What is a System Structure for an Operating System?

- Because operating systems have complex structures, we want a structure that is easy to understand so that we can adapt an operating system to meet our specific needs.
- Similar to how we break down larger problems into smaller, more manageable subproblems, building an operating system in pieces is simpler. The operating system is a component of every segment.
- The strategy for integrating different operating system components within the kernel can be thought of as an operating system structure. As will be discussed below, various types of structures are used to implement operating systems.

Operating Systems Structures

1.Simple Structure

2.Monolithic Structure

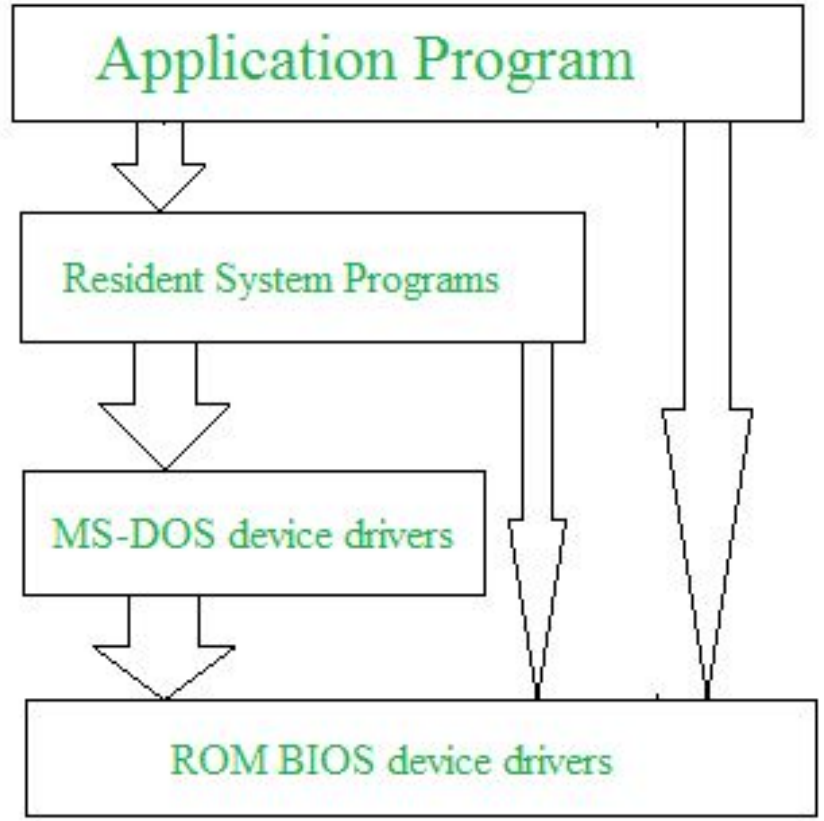
3.Layered Structure

4.Micro -Kernel Structure

5.Exo-Kernel Structure

6.Virtual Machines

Simple structure:



- Simple structure operating systems do not have well-defined structures and are small, simple, and limited.
- 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.
- If any one of program gets debug means all will get crassed, so easily entire system will crassed.
- Advantages of Simple structure :
- we can assess the base hardware all levels
- examples: MS-DOS, UNIX,
- Easy develope, Simle functionalities
- Deliver better applications

- Disadvantages of Simple Structure:
- Security issue high
- Simple structure does not well protected,
- Simple structure does not well designed,
- Simple structure does not good structured
- **3D Animated vedio Link for Operating System**
- https://youtu.be/oejEdmqUgJM?si=bD2P_LZ4OgnEaD7p

Monolithic Structure

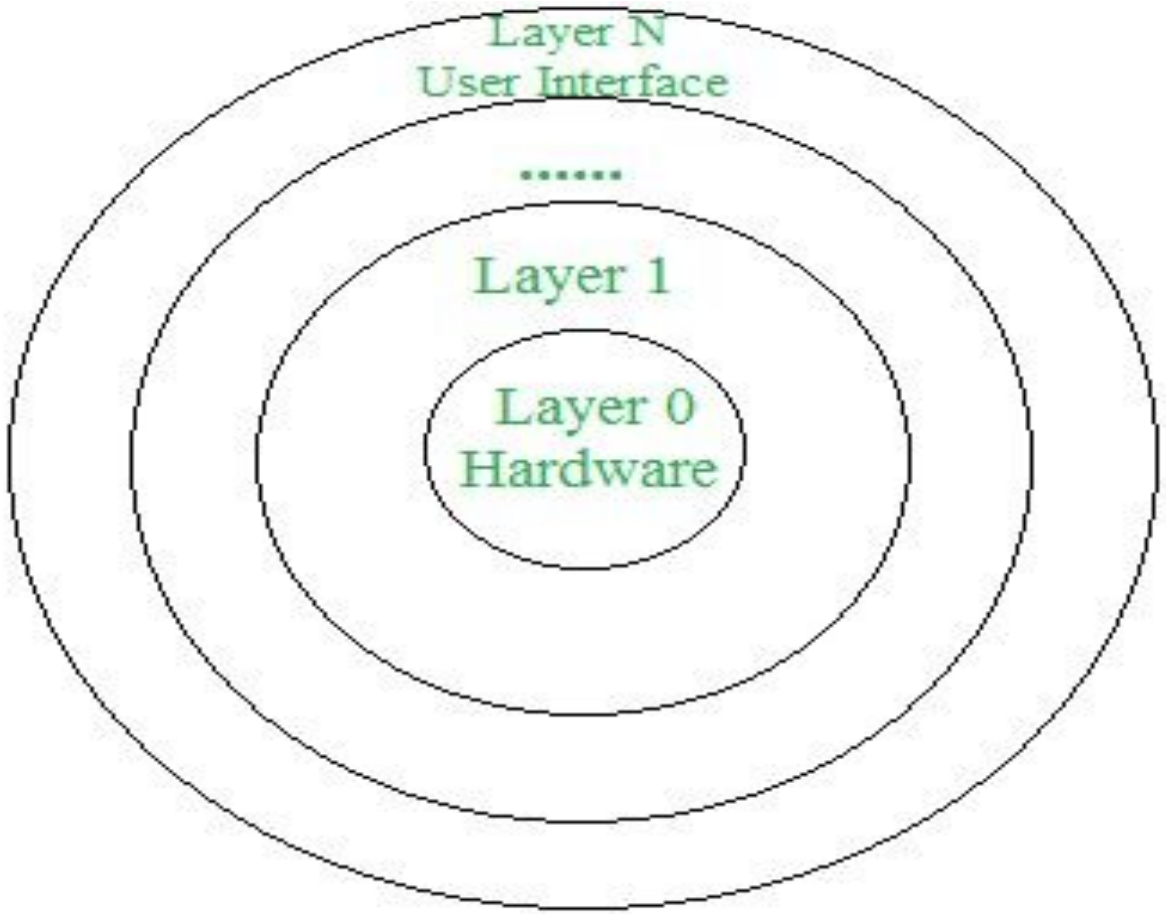
The Users		
Shells and Commands Compilers and Interpreters system libraries		
System -Call Interface to the Kernel		
Signals terminal handling char I/O Systems terminal drivers	File system swapping block I/O system disk and tape drivers	Cpu sheduling page replacement demand paging virutual memory
Kernal Interface to the hardware Terminal controllers Device controllers Memory controllers		
base hardware Kernal Space--->> we can access resources directly		

Monolithic Structure:

- Banking System used this (MLS) Batch process,Timesharing
- Virtual machine likely act this so its can controller the whole h/w
- this entire (MLS) worked with kernal space
- We can use this structure all type of needed functionalities , its doesnot have the work nature.
- Examples: UNIX, Windows 95, Windows, Linux,Solaris,IBM[aix]

- Advantage of MLS:
- Simple and easy to implement
- Faster execution due to direct access
- It delivers better application performance because of the few interfaces between the application program and the hardware.
- It is easy for kernel developers to develop such an operating system.
- Disadvantages of Monolithic Structure
- The structure is very complicated, as no clear boundaries exist between modules.
- It does not enforce data hiding in the operating system.
- Add or remove feauters very difficult
- Security issue high.

Layered structure:

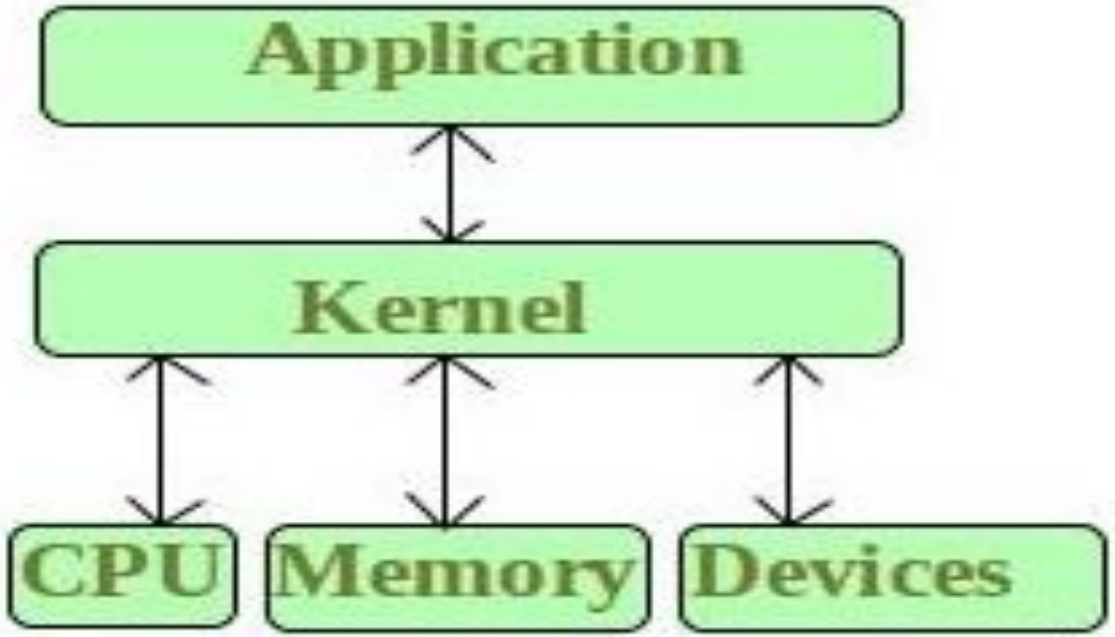


- An OS can be broken into pieces and retain much more control over the system. In this structure, the OS is broken into a number of layers (levels).
- The bottom layer (layer 0) is the hardware, and the topmost layer (layer N) is the user interface.
- These layers are so designed that each layer uses the functions of the lower-level layers.
- This simplifies the debugging process, 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.

- Implementation of the layers:
- The Outermost layer must be the user interface layer
- The innermost layer must be the hardware layer
- A particular layer can access the all layers present below it, but it cannot access the layers present above it.
- That is layer n-1 can access all the layers from n-2 to 0 but it cannot access the nth layer
- Thus if the user layer wants to interact with the hardware layer, the response will be travelled through all the layers from n-1 to 1
- Each layer must be designed and implemented such that it will need only the service provided by the layers below it

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

Micro -Kernel Structure:



- A microkernel is a type of operating system kernel that is designed to provide only the most basic services required for an operating system to function, such as memory management and process scheduling.
- Other services, such as device drivers and file systems, are implemented as user-level processes that communicate with the microkernel via message passing.
- This design allows the operating system to be more modular and flexible than traditional monolithic kernels, which implement all operating system services in kernel space.

The main advantage of a microkernel architecture:

- The main advantage of a microkernel architecture is that it provides a more secure and stable operating system. Since only the most essential services run in kernel space, the attack surface of the operating system is reduced, making it more difficult for an attacker to exploit vulnerabilities.
- Additionally, if a user-level process crashes, it will not affect the stability of the entire system, since the microkernel is responsible only for managing processes and memory.

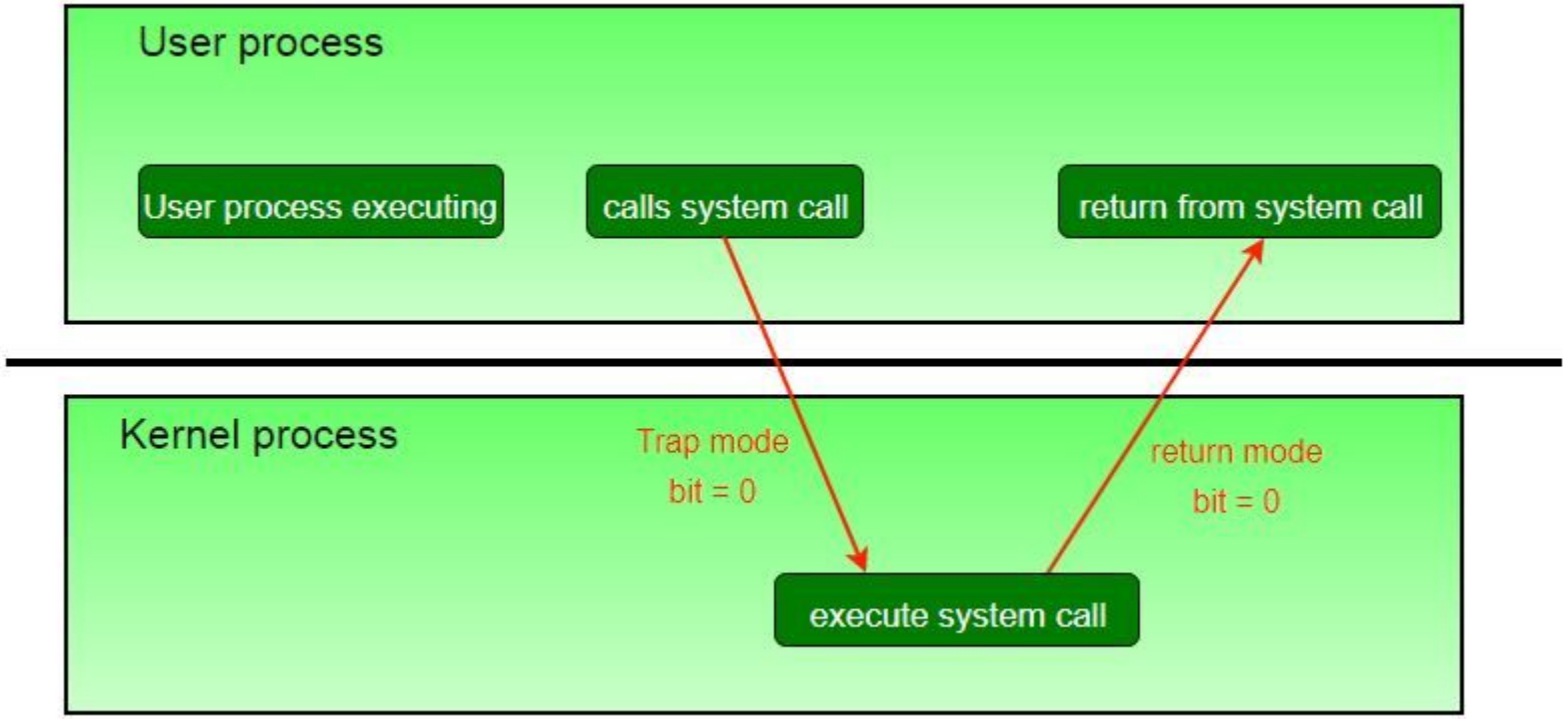
- Another advantage of a microkernel architecture is that it makes the operating system more modular and flexible. Since services are implemented as user-level processes, it is easier to add, remove, or replace services without affecting other parts of the system.
- This makes it easier to customize the operating system to meet specific requirements.

Disadvantages to a microkernel architecture:

- One major disadvantage is that message passing between user-level processes can be slower than direct system calls in a monolithic kernel.
- This can affect the performance of the operating system, especially in high-performance applications. Additionally, the modular design of a microkernel can lead to increased complexity, which can make it more difficult to develop and maintain the operating system.
- Overall, a microkernel architecture can provide a more secure and flexible operating system, but it may also come with some performance and complexity trade-offs. The choice between a microkernel and a monolithic kernel architecture depends on the specific needs and requirements of the operating system being developed.

- Kernel is the core part of an operating system that manages system resources. It also acts as a bridge between the application and hardware of the computer. It is one of the first programs loaded on start-up (after the Bootloader).
- Kernel mode and User mode of CPU operation
- The CPU can execute certain instructions only when it is in kernel mode. These instructions are called privilege instruction.
- They allow the implementation of special operations whose execution by the user program could interface with the functioning of the operating system or activity of another user program. For example, instruction for managing memory protection.
-

Kernel mode and User mode of CPU operation :



Kernel mode and User mode of CPU operation :

- The operating system puts the CPU in kernel mode when it is executing in the kernel so, that kernel can execute some special operation.
- The operating system puts the CPU in user mode when a user program is in execution so, that the user program cannot interface with the operating system program.
- User-level instruction does not require special privilege. Example are ADD,PUSH,etc.

Exo-Kernel Structure

- Exokernel is an operating system developed at MIT to provide application-level management of hardware resources. By separating resource management from protection, the exokernel architecture aims to enable application-specific customization. Due to its limited operability, exokernel size typically tends to be minimal.
- The OS will always have an impact on the functionality, performance, and scope of the apps that are developed on it because it sits in between the software and the hardware. The exokernel operating system makes an attempt to address this problem by rejecting the notion that an operating system must provide abstractions upon which to base applications. The objective is to limit developers use of abstractions as little as possible while still giving them freedom.

Advantages of Exo-kernel

- Support for improved application control.
- Separates management from security.
- It improves the performance of the application.
- A more efficient use of hardware resources is made possible by accurate resource allocation and revocation.
- It is simpler to test and create new operating systems.
- Each user-space program is allowed to use a custom memory management system.

Disadvantages of Exo-kernel

- A decline in consistency.
- Exokernel interfaces have a complex architecture.

VMs (virtual machines)

- Based on our needs, a virtual machine abstracts the hardware of our personal computer, including the CPU, disc drives, RAM, and NIC (Network Interface Card), into a variety of different execution contexts, giving us the impression that each execution environment is a different computer. An illustration of it is a virtual box.
- An operating system enables us to run multiple processes concurrently while making it appear as though each one is using a different processor and virtual memory by using CPU scheduling and virtual memory techniques.

- The fundamental issue with the virtual machine technique is disc systems. Let’s say the physical machine only has three disc drives, but it needs to host seven virtual machines.
- The program that creates virtual machines would need a significant amount of disc space in order to provide virtual memory and spooling, so it should be clear that it is impossible to assign a disc drive to every virtual machine. The answer is to make virtual discs available.

some sample questions

- What is Microkernel?
- Microkernel Architecture ?
- What is a Hybrid Kernel?
- What is a Virtualization-based Kernel?