Module 1

INTRODUCTION

Syllabus:

1.1. Introduction to Operating System

1.2. Operating System operations, functions, service

1.3. System calls, Types

1.4. Operating System Structure: Simple, Layered, Microkernel, Modules

1.5. System Boot Process

1.1 Introduction to Operating System

Operating System

- A program that controls the execution of application programs.
- An interface between applications and hardware.
- Controls and coordinates the usage of hardware among various applications and users.
- It is responsible for the execution of all the processes, Resource Allocation, CPU management, File Management and many other tasks.
- The purpose of an operating system is to provide an environment in which a user can execute programs in a convenient and efficient manner.
- **Main objectives of an OS:**
  - • convenience
  - • efficiency
  - • ability to evolve

- An operating system (OS) is the program that, after being initially loaded into the computer by a boot program, manages all of the other application programs in a computer.
- The application programs make use of the operating system by making requests for services through a defined application program interface (API).
- In addition, users can interact directly with the operating system through a user interface, such as a command-line interface (CLI) or a graphical UI (GUI).

1.2 Operating System operations, functions, service

- **What does an Operating system do?(Operations and Functions)**
    - **Security –**
        - The operating system uses password protection to protect user data and similar other techniques.
        - it also prevents unauthorized access to programs and user data.
    - **Control over system performance –**
        - Monitors overall system health to help improve performance.
        - Records the response time between service requests and system response to have a complete view of the system health.
        - This can help improve performance by providing important information needed to troubleshoot problems.
    - **Job accounting –**
        - Operating system Keeps track of time and resources used by various tasks and users.
        - This information can be used to track resource usage for a particular user or group of users.
    - **Error detecting aids –**
        - Operating system constantly monitors the system to detect errors and avoid the malfunctioning of the computer system.
    - **Coordination between other software and users –**
        - Operating systems also coordinate and assign interpreters, compilers, assemblers and other software to the various users of the computer systems.
    - **Memory Management –**
        - The operating system manages the Primary Memory or Main Memory.
        - It keeps track of primary memory, i.e., which bytes of memory are used by which user program.

    - In multiprogramming, the OS decides the order in which processes are granted access to memory, and for how long.
  - **Processor Management –**
    - In a multiprogramming environment, the OS decides the order in which processes have access to the processor, and how much processing time each process has.
    - This function of the OS is called process scheduling.
  - **Device Management –**
    - An OS manages device communication via their respective drivers.
    - Keeps track of all devices connected to the system. designates a program responsible for every device known as the Input/Output controller.
    - Decides which process gets access to a certain device and for how long. Allocates devices in an effective and efficient way.
  - **File Management –**
    - A file system is organized into directories for efficient or easy navigation and usage.
    - OS keeps track of where information is stored, user access settings and status of every file and more…
    - These facilities are collectively known as the file system.

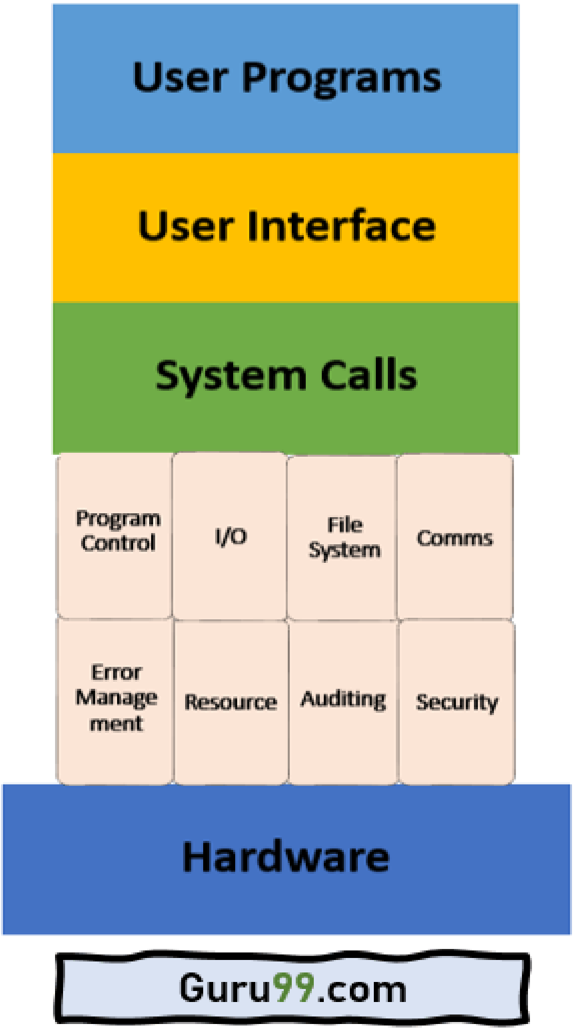
- **Operating System Services**

  - **Program Execution:**
    - The Operating System is responsible for execution of all types of programs whether it be user programs or system programs.
    - The Operating System utilises various resources available for the efficient running of all types of functionalities.
  - **Handling Input/Output Operations:**
    - The Operating System is responsible for handling all sorts of inputs, i.e, from keyboard, mouse, desktop, etc.

        - The Operating System does all interfacing in the most appropriate manner regarding all kinds of Inputs and Outputs.
    - **Manipulation of File System:**
        - The Operating System is responsible for making decisions regarding the storage of all types of data or files, i.e, floppy disk/hard disk/pen drive, etc.
        - The Operating System decides how the data should be manipulated and stored.
    - **Error Detection and Handling:**
        - The Operating System is responsible for detection of any types of error or bugs that can occur while any task.
        - The well secured OS sometimes also acts as countermeasure for preventing any sort of breach to the Computer System from any external source and probably handling them.
    - **Resource Allocation:**
        - The Operating System ensures the proper use of all the resources available by deciding which resource to be used by whom for how much time.
        - All the decisions are taken by the Operating System.
    - **Accounting:**
        - The Operating System tracks an account of all the functionalities taking place in the computer system at a time.
        - All the details such as the types of errors occurred are recorded by the Operating System.
    - **Information and Resource Protection:**
        - The Operating System is responsible for using all the information and resources available on the machine in the most protected way.
        - The Operating System must foil an attempt from any external resource to hamper any sort of data or information.

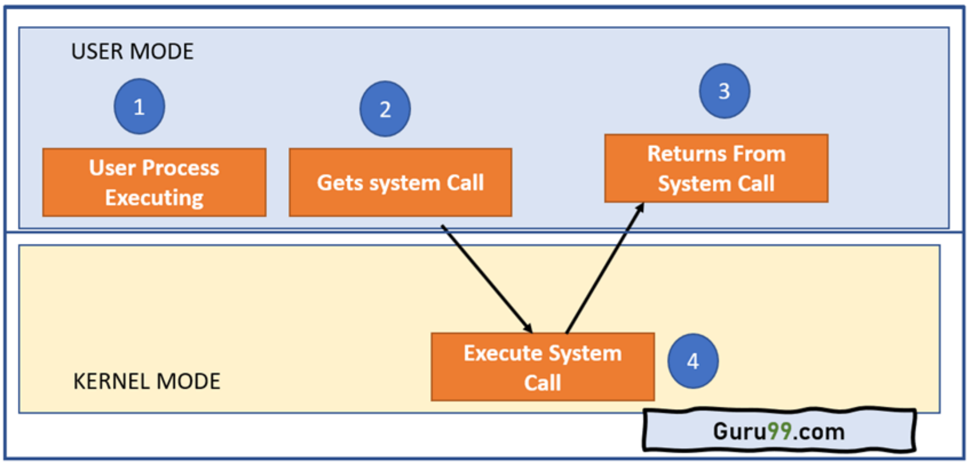
1.3 System Calls and Types

- **What is System Call in Operating Systems?**
  - A system call is a mechanism that provides the interface between a process and the operating system.
  - It is a programmatic method in which a computer program requests a service from the kernel of the OS.
  - System call offers the services of the operating system to the user programs via API (Application Programming Interface).
  - System calls are the only entry points for the kernel system.



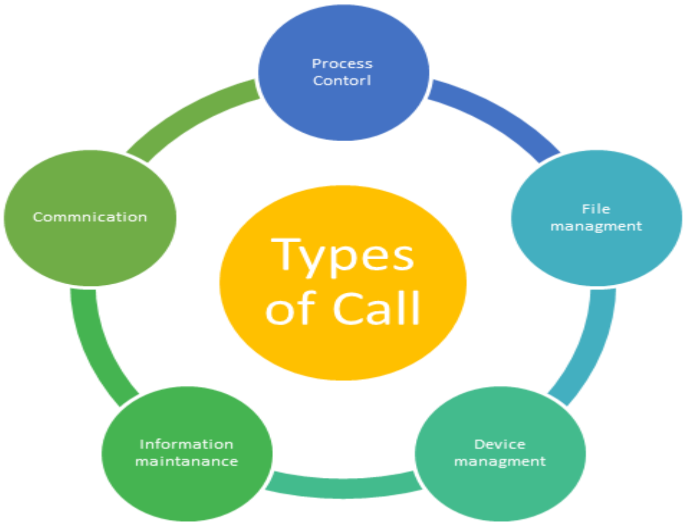
- **How Does System Call Work?**
  - **Step 1)** The processes executed in the user mode till the time a system call interrupts it.
  - **Step 2)** After that, the system call is executed in the kernel-mode on a priority basis.

- **Step 3)** Once system call execution is over, control returns to the user mode.,
- **Step 4)** The execution of user processes resumed in Kernel mode.



- **Why do you need System Calls in OS?**
  - Reading and writing from files demand system calls.
  - If a file system wants to create or delete files, system calls are required.
  - System calls are used for the creation and management of new processes.
  - Network connections need system calls for sending and receiving packets.
  - Access to hardware devices like scanner, printer, need a system call.

- **Types of System calls**



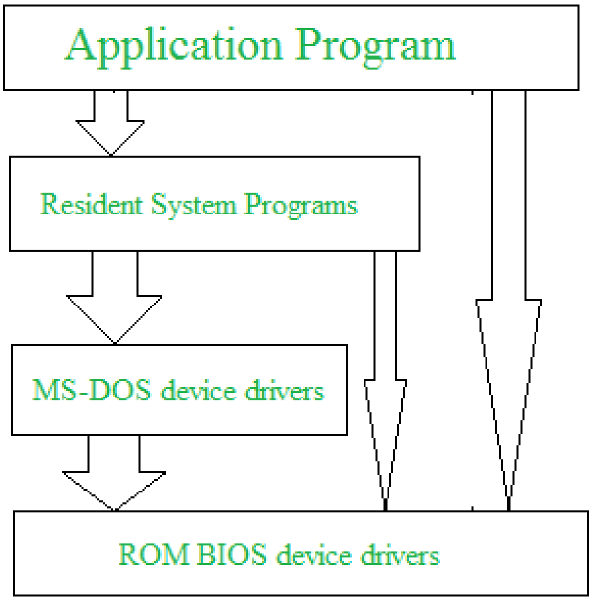
- Process Control
    - This system calls perform the task of process creation, process termination, etc.
    - Functions:
        - End and Abort
        - Load and Execute
        - Create Process and Terminate Process
        - Wait and Signed Event
        - Allocate and free memory
- File Management
    - File management system calls handle file manipulation jobs like creating a file, reading, and writing, etc.
    - Functions:
        - Create a file
        - Delete file
        - Open and close file
        - Read, write, and reposition
        - Get and set file attributes
- Device Management
    - Device management does the job of device manipulation like reading from device buffers, writing into device buffers, etc.
    - Functions
        - Request and release device
        - Logically attach/ detach devices
        - Get and Set device attributes
- Information Maintenance
    - It handles information and its transfer between the OS and the user program.
    - Functions:
        - Get or set time and date
        - Get process and device attributes
- Communications
    - These types of system calls are specially used for interprocess communications.
    - Functions:
        - Create, delete communications connections
        - Send, receive message
        - Help OS to transfer status information
        - Attach or detach remote devices

- **Important System Calls Used in OS**

Categories	Windows	Unix
Process control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
Device manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
File manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	Open() Read() write() close!)
Information maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()
Communication	CreatePipe() CreateFileMapping() MapViewOfFile()	Pipe() shm_open() mmap()
Protection	SetFileSecurity() InitializeSecurityDescriptor( ) SetSecurityDescriptor Group ()	Chmod() Umask() Chown()
Process control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()
Device manipulation	SetConsoleMode() ReadConsole() WriteConsole()	ioctl() read() write()
File manipulation	CreateFile() ReadFile() WriteFile() CloseHandle()	Open() Read() write() close!)
Information maintenance	GetCurrentProcessID() SetTimer() Sleep()	getpid() alarm() sleep()

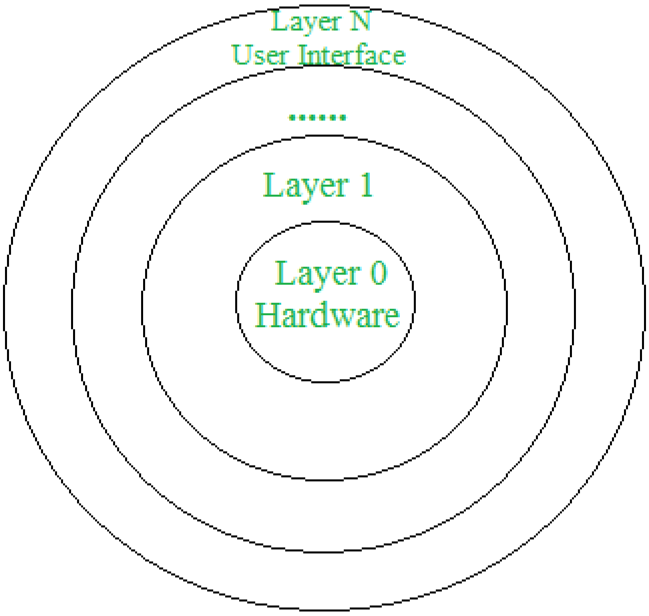
1.4 Operating System Structure: Simple, Layered,Microkernel, Modules

- **Simple structure:**
  - Such operating systems do not have well defined structure and are small, simple and limited systems.
  - The interfaces and levels of functionality are not well separated.
  - MS-DOS is an example of such an operating system.
  - In MS-DOS application programs are able to access the basic I/O routines.
  - These types of operating systems cause the entire system to crash if one of the user programs fails.
  - Diagram of the structure of MS-DOS is shown below.



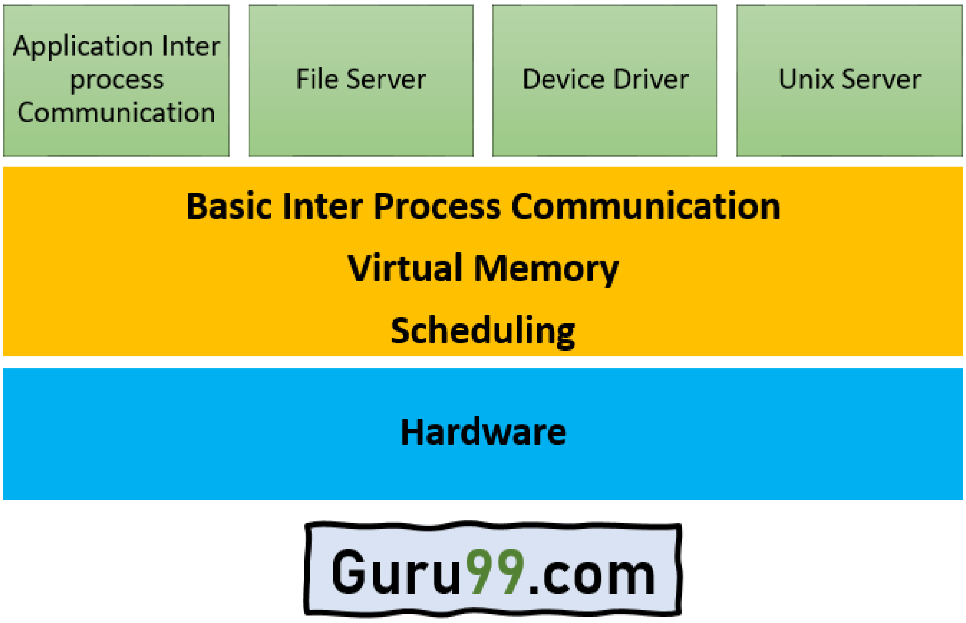
  -
  - **Advantages of Simple structure:**
    - It delivers better application performance because of the few interfaces between the application program and the hardware.
    - Easy for kernel developers to develop such an operating system.
  - **Disadvantages of Simple structure:**
    - The structure is very complicated as no clear boundaries exist between modules.
    - It does not enforce data hiding in the operating system.

- **Layered structure:**
    - An OS can be broken into pieces and retain much more control on the system.
    - In this structure the OS is broken into a 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.
    - 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.
    -



    - **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.

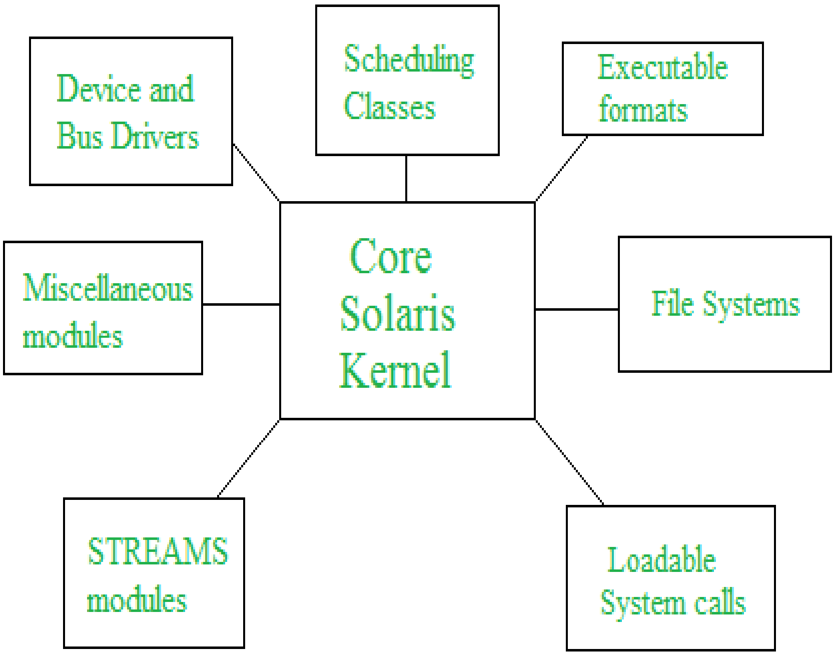
- **Micro-kernel:**
  - This structure designs the operating system by removing all non-essential components from the kernel and implementing them as system and user programs.
  - This results in a smaller kernel called the micro-kernel.
  - Advantages of this structure are that all new services need to be added to user space and does not require the kernel to be modified.
  - Thus it is more secure and reliable as if a service fails then the rest of the operating system remains untouched.
  - Mac OS is an example of this type of OS.
  - Microkernel architecture is small and isolated therefore it can function better.
  - Providing services in a microkernel system are expensive compared to the normal monolithic system
  -



  - **Advantages of Microkernel structure:**
    - It makes the operating system portable to various platforms.
    - As microkernels are small so these can be tested effectively.

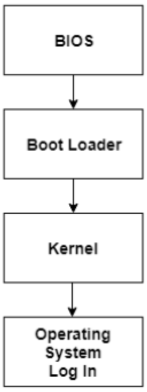
  - **Disadvantages of Microkernel structure:**
    - Increased level of inter module communication degrades system performance.

- **Modular structure or approach:**
  - It is considered as the best approach for an OS.
  - It involves designing a modular kernel.
  - The kernel has only a set of core components and other services are added as dynamically loadable modules to the kernel either during run time or boot time.
  - It resembles layered structure due to the fact that each kernel has defined and protected interfaces but it is more flexible than the layered structure as a module can call any other module.
  - For example Solaris OS is organized as shown in the figure.



  -
  - **Advantages of Modular structure**
    - advantages of layered structure but with more flexible
    - advantages of microkernel approach, without message passing overhead
  - **Disadvantages of Modular structure**
    - not as clean a design as the layered approach
    - not as small a kernel as a microkernel
    - but, achieves best of both worlds as far as possible

1.5 System Boot Process

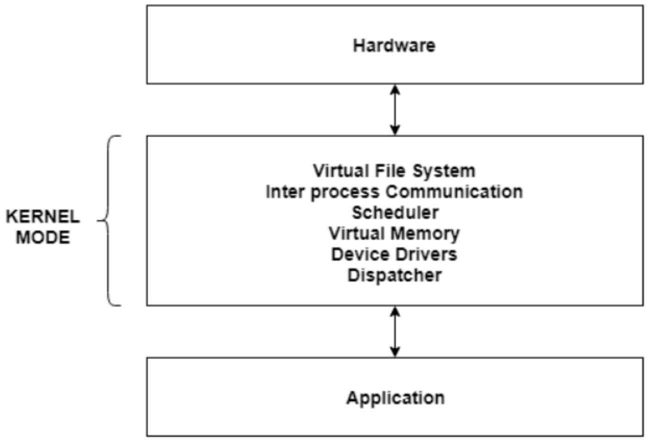


- The CPU initializes itself after the power in the computer is first turned on.
- This is done by triggering a series of clock ticks that are generated by the system clock.
- After this, the CPU looks for the system’s ROM BIOS to obtain the first instruction in the start-up program.
- This first instruction is stored in the ROM BIOS and it instructs the system to run POST (Power On Self Test) in a memory address that is predetermined.
- POST first checks the BIOS chip and then the CMOS RAM.
- If there is no battery failure detected by POST, then it continues to initialize the CPU.
- POST also checks the hardware devices, secondary storage devices such as hard drives, ports etc. And other hardware devices such as the mouse and keyboard.
- This is done to make sure they are working properly.
- After POST makes sure that all the components are working properly, then the BIOS finds an operating system to load.
- In most computer system’s, the operating system loads from the C drive onto the hard drive.
- The CMOS chip typically tells the BIOS where the operating system is found.
- The order of the different drives that CMOS looks at while finding the operating system is known as the boot sequence.
- This sequence can be changed by changing the CMOS setup.
- After finding the appropriate boot drive, the BIOS first finds the boot record which tells it to find the beginning of the operating system.
- After the initialization of the operating system, the BIOS copies the files into the memory. Then the operating system controls the boot process.
- In the end, the operating system does a final inventory of the system memory and loads the device drivers needed to control the peripheral devices.
- The users can access the system applications to perform various tasks.

Questions

1. How does hardware find the Operating System kernel after system switch-on*?* **(3)(Page no 13)**
2. What is the purpose of system calls in an operating system?**(3)(Page no 6)**
3. Explain the following structures of operating system**(12)**
    a. Monolithic systems **(3)(Page no 15,16)**
    b. Layered Systems **(3)(Page no 10)**
    c. Micro Kernel **(3)(page no 11)**
    d. Modular approach. **(3)(Page no 12)**
4. Under what circumstances would a user be better off using a time sharing system than a PC or a single user workstation? **(2)(HomeWork Solutions Qn 2 Page 17)**
5. a) What is the main advantage of the microkernel approach to system design? How do user programs and system programs interact in a microkernel architecture?**(8)(page no 11)**
6. Describe the differences between symmetric and asymmetric multiprocessing? What are the advantages and disadvantages of multiprocessor systems*?* ***(6)(Page no 22)***

- **Monolithic System**
    - The entire operating system works in the kernel space in the monolithic system.
    - This increases the size of the kernel as well as the operating system.
    - This is different from the microkernel system where the minimum software that is required to correctly implement an operating system is kept in the kernel.
    - The kernel provides various services such as memory management, file management, process scheduling etc. using function calls.
    - This makes the execution of the operating system quite fast as the services are implemented under the same address space.



Monolithic Kernel Operating System

    -
    - **Differences Between Microkernel and Monolithic Kernel**
        - The microkernel is much smaller in size as compared to the monolithic kernel.
        - The microkernel is easily extensible whereas this is quite complicated for the monolithic kernel.
        - The execution of the microkernel is slower as compared to the monolithic kernel.
        - Much more code is required to write a microkernel than the monolithic kernel.
        - Examples of Microkernel are QNX, Symbian, L4 Linux etc. Monolithic Kernel examples areLinux, BSD etc.
    - **Advantages of Monolithic Kernel**
        - The execution of the monolithic kernel is quite fast as the services such as memory management, file management, process scheduling etc.are implemented under the same address space.
        - A process runs completely in a single address space in the monolithic

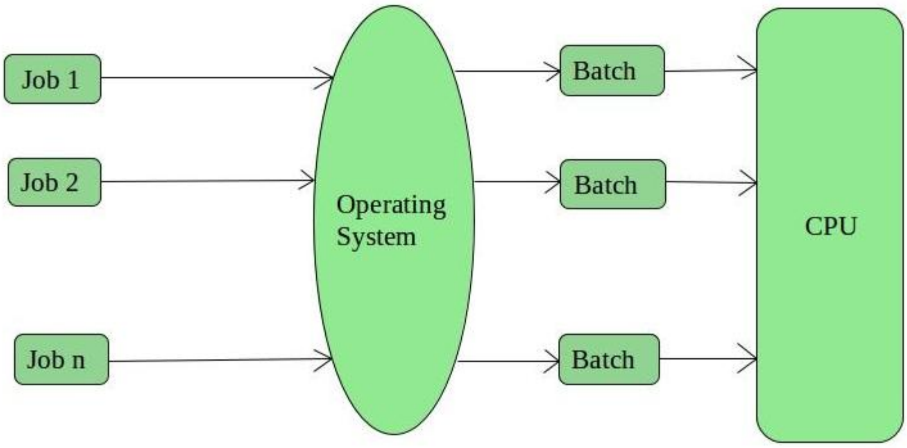
kernel.
- The monolithic kernel is a static single binary file.

- **Disadvantages of Monolithic Kernel**
  - If any service fails in the monolithic kernel, it leads to the failure of the entire system.
  - To add any new service, the entire operating system needs to be modified by the user.

Types of Operating Systems:

1. Batch Operating System –

- This type of operating system does not interact with the computer directly.
- There is an operator which takes similar jobs having the same requirement and groups them into batches.
- It is the responsibility of the operator to sort jobs with similar needs.
- Examples of Batch based Operating System: Payroll System, Bank Statements, etc.

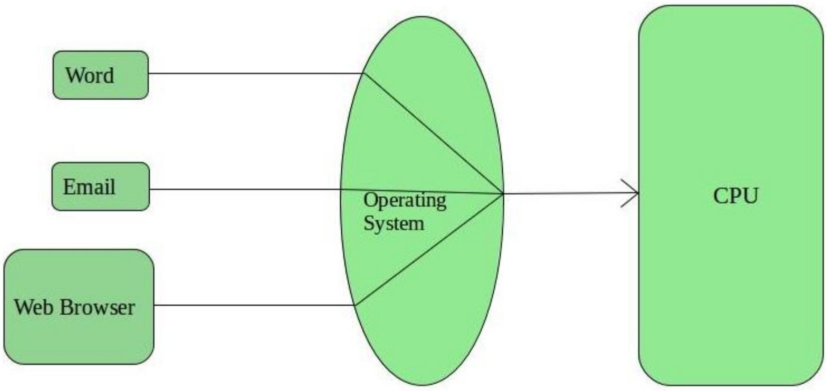


-
- **Advantages of Batch Operating System:**
  - It is very difficult to guess or know the time required for any job to complete. Processors of the batch systems know how long the job would be when it is in queue
  - Multiple users can share the batch systems
  - The idle time for the batch system is very less
  - It is easy to manage large work repeatedly in batch systems
- **Disadvantages of Batch Operating System:**
  - The computer operators should be well known with batch systems

  - Batch systems are hard to debug
  - It is sometimes costly
  - The other jobs will have to wait for an unknown time if any job fails

2. Time-Sharing Operating Systems –

- Each task is given some time to execute so that all the tasks work smoothly.
- Each user gets the time of CPU as they use a single system.
- These systems are also known as Multitasking Systems.
- The task can be from a single user or different users also.
- The time that each task gets to execute is called quantum. After this time interval is over OS switches over to the next task.
- Examples of Time-Sharing OSs are: Multics, Unix, etc.

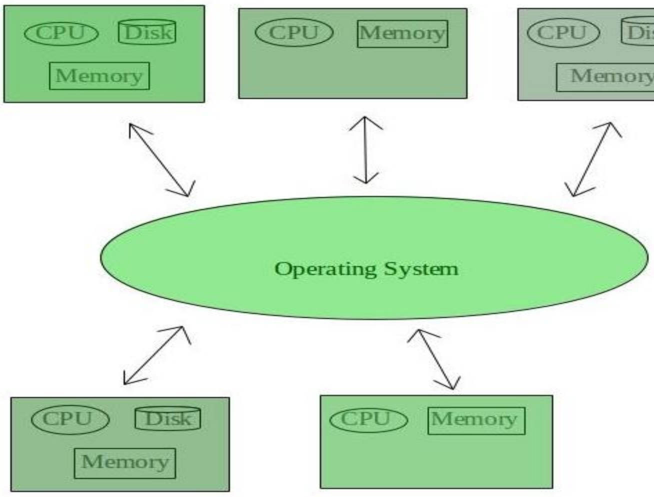


-
- **Advantages of Time-Sharing OS:**
  - Each task gets an equal opportunity
  - Fewer chances of duplication of software
  - CPU idle time can be reduced
- **Disadvantages of Time-Sharing OS:**
  - Reliability problem
  - One must have to take care of the security and integrity of user programs and data
  - Data communication problem

3. Distributed Operating System –

- These types of operating systems are a recent advancement in the world of computer technology and are being widely accepted all over the world and, that too, with a great pace.

- Various autonomous interconnected computers communicate with each other using a shared communication network.
- Independent systems possess their own memory unit and CPU.
- These are referred to as loosely coupled systems or distributed systems.
- These system's processors differ in size and function.
- The major benefit of working with these types of the operating system is that it is always possible that one user can access the files or software which are not actually present on his system but some other system connected within this network i.e., remote access is enabled within the devices connected in that network.
- Examples of Distributed Operating Systems are- LOCUS, etc.

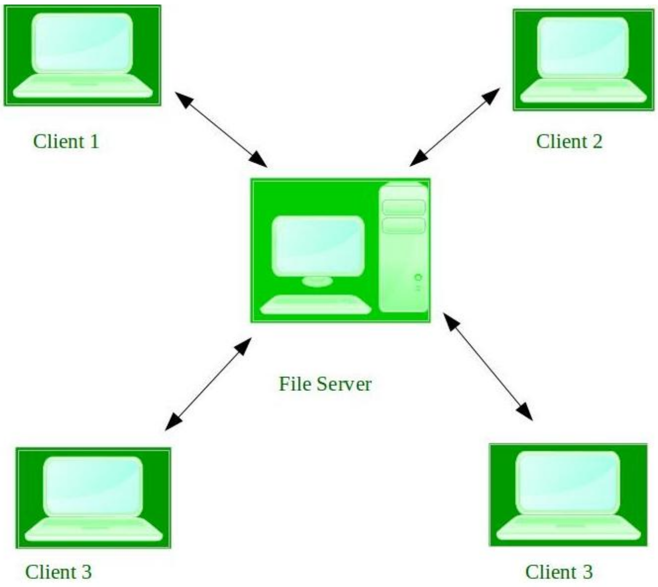


-
- **Advantages of Distributed Operating System:**
  - Failure of one will not affect the other network communication, as all systems are independent from each other
  - Electronic mail increases the data exchange speed
  - Since resources are being shared, computation is highly fast and durable
  - Load on host computer reduces
  - These systems are easily scalable as many systems can be easily added to the network
  - Delay in data processing reduces

- **Disadvantages of Distributed Operating System:**
  - Failure of the main network will stop the entire communication
  - To establish distributed systems the language which is used are not well defined yet
  - These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet

4. Network Operating System –

- These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions.
- These types of operating systems allow shared access of files, printers, security, applications, and other networking functions over a small private network.
- One more important aspect of Network Operating Systems is that all the users are well aware of the underlying configuration, of all other users within the network, their individual connections, etc. and that's why these computers are popularly known as tightly coupled systems.
- Examples of Network Operating System are: Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD, etc.

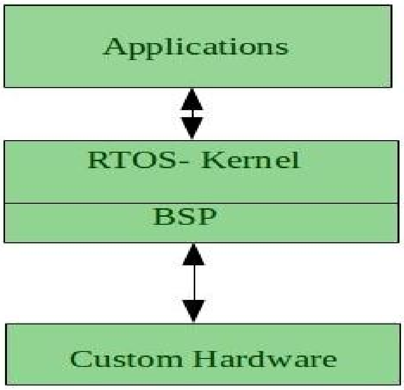


-
- Advantages of Network Operating System:
  - Highly stable centralized servers
  - Security concerns are handled through servers

    - New technologies and hardware up-gradation are easily integrated into the system
    - Server access is possible remotely from different locations and types of systems
- **Disadvantages of Network Operating System:**
    - Servers are costly
    - User has to depend on a central location for most operations
    - Maintenance and updates are required regularly

5. Real-Time Operating System –

- These types of OSs serve real-time systems.
- The time interval required to process and respond to inputs is very small.
- This time interval is called response time.
- Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.
- Two types of Real-Time Operating System which are as follows:
    - Hard Real-Time Systems:
        - These OSs are meant for applications where time constraints are very strict and even the shortest possible delay is not acceptable.
        - These systems are built for saving life like automatic parachutes or airbags which are required to be readily available in case of any accident.
        - Virtual memory is rarely found in these systems.
    - Soft Real-Time Systems:
        - These OSs are for applications where time-constraint is less strict



-
- Examples of Real-Time Operating Systems are: Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.
- Advantages of RTOS:
    - Maximum Consumption: Maximum utilization of devices and system, thus more output from all the resources
    - Task Shifting: The time assigned for shifting tasks in these systems are very less. For example, in older systems, it takes about 10 microseconds in shifting one task to another, and in the latest systems, it takes 3 microseconds.
    - Focus on Application: Focus on running applications and less importance to applications which are in the queue.
    - Real-time operating system in the embedded system: Since the size of programs are small, RTOS can also be used in embedded systems like in transport and others.
    - Error Free: These types of systems are error-free.
    - Memory Allocation: Memory allocation is best managed in these types of systems.
- Disadvantages of RTOS:
    - Limited Tasks: Very few tasks run at the same time and their concentration is very less on few applications to avoid errors.
    - Use heavy system resources: Sometimes the system resources are not so good and they are expensive as well.

  - Complex Algorithms: The algorithms are very complex and difficult for the designer to write on.
  - Device driver and interrupt signals: It needs specific device drivers and interrupts signals to respond earliest to interrupts.
  - Thread Priority: It is not good to set thread priority as these systems are very less prone to switching tasks.

□ **MultiProcessing System**

Asymmetric Multiprocessing	Symmetric Multiprocessing
In asymmetric multiprocessing, the processors are not treated equally.	In symmetric multiprocessing, all the processors are treated equally.
Tasks of the operating system are done by the master processor.	Tasks of the operating system are done individual processor
No Communication between Processors as they are controlled by the master processor.	All processors communicate with another processor by a shared memory.
In asymmetric multiprocessing, processes are master-slave.	In symmetric multiprocessing, the process is taken from the ready queue.
Asymmetric multiprocessing systems are cheaper.	Symmetric multiprocessing systems are costlier.
Asymmetric multiprocessing systems are easier to design	Symmetric multiprocessing systems are complex to design

- **Advantages of Multiprocessor Systems**
  - **More reliable Systems**
    - In a multiprocessor system, even if one processor fails, the system will not halt.
    - This ability to continue working despite hardware failure is known as graceful degradation.
    - For example: If there are 5 processors in a multiprocessor system and one of them fails, then also 4 processors are still working.
    - So the system only becomes slower and does not ground to a halt.

  - **Enhanced Throughput**
    - If multiple processors are working in tandem, then the throughput of the system increases i.e. number of processes getting executed per unit

of time increase.

- If there are N processors then the throughput increases by an amount just under N.

- **More Economic Systems**
    - Multiprocessor systems are cheaper than single processor systems in the long run because they share the data storage, peripheral devices, power supplies etc.
    - If there are multiple processes that share data, it is better to schedule them on multiprocessor systems with shared data than have different computer systems with multiple copies of the data.

- **Disadvantages of Multiprocessor Systems**
    - **Increased Expense**
        - Even though multiprocessor systems are cheaper in the long run than using multiple computer systems, still they are quite expensive.
        - It is much cheaper to buy a simple single processor system than a multiprocessor system.
    - **Complicated Operating System Required**
        - There are multiple processors in a multiprocessor system that share peripherals, memory etc.
        - So, it is much more complicated to schedule processes and impart resources to processes.than in single processor systems. Hence, a more complex and complicated operating system is required in multiprocessor systems.
    - **Large Main Memory Required**
        - All the processors in the multiprocessor system share the memory. So a much larger pool of memory is required as compared to single processor systems.