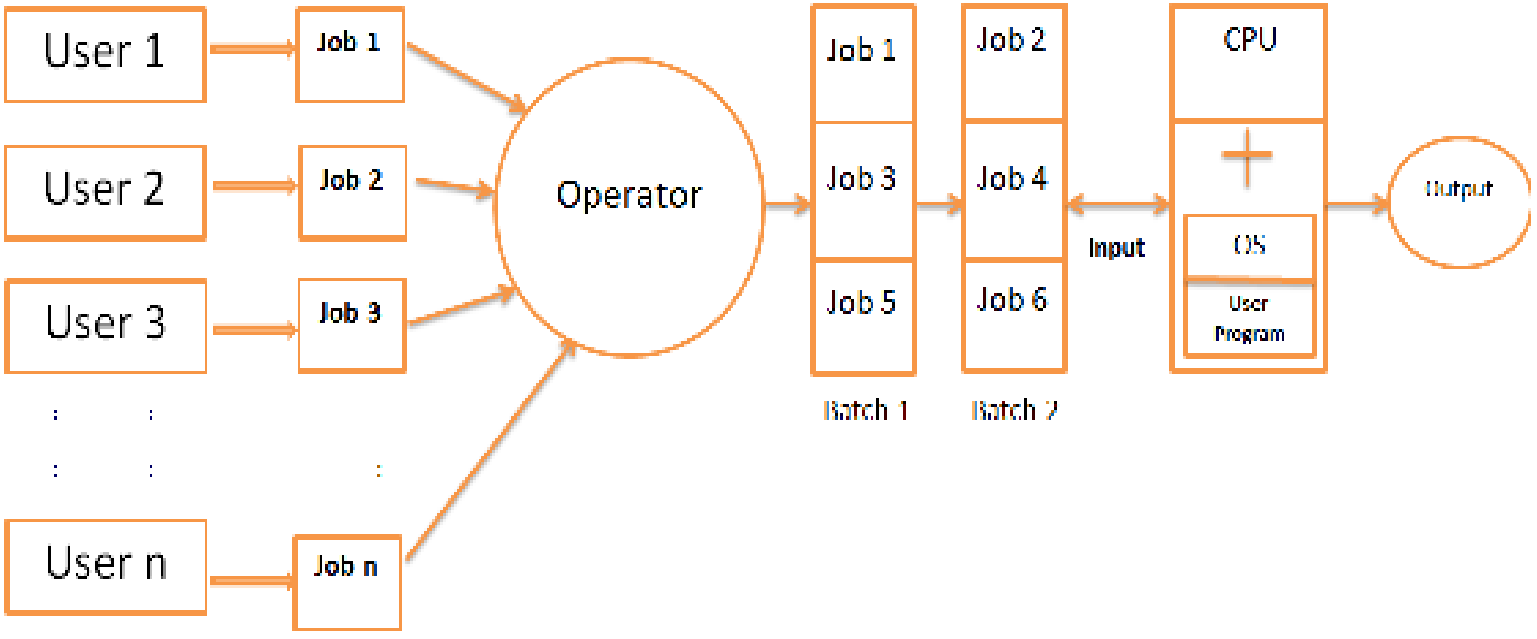
- Batch Operating System
- Multiprogramming Operating System
- Multitasking/Time Sharing OS
- Multiprocessing OS
- Real Time OS
- Distributed OS
- Network OS
- Mobile OS

Batch Processing OS

- In starting mainframe computers
- Common I/P and O/P devices were card reader and tape drive.
- User prepare a job which consisted of the program I/P data and control instruction.
- I/P job given in the form of punch cards and result also appear in form of punch card after processing.
- So, OS was very simple, always present in memory major task is to transfer control from one job to another job.



Batch Operating System

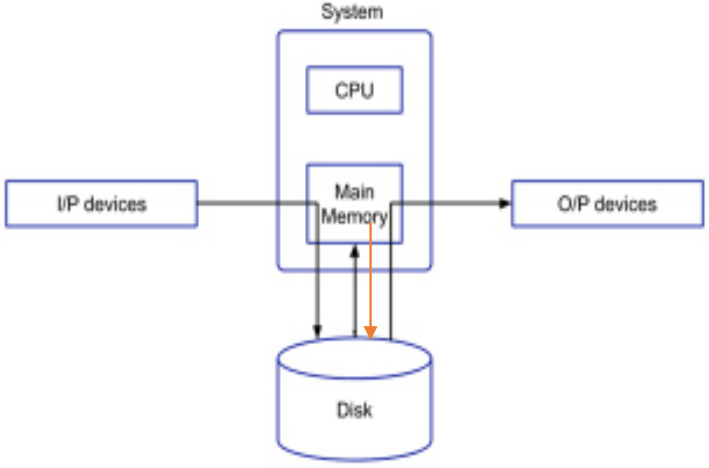


Batch Operating System

- Jobs with similar needs are batched together and execute through the processor as a group.
- Operator sort jobs as a deck of punch cards into batch with similar needs. Ex. FORTRAN batch, COBOL batch etc.
- **Advantage:** In batch operating, job execute one after another, saving time from activities like loading and unloading the compiler.
- **Disadvantage:** Priority can not be set for the jobs
  - CPU may remain idle for a long time
  - Lack of interaction between user and his job

Spooling (Simultaneous Peripheral Operation On-Line)

- I/P and O/P devices are relatively slow compare to CPU (Digital)
- In spooling data is stored first and the disk and then CPU interacts with disk (digital) via main memory
- Spooling is capable of overlapping I/O operations for one job with CPU operations of other jobs.
- Maintains parallel computation because of spooling process as a computer can perform I/O in parallel fashion.



Spooling (Simultaneous Peripheral Operation On-Line)

- **Advantages of Spooling:-**

➢The spooling operation uses a disk in which data is stored first and then the CPU interact with main memory for fetching the data.

➢Spooling is capable of overlapping input/output operation for one job with CPU operation of other jobs.

➢In this OS CPU utilization is more as CPU is busy most of the time.

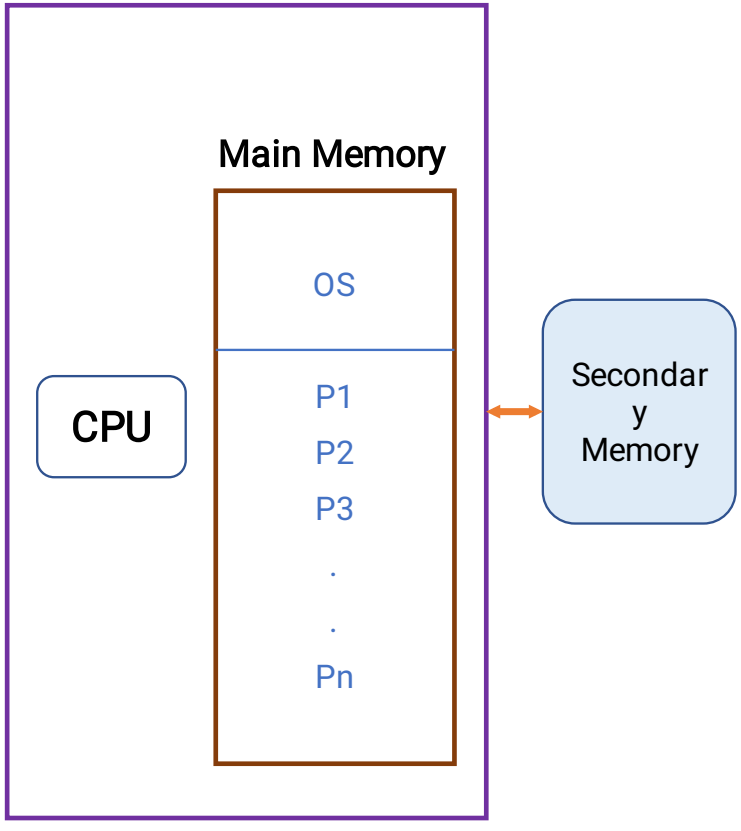
- **Disadvantage of Spooling:-**

➢The main disadvantage of Spooling is that it is very difficult to debug.

➢Due to lack of protection scheme, one batch job can affect the pending jobs

Multiprogramming Operating System

- Maximize CPU utilization
- Multiprogramming means more than one process in main memory which are ready to execute.
- Process generally requires CPU time and I/O time so if running process perform I/O or some other event which do not require CPU then instead of sitting idle CPU make a context switch and execute some other process and this idea will continue.
- CPU never idle unless that is not process ready to execute or at time of context switch.



Multiprogramming Operating System

❑**Advantage:**

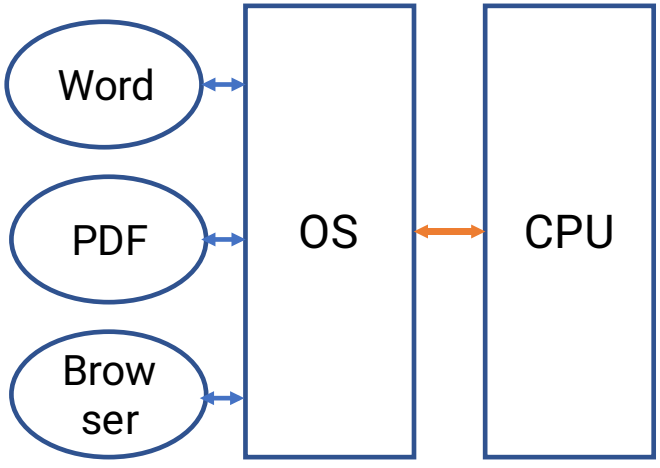
- ➢High CPU utilization
- ➢Less waiting time, response time
- ➢May be executed to multiple users
- ➢Now a days useful then load is more

❑**Disadvantage:**

- ➢Difficult scheduling
- ➢Main memory management is required.
- ➢Memory fragmentation
- ➢Paging (Non contiguous memory allocation)

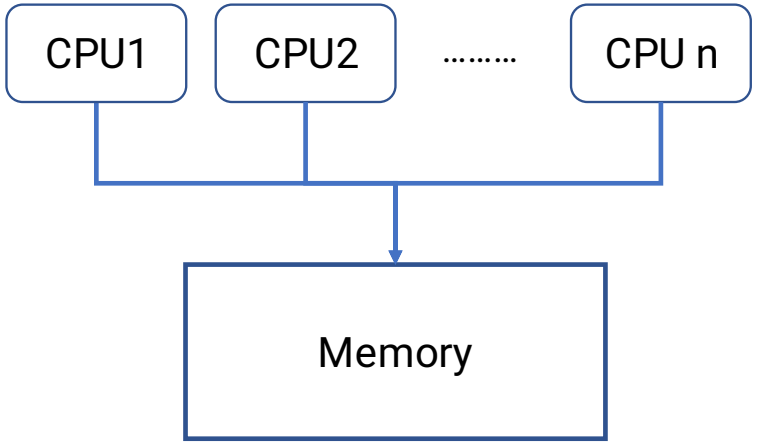
Multitasking / Time-sharing / Multiprogramming with RR Operating System

- Time sharing (Multitasking) is a logical extension of multiprogramming.
- Only one CPU but switches between process quickly that is given an illusion that all executing at same time.
- The task in multitasking may refer to multiple threads of the same program.
- Main idea is better response time and executing multiple process together.



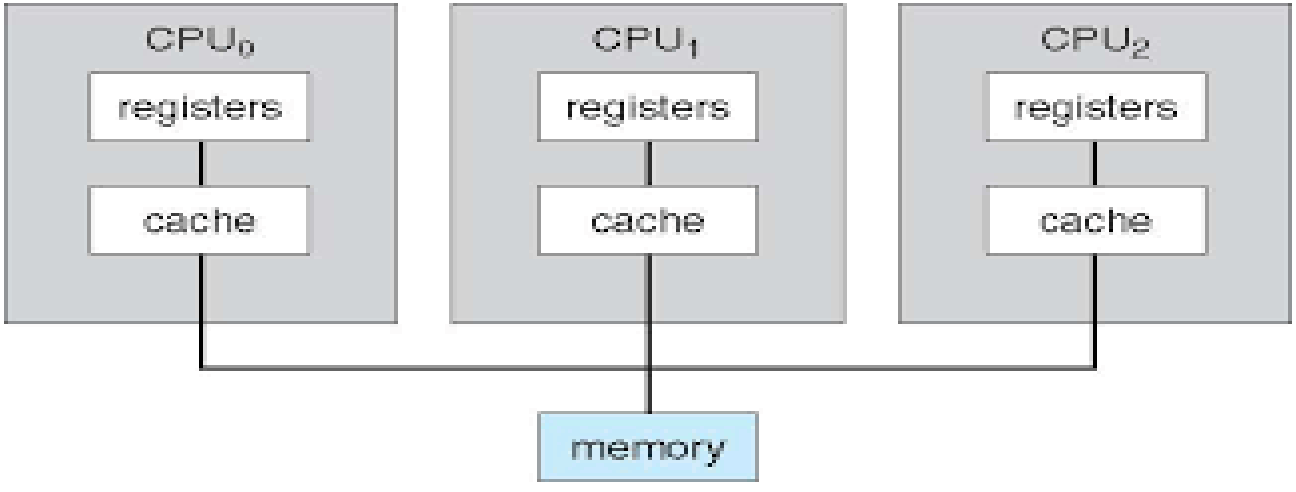
Multiprocessing Operating System

- Two or more CPU with in a single computer, in close communication sharing the system bus, memory and other I/O devices.
- Different process may run on different CPU, true parallel execution.
- Symmetric:- One OS control all CPU, each CPU has equal rights.
- Asymmetric:- A master slave architecture, system task on one processor and application on other as one CPU will handle all H/W interrupts as I/O devices, they are easy to design but less efficient.



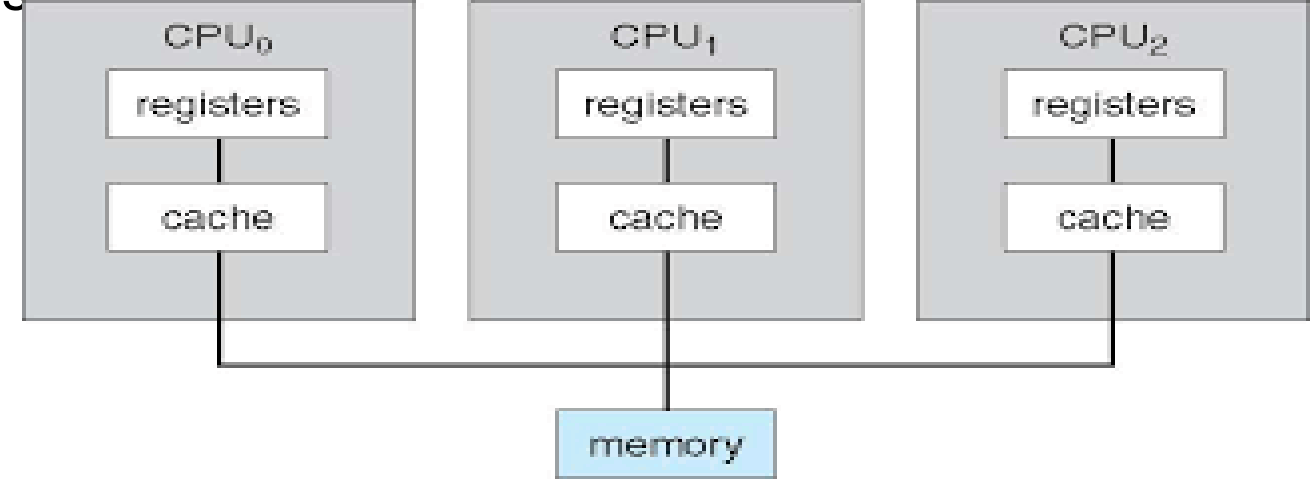
1. Symmetric Multiprocessing :

- Each processor runs an identical copy of the OS and copies communicate with one another as needed.
- Each processor performs all tasks within OS.
- All Processor are Peer and no Master Slave relationship exists.
- In Symmetric Multiprocessing treats all processors as equals and no I/O can be processed on any CPU.



Asymmetric Multiprocessing :

- Each processor is assigned a specific task.
- A master processor controls the system. The other processors look to the master for instructions or predefined task. Thus, this defines master-slave relationship.
- Master Processor schedules and allocates the work to the slave processor.
- Each processor has its own address space



Multiprocessing Operating System

❑**Advantage:**

➢Increase throughput

➢Increase Reliability

➢Cost saving

➢Battery efficient

➢True parallel processing

❑**Disadvantage:**

➢More complex

➢Overhead or coupling reduce throughput

➢Large main memory

Distributed System

- Distribute computation among several processors.
- The processor does not share memory or a clock. Instead each processor has its own local memory
- The Processor communicates with one another through various communications line such as high speed buses and telephone lines.
- These systems are usually referred as **Loosely Coupled Systems.**
- The processors may vary in size and functionality. These processors are referred as sites, nodes and computers.

Advantages :

- Resources Sharing:
- Computational Speed-up
- Reliability
- Communication

Real Time Operating System

- A real time system is defined as a data processing system in which the time interval required to process and response to I/P is so small that it controls the environment.
- RTOS responds to I/P immediately i.e., task is completed with in specific time.
- A system is said to be real time if it is required to complete its work & deliver its service on time.
- Ex. Flight control system, nuclear reaction, controlling traffic signal

Types of RTOS

❑Hard RTOS:

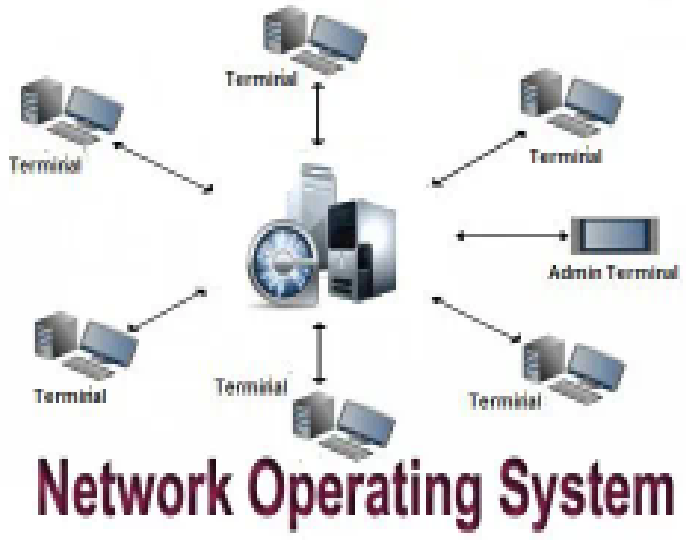
- ➢Hard RTOS is purely deterministic and time constraint system.
- ➢Ex- Missile Launching System, Satellite System, Airbag control in cars

❑Soft RTOS:

- ➢In soft real time system the meeting of deadline is not compulsory for every time for every task but process should get processed and give the result.
- ➢Ex. Personal Computer, Audio & video system

Networks Operating Systems

- Network operating system refers to software that implements an operating system of some kind that is oriented to computer networking.
- Network operating system is like as software that is installed on the server side on the network infrastructure.
- For example, one that runs on a server and enables the server to manage data, users, groups, security, applications, and other networking functions.
- The network operating system is designed to allow shared file and printer access among multiple computers in a network,
- Typically a local area network (LAN), a private network or to other networks.

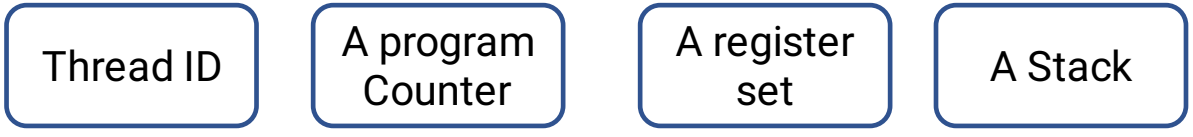


Multithreaded System (1)

❑Process: A process is a program in execution.

➢A process can further divided into sub-processes known as threads or light weight processes.

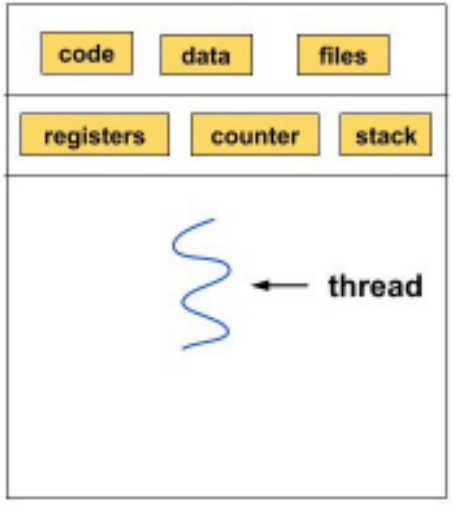
➢A thread is a basic unit of CPU utilization that reside in a process. It Comprises



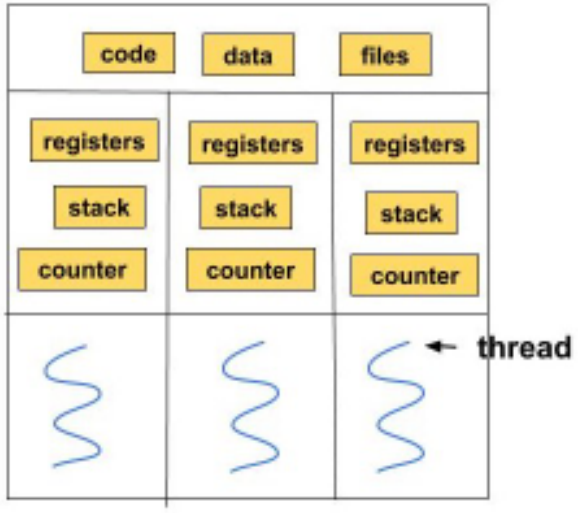
➢It shares with other threads belonging to the same process its code section, data section, and other operating system resources. Such as open files and signals.

Multithreaded System (2)

- Traditionally / heavy weight process has a single thread of control.
- If a process has multiple threads of controls, it can perform more than one task at a time.
- Threads are popularly used to improve the application through parallelism.



Single-threaded process



Multi-threaded process

Multithreaded System (2)

Types of Thread

❑User-Level Thread

- The user-level threads are managed by users and the kernel is not aware of it.
- These threads are faster to create and manage.
- The kernel manages them as if it was a single-threaded process.
- It is implemented using user-level libraries and not by system calls. So, no call to the operating system is made when a thread switches the context.
- Each process has its own private thread table to keep the track of the threads.

❑Kernel-Level Thread

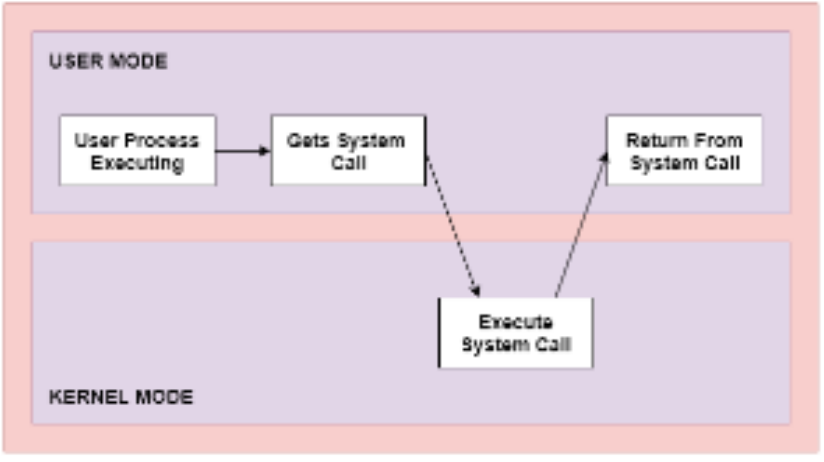
- The kernel knows about the thread and is supported by the OS.
- The threads are created and implemented using system calls.
- The thread table is not present here for each process. The kernel has a thread table to keep the track of all the threads present in the system.
- Kernel-level threads are slower to create and manage as compared to user-level threads.

Advantages of threads

- **Performance:** Threads improve the overall performance(throughput, computational speed, responsiveness) of a program.
- **Resource sharing:** As the threads can share the memory and resources of any process it allows any application to perform multiple activities inside the same address space.
- **Utilization of Multiple Processor Architecture:** The different threads can run parallel on the multiple processors hence, this enables the utilization of the processor to a large extent and efficiency.
- **Reduced Context Switching Time:** The threads minimize the context switching time as in Thread Switching, the virtual memory space remains the same.
- **Concurrency:** Thread provides concurrency within a process.

System Calls (1)

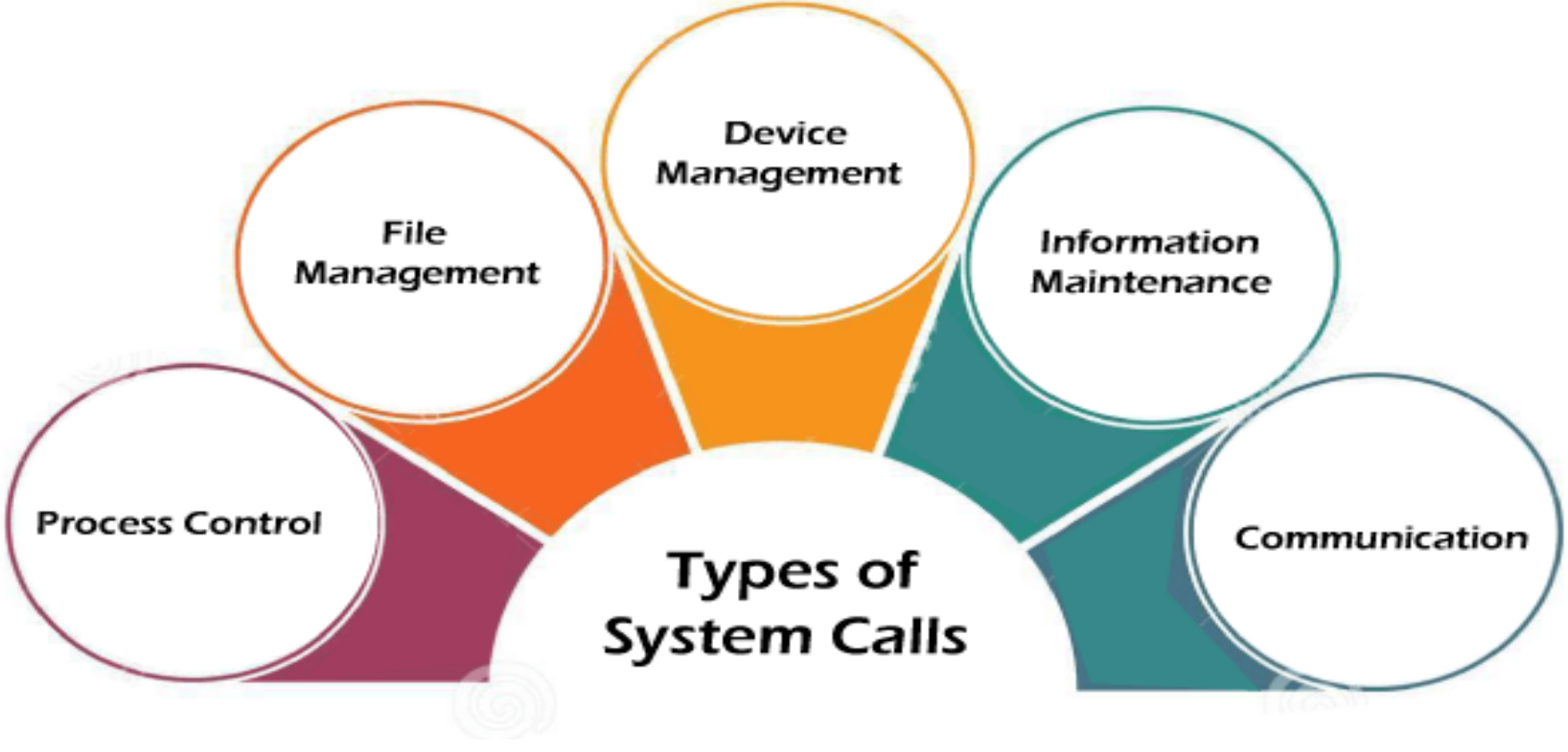
- The interface between a process and an operating system is provided by system calls.
- System calls are usually made when a process in user mode requires access to a resource.
- The key difference between User Mode and Kernel Mode is that user mode is the mode in which the applications are running and kernel mode is the privileged mode to which the computer enters when accessing hardware resources.



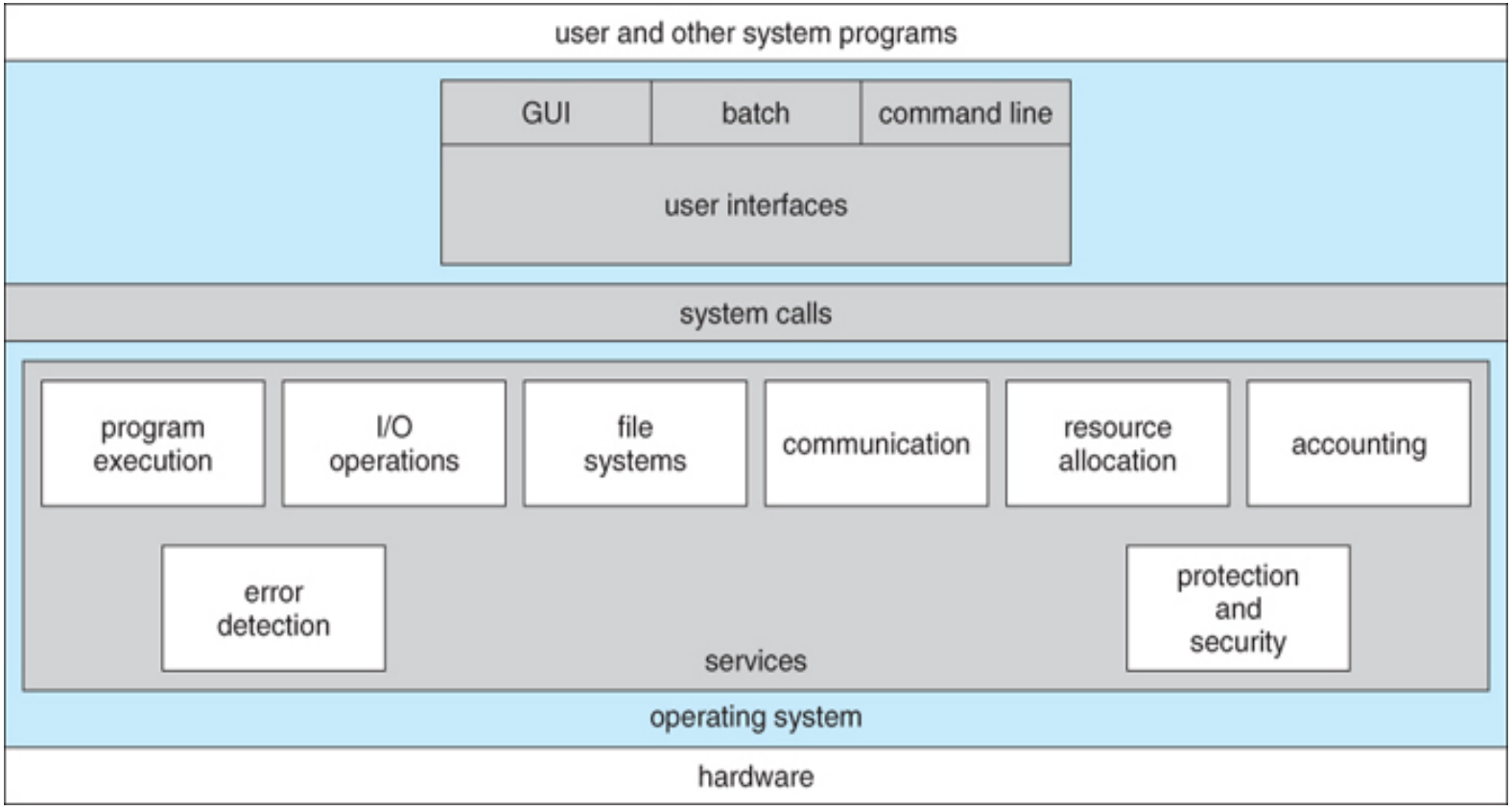
System Calls (2)

- In general, system calls are required in the following situations −
  - If a file system requires the creation or deletion of files. Reading and writing from files also require a system call.
  - Creation and management of new processes.
  - Network connections also require system calls. This includes sending and receiving packets.
  - Access to a hardware devices such as a printer, scanner etc. requires a system call.

Types of System Calls



Operating System Services



A view of Operating System Services

Objective of operating system

- Today's operating system has three major roles.
  - To hides details of hardware by creating abstraction.
  - To allocate resources for process that is to manage resources.
  - To provide a pleasant effective and efficient user interface.
    - An abstraction is a software that hides lower level details and provides a set of high level function.
    - An O.S. transform the physical words of device, instruction memory, time into virtual word that is result of abstraction build by O.S.

Process Management

- The O.S. manage many kind of activity ranging from user programs to system programs like printer-spooler, name servers, file servers etc. A process includes the complete execution context (code, data PC, registers OS resources in use etc.);
- It is the important to note that a process is only one instance of a program in execution.
- There are many process that can be running the same program

- The 5 component of an O.S. with regards to process management are
  - Creation and deletion of user and system process
  - Suspension and resumption of process
  - A mechanism for process synchronization
  - A mechanism for process communication
  - A mechanism for deadlock handling

Main memory management

- Primary memory or main memory is a large array of words or bytes. Each word or byte has its own address.
- Main memory provides storage that can be accessed directly by the CPU. That is to say for a program to be executed it must be in the main memory.
- The major activity of an OS with regards to memory management are
  - Keep track of which part of memory are currently being used and by whom
  - Decide which process are loaded into memory when memory space becomes available
  - Allocate and deallocate memory space as needed

Secondary memory management

- Generally speaking system have several levels of storage, including primary storage secondary storage & cache storage.
- Because main memory is too small to accommodate all data & programs and its data are lost when power is lost, the computer system must provide secondary storage to back up main memory.
- Secondary memory consist of tapes disks and other media designed to hold information that will eventually be accessed in primary storage.
- (primary, secondary, cache) is ordinarily divided into bytes or words consisting of a fixed no. of bytes.
- Each location in storage has an address the set of all address available to a program is called an address space.

- The three major activities of an OS with regards to secondary storage management are
  - Managing the free space available on the secondary-storage device
  - Allocation of storage space when new files have to be written
  - Scheduling the request for memory access.

Device management:

- I/O subsystem hides the peculiarities of specific hardware devices from the user.
- Only the device knows the peculiarities of the specific device to whom its assigned.

File management

- A file is the collection of related information defined by its creator. Computer can store files on the disk which provide long term storage .
- A file system is normally organized into directories to case their use, these directories may contain file and other sub directories.
- The 5 main major activities of an OS with regards to file management are
  - The creation and deletion of a file
  - The creation and deletion of a directories.
  - The support of primitives for manipulation files and directories
  - The mapping of files onto secondary storages
  - The back-up of files on stable storage media.

Command interpreter

- A command interpreter is an interface of the OS with the user.
- The user gives the commands which are executed by OS.
- The main function of a command interpreter is to get and executed the next user specified command.
- Command interpreter is not usually part of kernel, since multiple command interpreters may be supported by an operating system and they do not really need to run in kernel mode.

- There are two main advantages to separating the command interpreter from the kernel.

  - If we want to change the way the command interpreters looks i.e. I want to change the interface of command interpreter. I am able to do that if the command interpreter is separate from the kernel. I can not change the core of the kernel so I can not modify the interface.

  - If the command interpreter is a part of kernel it is possible for a malicious process to gain access to certain part of the kernel that it should not have, to avoid this it is advantageous to have the command interpreter separate from kernel.

Networking

- Distributed system is a collection of processor that do not share with memory, peripheral device, or a clock.
- The processor communicate with one another through communication lines called network.
- The communication network design must consider routing and connection strategies, and problem of protection and security.

Protection system

- If a computer systems has multiple user and allows the concurrent execution of multiple processes, then the various processes must be protected from one another’s activities.

- Protection refers to mechanism for controlling the access of programs, processes or users to the resources defined by computer system.

Operating System Services

- Following are the five services provided by an operating system for the convenience of the users.

- Program execution
- I/O operation
- File system manipulation
- Communication
- Error detection

Program execution

- The purpose of computer systems is to allow the user to execute programs.
- So the OS provide an environment where the user can conveniently run programs.
- The user dose not have to worry about the memory allocation or multitasking or anything. These things are taken care of by the operating system.
- Running a program involves all allocating and deallocating memory , CPU scheduling in case of multiprocess.
- These function can not be gives to the user-level programs. So user level programs can not help the user to run program independently without the help of O.S.

Input/output Operation

- Each program requires an input and produces output. This involves the use of I/O.

- The OS hides the details of the underlying hardware for the I/O from the user. All that user see that I/O has been performed without any details of the underlying.

- It has been actually performed for efficiency and protection user is not allowed any control over I/O. so this service can not provided by user level program.

File system manipulation

- The output of an program may need to be written into a new file. The OS has to provide this service.
- User also does not have to worry about secondary storage management. User simply gives the command for reading or writing to a file and sees his /her task accomplished.
- Thus OS makes it easier for user program to accomplished their task.
- This service involved the secondary storage management is critical to the speed of many programs and hence it is best relegated to the OS to manage it than giving individuals users the control of it.

Communication

- There are instances where processes need to communicate with each other to exchange information.
- It may be between processes running on the same computer or running on the different computer.
- By providing this service the OS relieves the user of the worry of passing messages need to be passed to processes on the other computers through a network, it can be done by user programs.
- The user program may be customized to the specifies of the hardware through which the message transits and provides the service interface to the OS

Error Detection

- An error in one part of the system may cause malfunctioning of the complete system.
- To avoid such a situation the OS constantly monitors the system for detecting the errors. This relieves the user of the worry of errors propagating to various part of the system and causing malfunctioning.
- This service can not be allowed to be handled by user programs because it involves monitoring and in cases altering area of memory or deallocation of memory for a faulty process or may be relinquishing the CPU of a process that goes into infinite loop
- These task are too critical to be handed over to the user programs if given these previleges can interfere with the correct operation of OS

Mechanism & policies

- The policies specify what will be done while the mechanism specify how it is to be done for instances the timer constructor for ensuring CPU protection is mechanism.
- On the other hand, the decision of how long the timer is to be set for a particular user in a policy decision.
- The separation of mechanism and policy is important to provide flexibility to a system. If the interface between mechanism and policy is well defined ,the change of policy may affect only a few parameters.
- Once the policy has been decided it gives the programmer the choice of using his/her own implementation. Also the underlying implementation may be changed for a more efficient one without much trouble if the mechanism and policy are well defined