Operating System Structures

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:

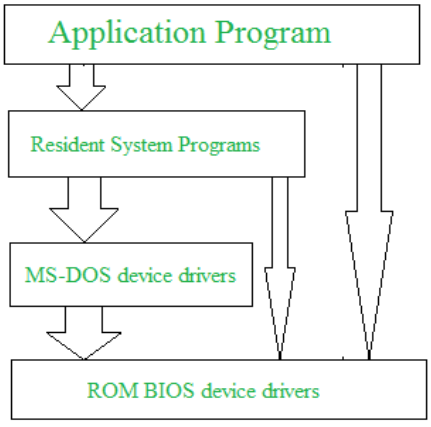
1. SIMPLE STRUCTURE

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 can 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.

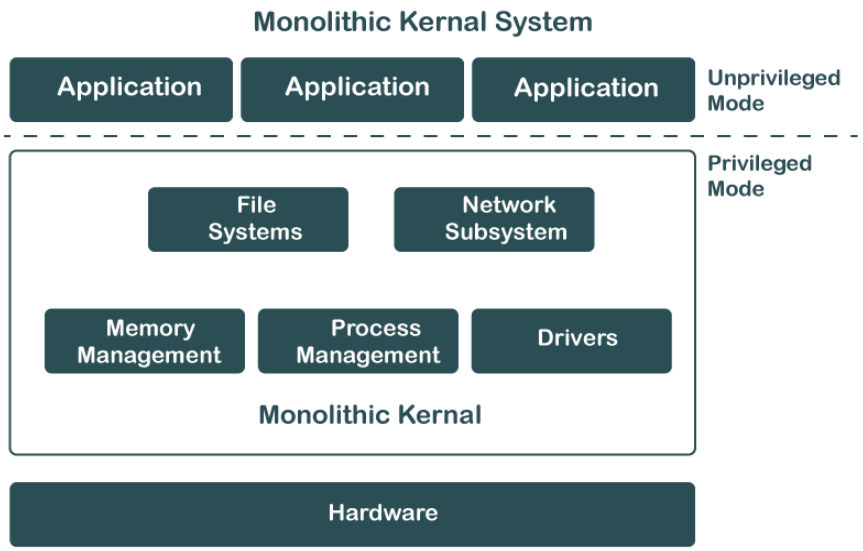
2. MONOLITHIC STRUCTURE

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.

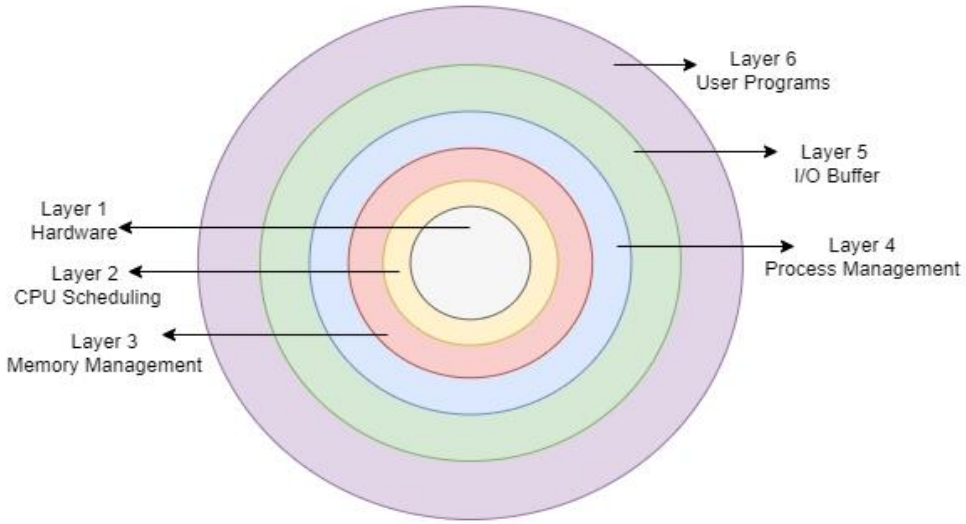
3. LAYERED STRUCTURE

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:



There are mainly six main layers in layered operating systems. These are:

Layer-1: Hardware

It is the lowest layer in the operating system architecture. The layer handles the hardware devices, where all the hardware devices are installed.

Layer-2: CPU Layer

All the tasks and CPU processes are scheduled in this layer.

Layer-3: Memory Management

The layer handles memory. It moves processes from the disk(secondary storage) to primary memory for execution and returns executed processes to the disk. The layer helps to utilize memory efficiently.

Layer-4: Process Management

The layer assigns the CPU to execute processes.

Layer-5: Input-Output Buffer

The layer provides an I/O device buffer and ensures that all the input-output devices work in synchronization. It allows the user to interact with the system.

Layer-6: User Programs

It is the highest layer in a layered operating system, and it is associated with the user programs like word processors, browsers, etc.

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:

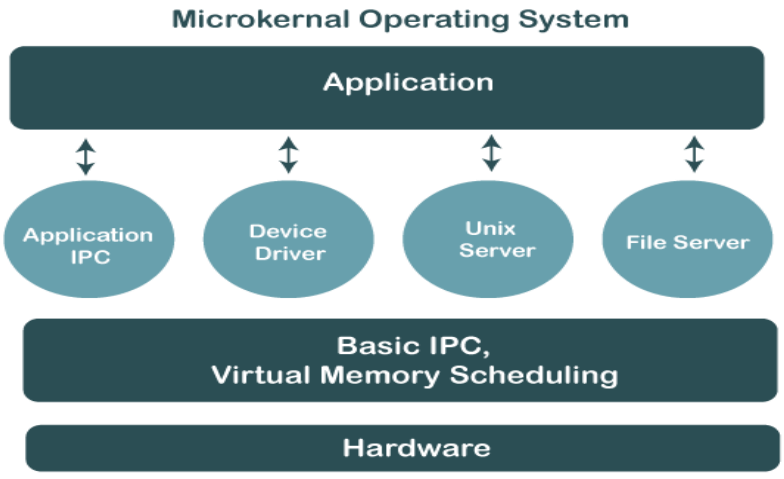
- Performance is compromised in layered structures due to layering.
- Construction of the layers requires careful design because upper layers only make use of lower layers' capabilities.

4. 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.

5. HYBRID-KERNEL STRUCTURE

Hybrid-kernel structure is nothing but just a combination of both monolithic-kernel structure and micro-kernel structure. Basically, it combines properties of both monolithic and micro-kernel and make a more advance and helpful approach. It implement speed and design of monolithic and modularity and stability of micro-kernel structure.

Advantages of Hybrid-Kernel Structure

- It offers good performance as it implements the advantages of both structure in it.
- It supports a wide range of hardware and applications.
- It provides better isolation and security by implementing micro-kernel approach.
- It enhances overall system reliability by separating critical functions into micro-kernel for debugging and maintenance.

Disadvantages of Hybrid-Kernel Structure

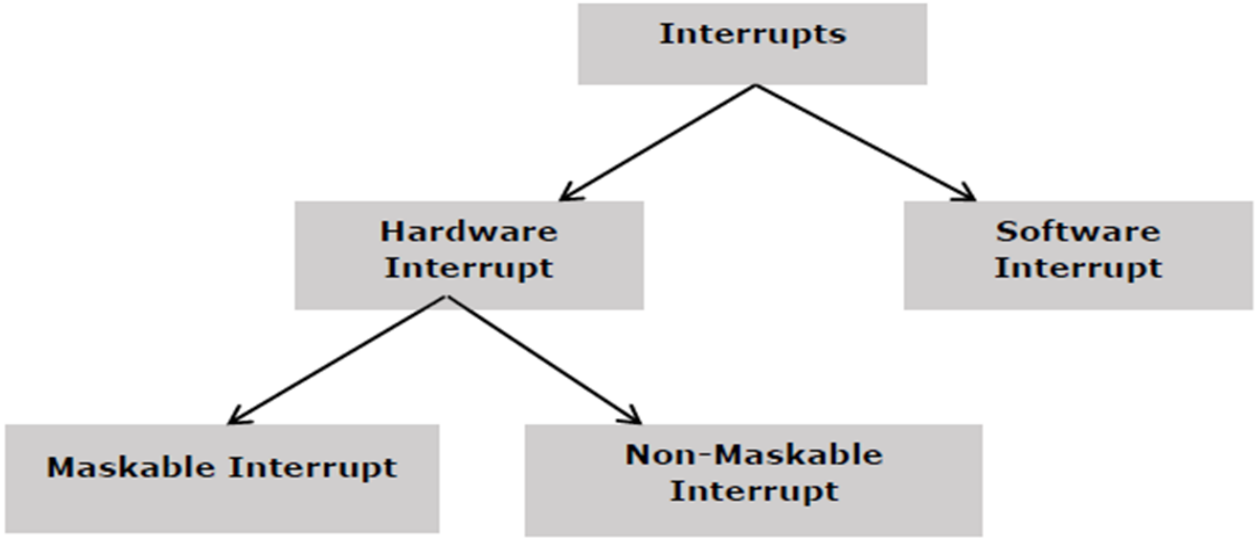
- It increases overall complexity of system by implementing both structure (monolithic and micro) and making the system difficult to understand.
- The layer of communication between micro-kernel and other component increases time complexity and decreases performance compared to monolithic kernel.

Difference Between Micro-Kernel and Monolithic Kernel

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

Interrupt Handling

The interrupt is a signal emitted by hardware or software when a process or an event needs immediate attention. It alerts the processor to a high-priority process requiring interruption of the current working process.

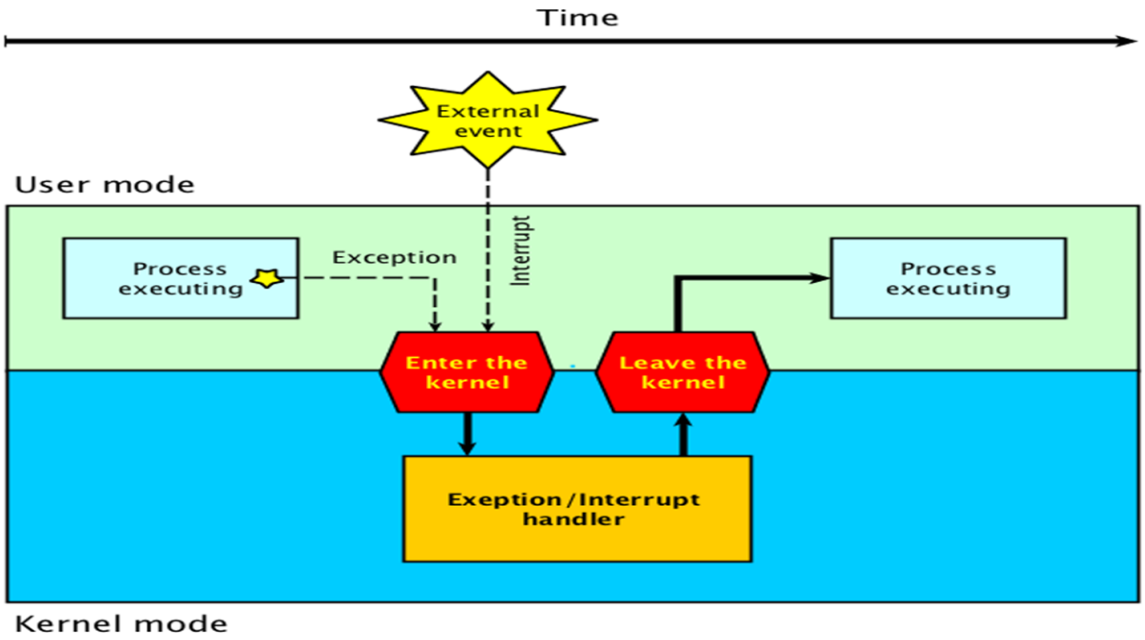


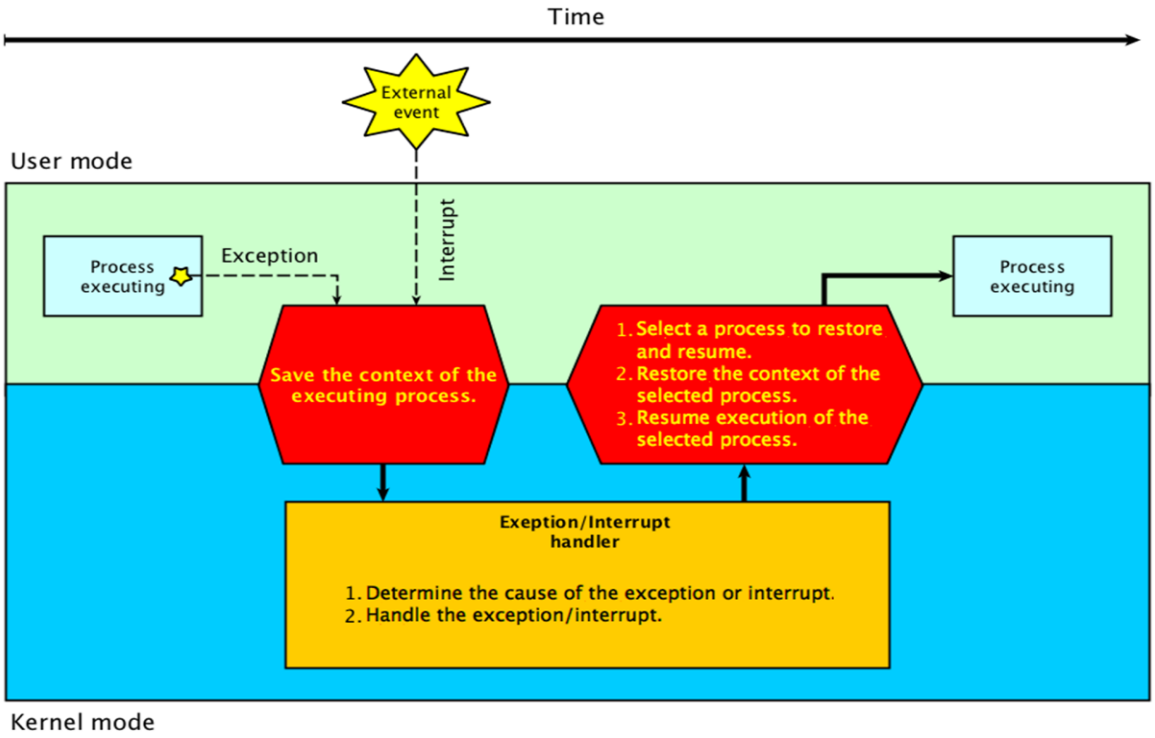
Types of Interrupt:

Interrupts have two types:

1. **Hardware Interrupt:** The interrupt signal generated from external devices and i/o devices are made interrupt to CPU when the instructions are ready.
- **Maskable Interrupt** − The hardware interrupts that can be delayed when a highest priority interrupt has occurred to the processor.
- **Non Maskable Interrupt** − The hardware that cannot be delayed and immediately be serviced by the processor.
2. **Software Interrupt**: The software interrupt occurs by executing a dedicated. The interrupt signal generated from internal devices and software programs need to access any system call then software interrupts are present.

When an interrupt occurs, execution transition from user mode to kernel mode where the interrupt is handled. When the interrupt has been handled execution resumes in user space.





- While entering the kernel, the context (values of all CPU registers) of the currently executing process must first be saved to memory.
- The kernel is now ready to handle the exception/interrupt.
- Determine the cause of the exception/interrupt.
- Handle the exception/interrupt.
- When the exception/interrupt have been handled the kernel performs the following steps:

- Select a process to restore and resume.
- Restore the context of the selected process.
- Resume execution of the selected process.

Bare Machine:

So basically Bare machine is logical hardware which is used to execute the program in the processor without using the operating system. as of now, we have studied that we can't execute any process without the Operating system. But yes with the help of the Bare machine we can do that.

Initially, when the operating systems are not developed, the execution of an instruction is done by directly on hardware without using any interfering hardware, at that time the only drawback was that the Bare machines accepting the instruction in only machine language, due to this those person who has

sufficient knowledge about Computer field are able to operate a computer. so after the development of the operating system Bare machine is referred to as inefficient.