Multiprogramming Operating System

In this article, you will learn about the multiprogramming operating system, its working, advantages, and disadvantages.

What is the Multiprogramming Operating System?

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

When a program is being performed, it is known as a **"Task", "Process"**, and **"Job"**. Concurrent program executions improve system resource consumption and throughput as compared to serial and batch processing systems.

The primary goal of multiprogramming is to manage the entire system's resources. The key components of a multiprogramming system are the file system, command processor, transient area, and I/O control system. As a result, multiprogramming operating systems are designed to store different programs based on sub-segmenting parts of the transient area. The resource management routines are linked with the operating system core functions.

Types of the Multiprogramming Operating System

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

1. **Multitasking Operating System**
2. **Multiuser Operating System**

Multitasking Operating System

A multitasking operating system enables the execution of two or more programs at the same time. The operating system accomplishes this by shifting each program into and out of memory one at a time. When a program is switched out of memory, it is temporarily saved on disk until it is required again.

Multiuser Operating System

A multiuser operating system allows many users to share processing time on a powerful central computer from different terminals. The operating system accomplishes this by rapidly switching between terminals, each of which receives a limited amount of processor time on the central computer. The operating system changes among terminals so quickly that each user seems to have continuous access to the central computer. If there are many users on a system like this, the time it takes the central computer to reply can become more obvious.

Working of the Multiprogramming Operating System

Multiple users can accomplish their jobs simultaneously in the multiprogramming system, and it can be stored in the main memory. When one program is engaged in I/O operations, the CPU may deliver time to various programs while sitting in idle mode.

When one application is waiting for an I/O transfer, another is ready to use the processor at all times, and numerous programs may share CPU time. All jobs are not run simultaneously, but there could be numerous jobs running on the processor at the same time, and parts of other processes being executed first, then another segment, etc. As a result, the overall goal of a multiprogramming system is to keep the CPU busy until some tasks are available in the job pool. Thus, the numerous programs can run on a single processor computer, and the CPU is never idle.

Examples of Multiprogramming Operating System

There are various examples of multiprogramming operating systems, including download apps, transfer data, MS-Excel, Google Chrome, Firefox browser, and many more apps. Other examples are Windows O/S, UNIX O/S, Microcomputers such as XENIX, MP/M, and ESQview.

Advantages and Disadvantages of Multiprogramming Operating System

There are various advantages and disadvantages of the multiprogramming operating system. Some of the advantages and disadvantages are as follows:

Advantages

There are various advantages of the multiprogramming operating system. Some of the advantages are as follows:

1. It provides less response time.
2. It may help to run various jobs in a single application simultaneously.
3. It helps to optimize the total job throughput of the computer.
4. Various users may use the multiprogramming system at once.
5. Short-time jobs are done quickly in comparison to long-time jobs.
6. It may help to improve turnaround time for short-time tasks.
7. It helps in improving CPU utilization and never gets idle.
8. The resources are utilized smartly.

Disadvantages

There are various disadvantages of the multiprogramming operating system. Some of the disadvantages are as follows:

1. It is highly complicated and sophisticated.
2. The CPU scheduling is required.
3. Memory management is needed in the operating system because all types of tasks are stored in the main memory.
4. The harder task is to handle all processes and tasks.
5. If it has a large number of jobs, then long-term jobs will require a long wait.

Multiprocessing Operating system

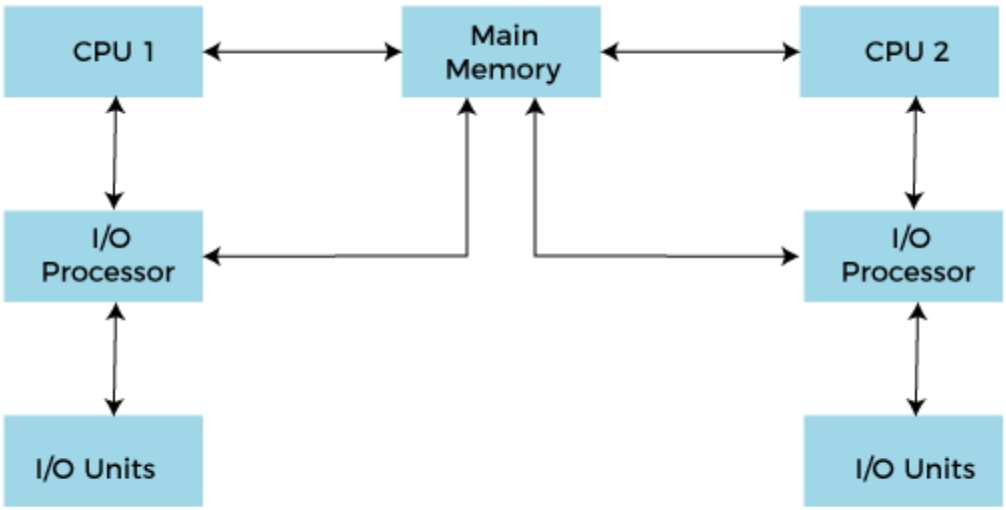
In operating systems, to improve the performance of more than one CPU can be used within one computer system called Multiprocessor operating system.

Multiple CPUs are interconnected so that a job can be divided among them for faster execution. When a job finishes, results from all CPUs are collected and compiled to give the final output. Jobs needed to share main memory and they may also share other system

resources among themselves. Multiple CPUs can also be used to run multiple jobs simultaneously.

For Example: UNIX Operating system is one of the most widely used multiprocessing systems.

The basic organization of a typical multiprocessing system is shown in the given figure.



Working of Multiprocessor System

To employ a multiprocessing operating system effectively, the computer system must have the following things:

- A motherboard is capable of handling multiple processors in a multiprocessing operating system.
- Processors are also capable of being used in a multiprocessing system.

Advantages of multiprocessing operating system are:

- **Increased reliability:** Due to the multiprocessing system, processing tasks can be distributed among several processors. This increases reliability as if one processor fails; the task can be given to another processor for completion.

- **Increased throughout:** As several processors increase, more work can be done in less
- **The economy of Scale:** As multiprocessors systems share peripherals, secondary storage devices, and power supplies, they are relatively cheaper than single-processor systems.

Disadvantages of Multiprocessing operating System

- Operating system of multiprocessing is more complex and sophisticated as it takes care of multiple CPUs at the same time.

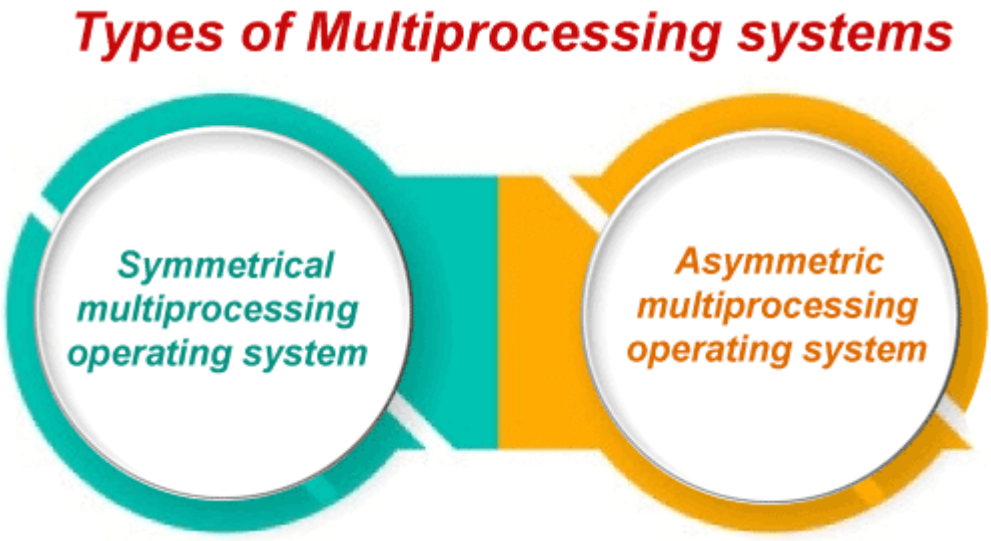
Types of multiprocessing systems

- Symmetrical multiprocessing operating system
- Asymmetric multiprocessing operating system

Symmetrical multiprocessing operating system:

In a Symmetrical multiprocessing system, each processor executes the same copy of the operating system, takes its own decisions, and cooperates with other processes to smooth the entire functioning of the system. The CPU scheduling policies are very simple. Any new job submitted by a user can be assigned to any processor that is least burdened. It also results in a system in which all processors are equally burdened at any time.

The symmetric multiprocessing operating system is also known as a "shared every-thing" system, because the processors share memory and the Input output bus or data path. In this system processors do not usually exceed more than 16.



Characteristics of Symmetrical multiprocessing operating system:

- In this system, any processor can run any job or process.
- In this, any processor initiates an Input and Output operation.

Advantages of Symmetrical multiprocessing operating system:

- These systems are fault-tolerant. Failure of a few processors does not bring the entire system to a halt.

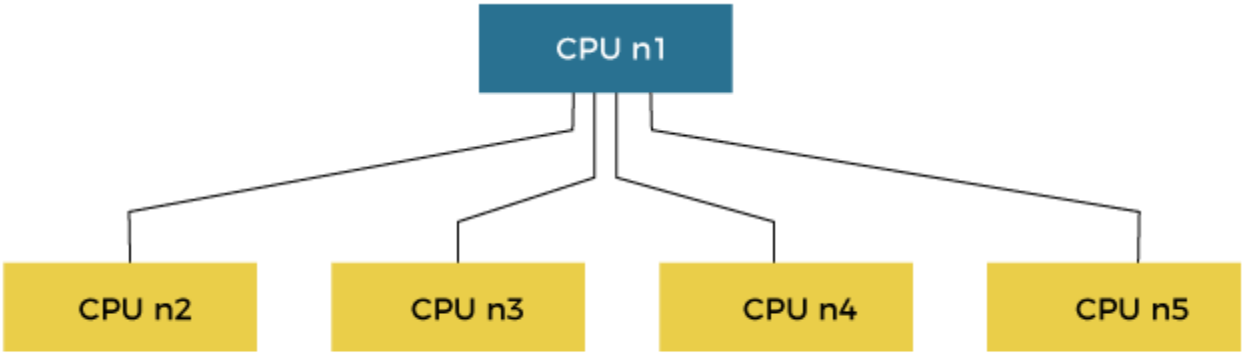
Disadvantages of Symmetrical multiprocessing operating system:

- It is very difficult to balance the workload among processors rationally.
- Specialized synchronization schemes are necessary for managing multiple processors.

Asymmetric multiprocessing operating system

In an asymmetric multiprocessing system, there is a master slave relationship between the processors.

Further, one processor may act as a master processor or supervisor processor while others are treated as shown below.



Asymmetric Multiprocessor System

In the above figure, the asymmetric processing system shows that CPU n1 acts as a supervisor whose function controls other following processors.

In this type of system, each processor is assigned a specific task, and there is a designated master processor that controls the activities of other processors.

For example, we have a math co-processor that can handle mathematical jobs better than the main CPU. Similarly, we have an MMX processor that is built to handle multimedia-related jobs. Similarly, we have a graphics processor to handle the graphics-related job better than the main processor. When a user submits a new job, the OS has to decide which processor can perform it better, and then that processor is assigned that newly arrived job. This processor acts as the master and controls the system. All other processors look for masters for instructions or have predefined tasks. It is the responsibility of the master to allocate work to other processors.

Advantages of Asymmetric multiprocessing operating system:

- In this type of system execution of Input and Output operation or an application program may be faster in some situations because many processors may be available for a single job.

Disadvantages of Asymmetric multiprocessing operating system:

- In this type of multiprocessing operating system the processors are unequally burdened. One processor may be having a long job queue, while another one may be sitting idle.

- In this system, if the process handling a specific work fails, the entire system will go down.

Head to head comparison between multiprogramming and multitasking

- Here, there are various head-to-head comparison between multiprogramming and multitasking. Some of them are as follows:

Features	Multiprogramming
Basic	It allows multiple programs to utilize the CPU simultaneously.
Mechanism	Based on the context switching mechanism.
Objective	It is useful for reducing/decreasing CPU idle time and increasing throughput as much possible.
Execution	When one job or process completes its execution or switches to an I/O task in a mu programmed system, the system momentarily suspends that process. It selects anoth process from the process scheduling pool (waiting queue) to run.
CPU Switching	In a multiuser environment, the CPU switches between programs/processes quickly.
Timing	It takes maximum time to execute the process.

Time-Sharing Operating system

Operating SystemWindowsMCA

An **operating system (OS)** is basically a collection of software that manages computer hardware resources and provides common services for computer programs. Operating system is a crucial component of the system software in a computer system.

Time-Sharing Operating Systems is one of the important **type of operating system**.

Time-sharing enables many people, located at various terminals, to use a particular computer system at the same time. Multitasking or Time-Sharing Systems is a logical extension of multiprogramming. Processor's time is shared among multiple users simultaneously is termed as time-sharing.

The main difference between Time-Sharing Systems and Multiprogrammed Batch Systems is that in case of **Multiprogrammed batch systems**, the objective is to maximize processor use, whereas in Time-Sharing Systems, the objective is to minimize response time.

Multiple jobs are implemented by the **CPU** by switching between them, but the switches occur so frequently. So, the user can receive an immediate response. For an example, in a transaction processing, the processor executes each user program in a short burst or quantum of computation, i.e.; if n users are present, then each user can get a time quantum. Whenever the user submits the command, the response time is in few seconds at most.

An operating system uses CPU scheduling and multiprogramming to provide each user with a small portion of a time. Computer systems which were designed primarily as batch systems have been modified to time-sharing systems.

Advantages of Timesharing operating systems are –

- It provides the advantage of quick response.
- This type of operating system avoids duplication of software.
- It reduces CPU idle time.

Disadvantages of Time-sharing operating systems are –

- Time sharing has problem of reliability.
- Question of security and integrity of user programs and data can be raised.

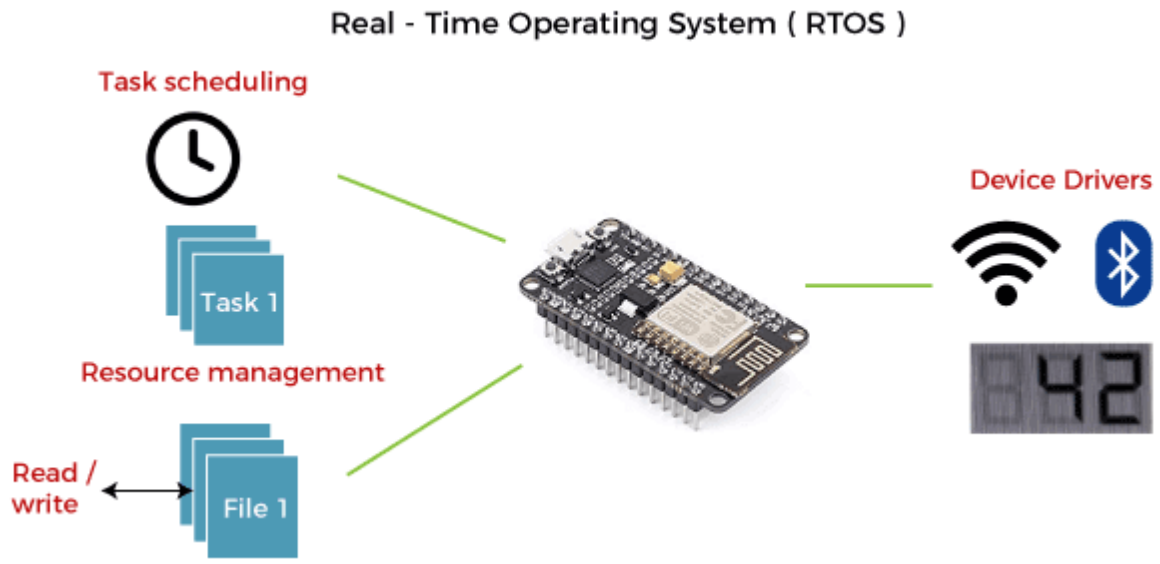
- Problem of data communication occurs.

Real-Time operating system

In this article, we understand the real time operating system in detail.

What do you mean by Real-Time Operating System?

A real-time operating system (RTOS) is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.



This process is completely uninterrupted unless a higher priority interrupt occurs during its execution. Therefore, there must be a strict hierarchy of priority among the interrupts. The interrupt with the highest priority must be allowed to initiate the process , while lower priority interrupts should be kept in a buffer that will be handled later. Interrupt management is important in such an operating system.

Real-time operating systems employ special-purpose operating systems because conventional operating systems do not provide such performance.

The various examples of Real-time operating systems are:

- MTS
- Lynx
- QNX
- VxWorks etc.

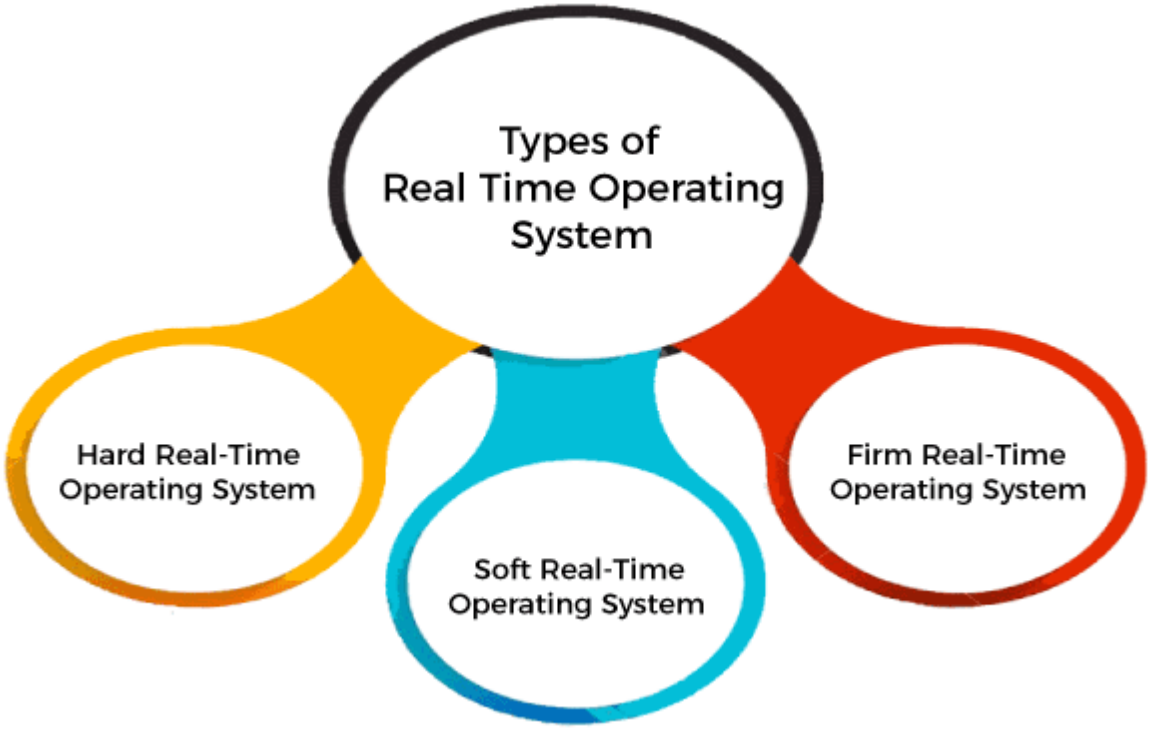
Applications of Real-time operating system (RTOS):

RTOS is used in real-time applications that must work within specific deadlines. Following are the common areas of applications of Real-time operating systems are given below.

- Real-time running structures are used inside the Radar gadget.
- Real-time running structures are utilized in Missile guidance.
- Real-time running structures are utilized in on line inventory trading.
- Real-time running structures are used inside the cell phone switching gadget.
- Real-time running structures are utilized by Air site visitors to manipulate structures.
- Real-time running structures are used in Medical Imaging Systems.
- Real-time running structures are used inside the Fuel injection gadget.
- Real-time running structures are used inside the Traffic manipulate gadget.
- Real-time running structures are utilized in Autopilot travel simulators.

Types of Real-time operating system

Following are the three types of RTOS systems are:



Hard Real-Time operating system:

In Hard RTOS, all critical tasks must be completed within the specified time duration, i.e., within the given deadline. Not meeting the deadline would result in critical failures such as damage to equipment or even loss of human life.

For Example,

Let's take an example of airbags provided by carmakers along with a handle in the driver's seat. When the driver applies brakes at a particular instance, the airbags grow and prevent the driver's head from hitting the handle. Had there been some delay even of milliseconds, then it would have resulted in an accident.

Similarly, consider an on-stock trading software. If someone wants to sell a particular share, the system must ensure that command is performed within a given critical time. Otherwise, if the market falls abruptly, it may cause a huge loss to the trader.

Soft Real-Time operating system:

Soft RTOS accepts a few delays via the means of the Operating system. In this kind of RTOS, there may be a closing date assigned for a particular job, but a delay for a small amount of time is acceptable. So, cut off dates are treated softly via means of this kind of RTOS.

For Example,

This type of system is used in Online Transaction systems and Livestock price quotation Systems.

Firm Real-Time operating system:

In Firm RTOS additionally want to observe the deadlines. However, lacking a closing date might not have a massive effect, however may want to purposely undesired effects, like a massive discount within the fine of a product.

For Example, this system is used in various forms of Multimedia applications.

Advantages of Real-time operating system:

The benefits of real-time operating system are as follows-:

- Easy to layout, develop and execute real-time applications under the real-time operating system.
- The real-time working structures are extra compact, so those structures require much less memory space.
- In a Real-time operating system, the maximum utilization of devices and systems.
- Focus on running applications and less importance to applications that are in the queue.
- Since the size of programs is small, RTOS can also be embedded systems like in transport and others.
- These types of systems are error-free.
- Memory allocation is best managed in these types of systems.

Disadvantages of Real-time operating system:

The disadvantages of real-time operating systems are as follows-

- Real-time operating systems have complicated layout principles and are very costly to develop.
- Real-time operating systems are very complex and can consume critical CPU cycles.

Layered Structure of Operating System

The Layered Architecture of Operating System is an operating system architecture that divides software components into layers, with hardware at the bottom of each layer. Each layer of the operating system is responsible for certain functions. The layered structure approach divides the operating system into layers and gives the user much more control over the system. The hardware is on the bottom layer (layer 0), and the user interface is on the top layer (layer N). These layers are designed in such a way that each layer only uses the functions of the lower-level layers. It makes debugging easier if lower-level layers are debugged and an error occurs while debugging. Because the lower-level layers have already been debugged, the error must be on that layer only.

Origin of Operating System Layers

The idea of designing a layered structure of operating system came to Djikstra and Powell where it was suggested that an operating system can be divided into different layers to increase modularity, and easy to maintain and debug-friendly system.

The operating system layers came into existence by MULTICS, an operating system developed around the early 1970s when the

Massachusetts Institute of Technology, Bell Labs and GE started collaborating on its development. MULTICS was the first of many that were about to launch in the following decades.

The layered approach in os will help into different parts and help in debugging. Taking an instance on debugging in general when an error is encountered, the error remains the same layer as all the underlying layers are already debugged and each layer can be separately debugged or tested without interference from other layers.

Different types of Layers in Layered Operating Systems

Layered Operating Systems are designed in such a manner that layers below a layer provide the service and the layers above are those that consume the service. This practice is used in software engineering to create applications that are modular, scalable and easy to maintain. Below is an illustration to help us understand the many layers that are fixed to the kernel. The layers sandwiched between the first and last layer are responsible for system services and management such as CPU scheduling, memory management, process management and I/O Management.

Architecture of Different Layers of Operating Systems

As we have some foundational knowledge of the layered structure of operating system, now we will be discussing each of the layers in detail as given in the ordered way below:-

Layer 1 – Hardware

This layer interacts with the internal components and works in partnership with devices such as monitors, speakers, webcam etc. It is regarded as the most autonomous layer in the layered structure of operating system.

Layer 2 – CPU Scheduling

CPU Scheduling is responsible to schedule the process that is yet to be run by the CPU. Processes lie in Job Queue when they are about to be executed by the CPU and remain in the Ready Queue when they are in memory and ready to be executed. Although there are multiple queues used for scheduling, the CPU Scheduler decides the process that will execute and the others that will wait.

Layer 3 – Memory Management

One of the layers in the middlemost region of the layered structure of operating system is responsible for allocating and deallocating memory to processes, here processes move to the main memory during execution and return back once they are run successfully, and the memory is freed. RAM and ROM are primarily the most popular examples.

Layer 4 – Process Management

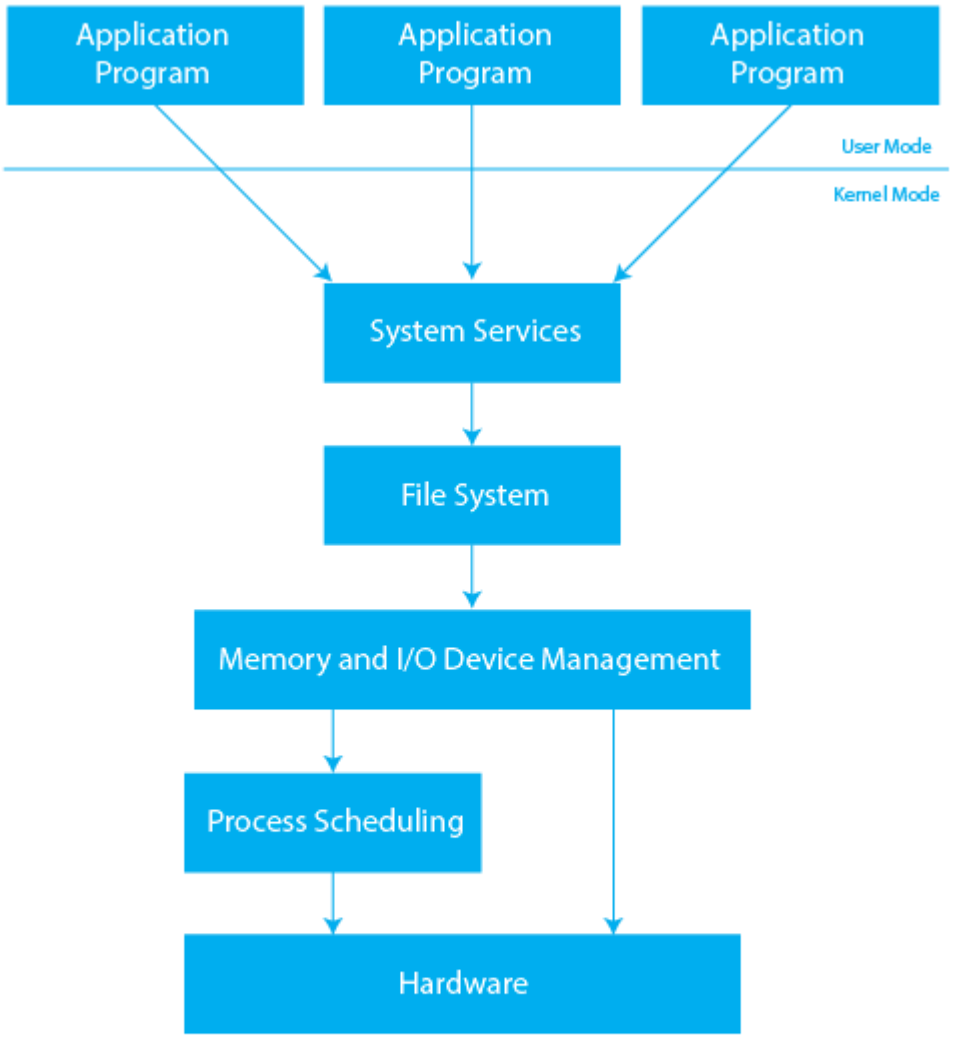
The layer decides which process will be executed by giving them the CPU and which will be waiting in the queue. The decision-making is performed with the help of scheduling algorithms like Shortest Job First, First Come First Serve, Shortest Remaining Time First, Priority Scheduling etc.

Layer 5 – I/O Buffer

This is the second layer from the top that is responsible for the interactivity of the user as input devices like the mouse, keyboard, and microphone are the source if communication between the computer and the user. Each device is assigned a buffer to avoid slow processing input by the user.

Layer 6 – User Application

This is the uppermost layer that gives the user easy and user-friendly access to the application to solve a real-world problem, play music or surf the internet etc.. It is also known as the Application Layer.



Advantages of Layered Operating System

Having seen the different layers of the architecture of the Layered Operating System, let us have a look at the advantages:-

1. **Abstraction**
   A layer is not concerned with the functioning of other layers in the structure which makes it suitable for debugging practices.

2. **Modularity**
   The operating system is divided into several units and each unit performs its task efficiently.

3. **Better Maintenance**
   Any updates or modifications made will restrict to the current layer and not impact the other layers in any manner.

4. **Debugging**
   Debugging can be performed well with the layer that is debugged will be corrected as the layers existing below are already functioning properly as a comparison to unreliable monolithic systems.

Disadvantages of Layered Operating System

However, there are certain drawbacks attached to the layered operating systems as well that can be justified as:-

1. **Lack of communication**
   As there is always a lack of communication between the non-neighbouring layers of the operating system, it can be problematic at times.

2. **Functionality**
   When the functionalities are interconnected, they cannot be separated from each other and in this case, layers are difficult to create.

3. **Slow processing**
   The layers are responsible for any request erupting and to reach another layer is a tedious and time-taking process, unlike monolithic kernels. More the number of layers, the lesser the efficiency.

4. **Complex Design**
   Layered Structure of Operating System has a complex design that must be carefully implemented as the services that are being used by a layer must be kept below that layer.

Applications of Layered Operating System

A layered operating system, also known as a layered software architecture, is a method of designing operating systems that divides functionality into distinct layers. Each layer provides services to the layer above it while also utilizing services from the layer beneath it. This modular design has several advantages, which make it widely used in a variety of applications. A layered operating system can be used for a variety of purposes, including:

1. Modularity and Maintainability: The layered structure allows developers to isolate different components of the operating system, making it easier to maintain and update individual layers without affecting the entire system. This modularity simplifies debugging, enhancements, and modifications.

2. Portability: The separation of concerns in the layered approach allows for easier porting of the operating system to different hardware architectures or platforms. As long as the hardware-dependent layer is adapted for the target platform, the higher layers can remain unchanged.

3. Extensibility: The layered design allows developers to add new features or functionality by introducing new layers or modifying existing ones. This extensibility enhances the capabilities of the operating system without disrupting the existing layers.

What are Virtual Machines in Operating System?

Operating SystemArduinoSoftware & Coding

A **virtual machine (VM)** is a virtual environment which functions as a virtual computer system with its own **CPU**, memory, network interface, and storage, created on a physical hardware system.

VMs are isolated from the rest of the system, and multiple VMs can exist on a single piece of hardware, like a server. That means, it as a simulated image of application software and operating system which is executed on a host computer or a server.

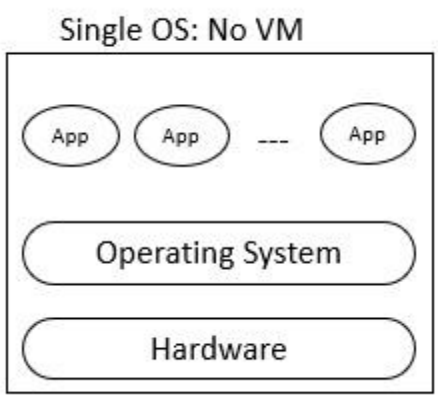
It has its own operating system and software that will facilitate the resources to virtual computers.

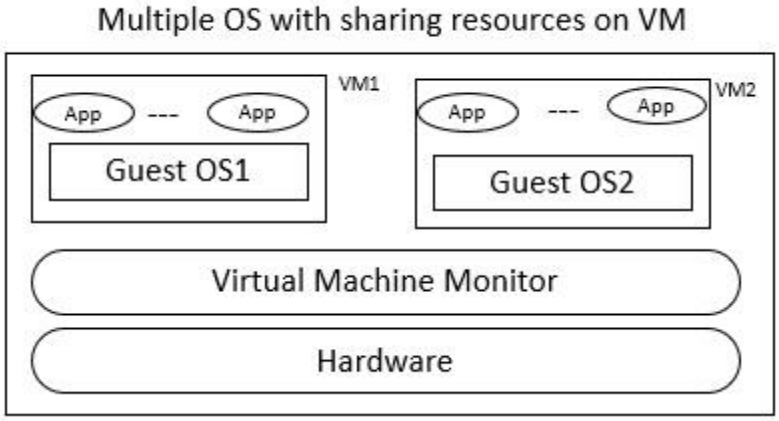
Characteristics of virtual machines

The characteristics of the virtual machines are as follows –

- Multiple OS systems use the same hardware and partition resources between virtual computers.
- Separate Security and configuration identity.
- Ability to move the virtual computers between the physical host computers as holistically integrated files.

The below diagram shows you the difference between the single OS with no VM and Multiple OS with VM –





Benefits

Let us see the major benefits of virtual machines for operating-system designers and users which are as follows –

- The multiple Operating system environments exist simultaneously on the same machine, which is isolated from each other.
- Virtual machine offers an instruction set architecture which differs from real computer.
- Using virtual machines, there is easy maintenance, application provisioning, availability and convenient recovery.

Virtual Machine encourages the users to go beyond the limitations of hardware to achieve their goals.

The operating system achieves virtualization with the help of a specialized software called a hypervisor, which emulates the PC client or server CPU, memory, hard disk, network and other hardware resources completely, enabling virtual machines to share resources.

The hypervisor can emulate multiple virtual hardware platforms that are isolated from each other allowing virtual machines to run Linux and window server operating machines on the same underlying physical host.

Basis for developing the OS

Create the illusion of having one or more objects to emulate the real object. It is closely related to abstraction. In developing the OS, abstraction provides simplification by combining multiple simple objects into a single complex object

Virtualization provides diversification and replication by creating the illusion of objects with desired characteristics.

Operating System Structure

Overview

An operating system is a design that enables user application programs to communicate with the hardware of the machine. The operating system should be built with the utmost care because it is such a complicated structure and should be simple to use and modify. Partially developing the operating system is a simple approach to accomplish this. Each of these components needs to have distinct inputs, outputs, and functionalities.

This article discusses many sorts of structures that implement operating systems, as listed below, as well as how and why they work. It also defines the operating system structure.

- Simple Structure
- Monolithic Structure
- Layered Approach Structure
- Micro-Kernel Structure
- Exo-Kernel Structure
- Virtual Machines

What is an operating System Structure?

We want a clear structure to let us apply an operating system to our particular needs because operating systems have complex structures. It is easier to create an operating system in pieces, much as we break down larger issues into smaller, more manageable subproblems. Every segment is also a part of the operating system. Operating system structure can be thought of as the strategy for connecting and incorporating various operating system components within the kernel. Operating systems are implemented using many types of structures, as will be discussed below:

SIMPLE STRUCTURE

It is the most straightforward operating system structure, but it lacks definition and is only appropriate for usage with tiny and restricted systems. Since the interfaces and degrees of functionality in this structure are clearly defined, programs are able to access I/O routines, which may result in unauthorized access to I/O procedures.

This organizational structure is used by the MS-DOS operating system:

- There are four layers that make up the MS-DOS operating system, and each has its own set of features.
- These layers include ROM BIOS device drivers, MS-DOS device drivers, application programs, and system programs.
- The MS-DOS operating system benefits from layering because each level can be defined independently and, when necessary, can interact with one another.
- If the system is built in layers, it will be simpler to design, manage, and update. Because of this, simple structures can be used to build constrained systems that are less complex.
- When a user program fails, the operating system as whole crashes.
- Because MS-DOS systems have a low level of abstraction, programs and I/O procedures are visible to end users, giving them the potential for unwanted access.

The following figure illustrates layering in simple structure:

Advantages of Simple Structure:

- Because there are only a few interfaces and levels, it is simple to develop.
- Because there are fewer layers between the hardware and the applications, it offers superior performance.

Disadvantages of Simple Structure:

- The entire operating system breaks if just one user program malfunctions.
- Since the layers are interconnected, and in communication with one another, there is no abstraction or data hiding.
- The operating system's operations are accessible to layers, which can result in data tampering and system failure.

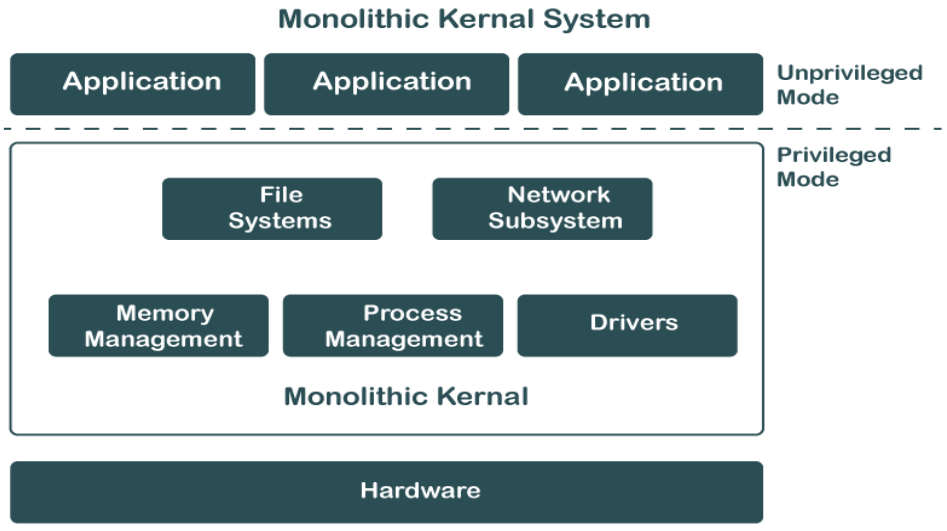
MONOLITHIC STRUCTURE

The monolithic operating system controls all aspects of the operating system's operation, including file management, memory management, device management, and operational operations.

The core of an operating system for computers is called the kernel (OS). All other System components are provided with fundamental services by the kernel. The operating system and the hardware use it as their main interface. When an operating system is built into a single piece of hardware, such as a keyboard or mouse, the kernel can directly access all of its resources.

The monolithic operating system is often referred to as the monolithic kernel. Multiple programming techniques such as batch processing and time-sharing increase a processor's usability. Working on top of the operating system and under complete command of all hardware, the monolithic kernel performs the role of a virtual computer. This is an old operating system that was used in banks to carry out simple tasks like batch processing and time-sharing, which allows numerous users at different terminals to access the Operating System.

The following diagram represents the monolithic structure:



Advantages of Monolithic Structure:

- Because layering is unnecessary and the kernel alone is responsible for managing all operations, it is easy to design and execute.

- Due to the fact that functions like memory management, file management, process scheduling, etc., are implemented in the same address area, the monolithic kernel runs rather quickly when compared to other systems. Utilizing the same address speeds up and reduces the time required for address allocation for new processes.

Disadvantages of Monolithic Structure:

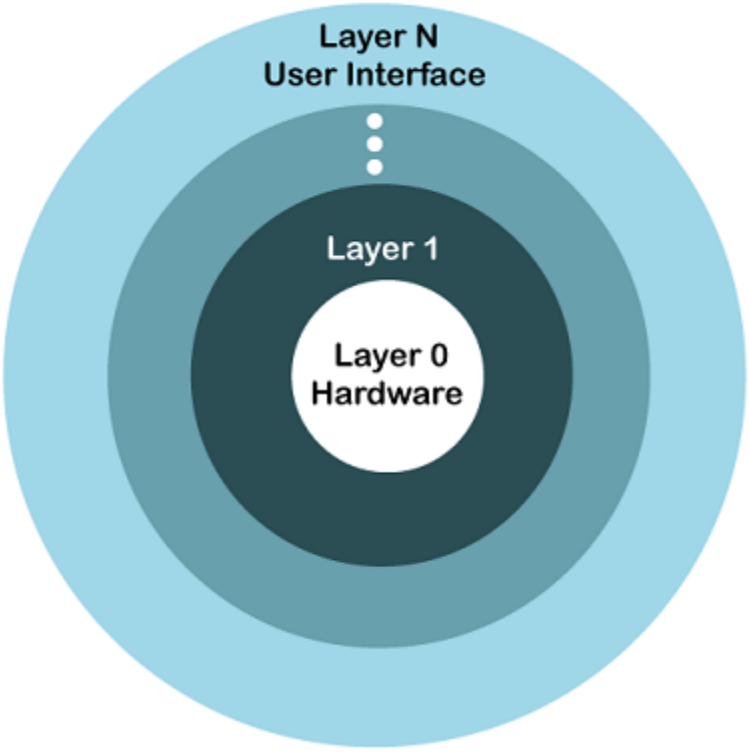
- The monolithic kernel's services are interconnected in address space and have an impact on one another, so if any of them malfunctions, the entire system does as well.
- It is not adaptable. Therefore, launching a new service is difficult.

LAYERED STRUCTURE

The OS is separated into layers or levels in this kind of arrangement. Layer 0 (the lowest layer) contains the hardware, and layer 1 (the highest layer) contains the user interface (layer N). These layers are organized hierarchically, with the top-level layers making use of the capabilities of the lower-level ones.

The functionalities of each layer are separated in this method, and abstraction is also an option. Because layered structures are hierarchical, debugging is simpler, therefore all lower-level layers are debugged before the upper layer is examined. As a result, the present layer alone has to be reviewed since all the lower layers have already been examined.

The image below shows how OS is organized into layers:



Advantages of Layered Structure:

- Work duties are separated since each layer has its own functionality, and there is some amount of abstraction.
- Debugging is simpler because the lower layers are examined first, followed by the top layers.

Disadvantages of Layered Structure:

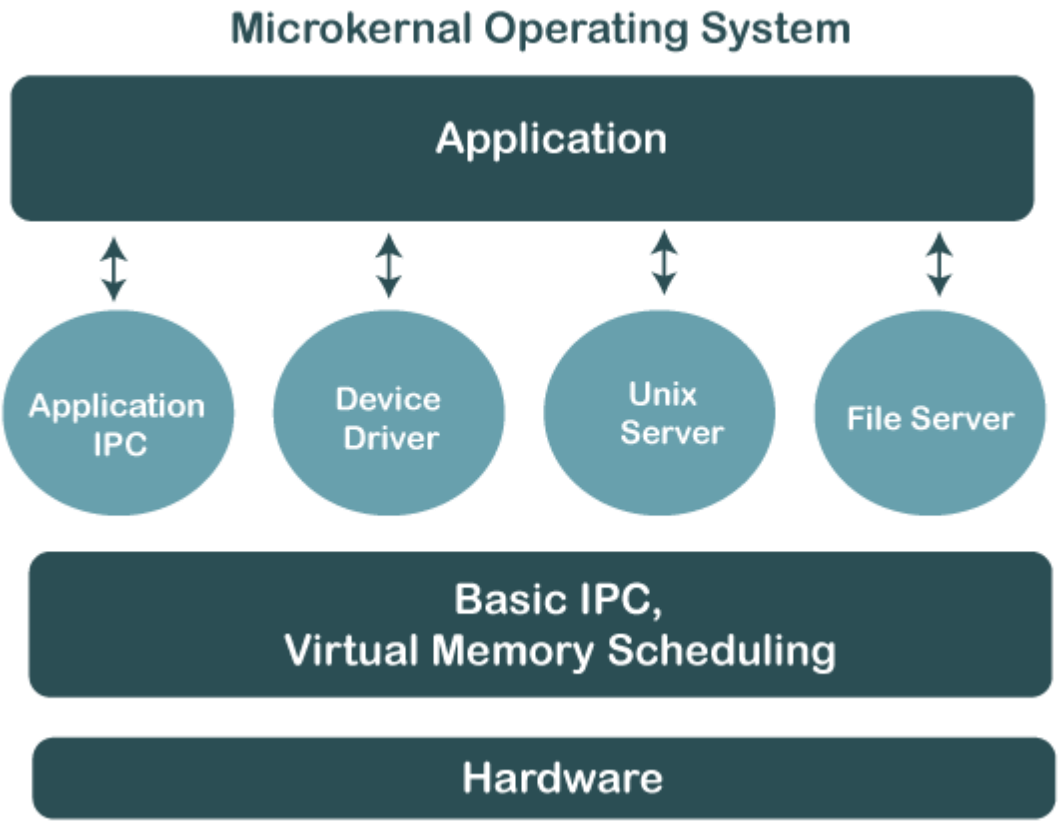
- Performance is compromised in layered structures due to layering.
- Construction of the layers requires careful design because upper layers only make use of lower layers' capabilities.

MICRO-KERNEL STRUCTURE

The operating system is created using a micro-kernel framework that strips the kernel of any unnecessary parts. Systems and user applications are used to implement these optional kernel components. So, Micro-Kernels is the name given to these systems that have been developed.

Each Micro-Kernel is created separately and is kept apart from the others. As a result, the system is now more trustworthy and secure. If one Micro-Kernel malfunctions, the remaining operating system is unaffected and continues to function normally.

The image below shows Micro-Kernel Operating System Structure:



Advantages of Micro-Kernel Structure:

- It enables portability of the operating system across platforms.
- Due to the isolation of each Micro-Kernel, it is reliable and secure.
- The reduced size of Micro-Kernels allows for successful testing.
- The remaining operating system remains unaffected and keeps running properly even if a component or Micro-Kernel fails.

Disadvantages of Micro-Kernel Structure:

- The performance of the system is decreased by increased inter-module communication.
- The construction of a system is complicated.

Microkernel in Operating System

In this article, you will learn about the microkernel with its architecture, advantages and disadvantages. But before discussing the microkernel, you must know about the kernel.

What is a kernel?

It is the central component of an OS that handles system resources. It also acts as a bridge between the computer's application and hardware. It is one of the initial programs that is loaded when the computer boots up. When an OS is loaded, the kernel is the first component that loads into memory and rests there until the OS is shut down. It is in charge of various activities, including task management, disk management, and memory management.

What is Microkernel?

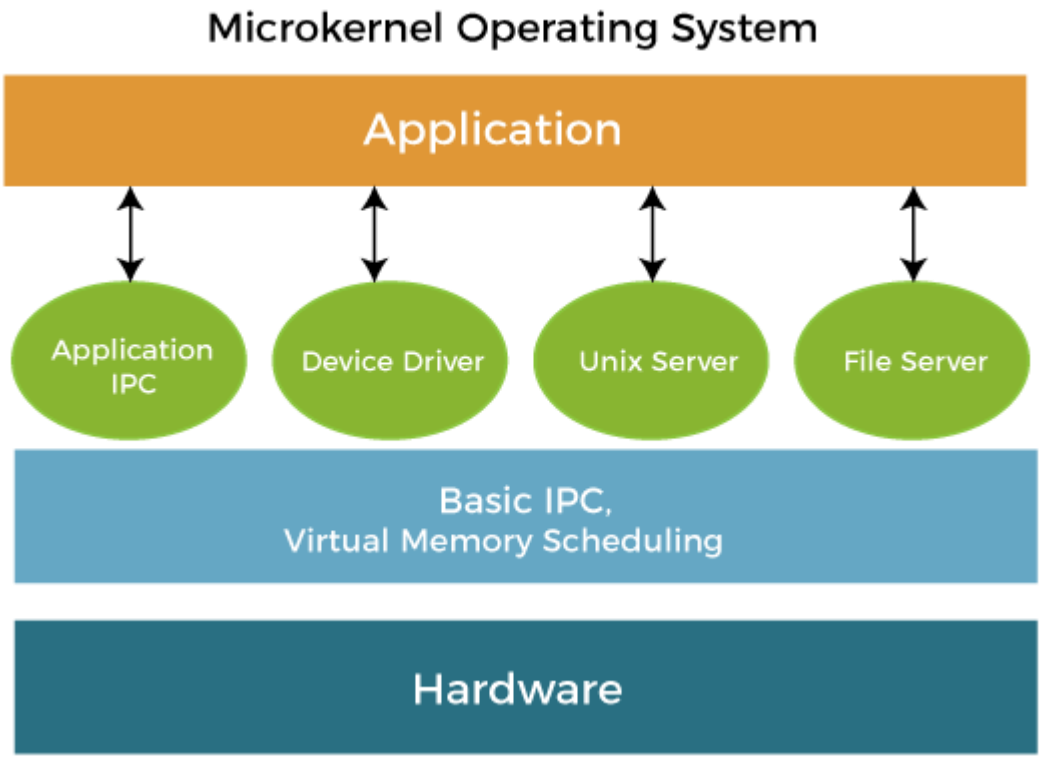
The microkernel is one of the kernel's classifications. Being a kernel, it handles all system resources. On the other hand, the user and kernel services in a microkernel are implemented in distinct address spaces. **User services** are kept in **user address space**, while **kernel services** are kept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

It provides a minimal amount of process and memory management services. The interaction between the client application and services running in user address space is established via message passing that helps to reduce the speed of microkernel execution. The OS is unaffected because kernel and user services are isolated, so if any of the user services fails, the kernel service is unaffected. It is extendable because new services are added to the user address space, hence requiring no changes in kernel space. It's also lightweight, secure, and reliable.

Microkernels and their user environments are typically used in C++ or C languages with a little assembly. On the other hand, other implementation programming languages may be possible with some high-level code.

Architecture of Microkernel

A microkernel is a minimum needed of software required to implement an operating system correctly. Memory, process scheduling methods, and fundamental inter-process communication are all included.



In the above figure, the microkernel includes basic needs like process scheduling mechanisms, memory, and interprocess communication. It is the only program that executes at the privileged level, i.e., kernel mode. The OS's other functions are moved from the kernel-mode and execute in the user mode.

The microkernel ensures that the code may be easily controlled because the services are split in the user space. It means some code runs in the kernel mode, resulting in improved security and stability.

Since the kernel is the most crucial OS component, it is responsible for the essential services. As a result, under this design, only the most significant services are present inside the kernel in this architecture. In contrast, the rest operating system services are available inside the system application software. As a result, users can interact with such unnecessary services within the system application. The microkernel is entirely responsible for the operating system's most significant services, which are as follows:

1. **Inter-Process Communication**
2. **Memory Management**
3. **CPU Scheduling**

Inter-Process Communication

Interprocess communication refers to how processes interact with one another. A process has several threads. In the kernel space, threads of any process interact with one another. Messages are sent and received across threads using ports. At the kernel level, there are several ports like process port, exceptional port, bootstrap port, and registered port. All of these ports interact with user-space processes.

Memory Management

Memory management is the process of allocating space in main memory for processes. However, there is also the creation of virtual memory for processes. Virtual memory means that if a process has a bigger size than the main memory, it is partitioned into portions and stored. After that, one by one, every part of the process is stored in the main memory until the CPU executes it.

CPU Scheduling

CPU scheduling refers to which process the CPU will execute next. All processes are queued and executed one at a time. Every process has a level of priority, and the process with the highest priority is performed out first. CPU scheduling aids in optimizing CPU utilization. In addition, resources are being used more efficiently. It also minimizes the waiting time. Waiting time shows that a process takes less time in the queue and that resources are allocated to the process more quickly. CPU scheduling also reduces response and turnaround times.

Components of Microkernel

A microkernel contains only the system's basic functions. A component is only included in the microkernel if putting it outside would disrupt the system's operation. The user mode should be used for all other non-essential components. The minimum functionalities needed in the microkernel are as follows:

1. In the microkernel, processor scheduling algorithms are also required. Process and thread schedulers are included.
2. Address spaces and other memory management mechanisms should be incorporated in the microkernel. Memory protection features are also included.
3. Inter-process communication (IPC) is used to manage servers that execute their own address spaces.

Advantages and Disadvantages of Microkernel

Various advantages and disadvantages of the microkernel are as follows:

Advantages

1. Microkernels are secure since only those parts are added, which might disturb the system's functionality.
2. Microkernels are modular, and the various modules may be swapped, reloaded, and modified without affecting the kernel.
3. Microkernel architecture is compact and isolated, so it may perform better.
4. The system expansion is more accessible, so it may be introduced to the system application without disrupting the kernel.

5. When compared to monolithic systems, microkernels have fewer system crashes. Furthermore, due to the modular structure of microkernels, any crashes that do occur are simply handled.
6. The microkernel interface helps in enforcing a more modular system structure.
7. Server failure is treated the same as any other user program failure.
8. It adds new features without recompiling.

Disadvantages

1. When the drivers are implemented as procedures, a context switch or a function call is needed.
2. In a microkernel system, providing services are more costly than in a traditional monolithic system.
3. The performance of a microkernel system might be indifferent and cause issues.

4. Difference Between Microkernel and Monolithic Kernel

Parameters	Monolithic kernel	M
Basic	It is a large process running in a single address space	It can be broken down int servers.
Code	In order to write a monolithic kernel, less code is required.	In order to write a microk
Security	If a service crashes, the whole system collapses in a monolithic kernel.	If a service crashes, it ne microkernel.
Communication	It is a single static binary file	Servers communicate thr
Example	Linux, BSDs, Microsoft Windows (95,98, Me), Solaris, OS-9, AIX, DOS, XTS-400, etc.	L4Linux, QNX, SymbianK