Operating Systems Structures

The operating system can be implemented with the help of various structures. The structure of the OS depends mainly on how the various standard components of the operating system are interconnected and merge into the kernel. This article discusses a variety of operating system implementation structures and explains how and why they function.

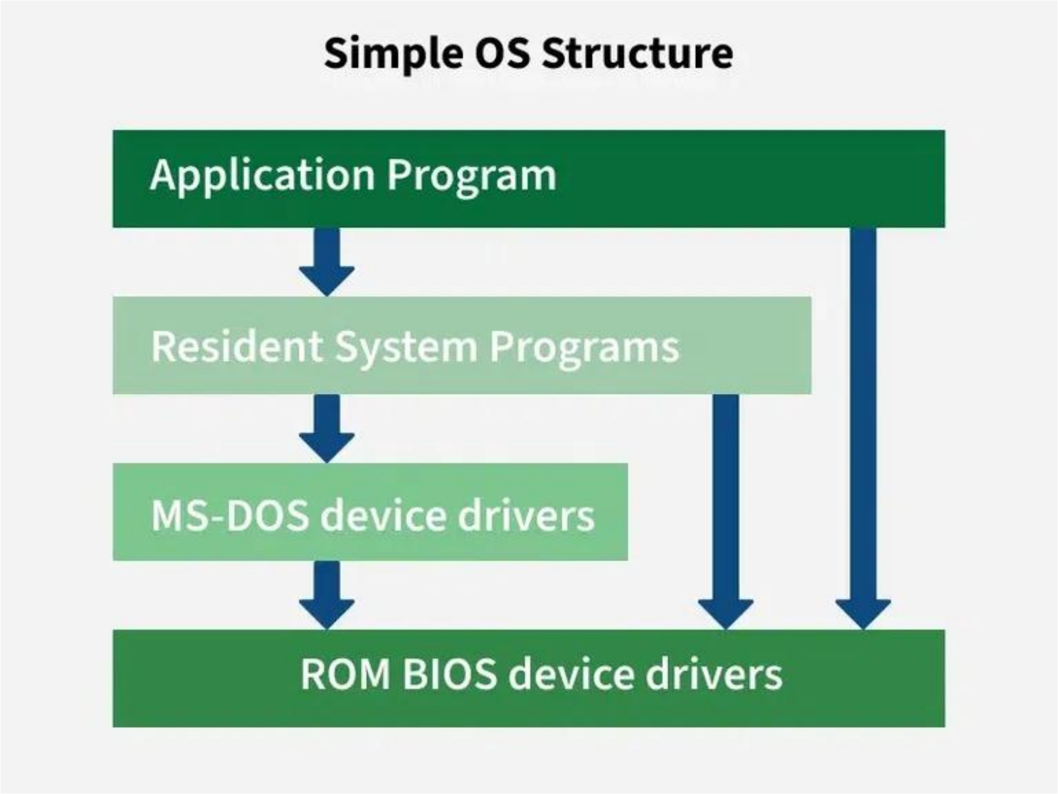
Types of Operating Systems Structures

Depending on this, we have the following structures in the operating system:

- Simple Structure
- Monolithic Structure
- Micro-Kernel Structure
- Hybrid-Kernel Structure
- Exo-Kernel Structure
- Layered Structure
- Modular Structure
- 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.



Advantages of Simple Structure

- 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 Simple Structure

- The structure is very complicated, as no clear boundaries exist between modules.
- It does not enforce data hiding in the operating system.

Monolithic Structure

A monolithic structure is a type of operating system architecture where the entire operating system is implemented as a single large process in kernel mode. Essential operating system services, such as process management, memory management, file systems, and device drivers, are combined into a single code block.

Advantages of Monolithic Structure

- Performance of Monolithic structure is fast as since everything runs in a single block, therefore communication between components is quick.
- It is easier to build because all parts are in one code block.

Disadvantages of Monolithic Architecture

- It is hard to maintain as a small error can affect entire system.

- There are also some security risks in the Monolithic architecture.

Micro-Kernel Structure

Micro-Kernel 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 rest of the operating system remains untouched. Mac OS is an example of this type of OS.

Advantages of Micro-kernel Structure

- It makes the operating system portable to various platforms.
- As microkernels are small so these can be tested effectively.

Disadvantages of Micro-kernel Structure

- Increased level of inter module communication degrades system performance.

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 implements 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.

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

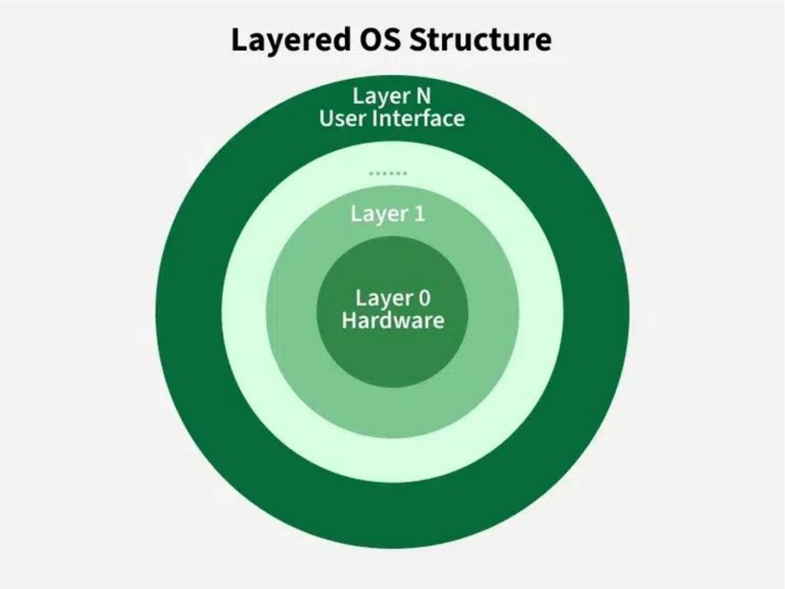
- Support for improved application control.
- Separates management from security.
- It improves the performance of the application.
- A more efficient use of hardware resources is made possible by accurate resource allocation and revocation.
- It is simpler to test and create new operating systems.
-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

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.



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.

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.