Operating Systems Structures

Last Updated : 16 Jul, 2024

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 melded into the kernel.

A design known as an operating system enables user application programs to communicate with the machine's hardware. Given its complex design and need to be easy to use and modify, the operating system should be constructed with the utmost care. A straightforward way to do this is to supernaturally develop the operating system. These parts must each have unique inputs, outputs, and functionalities.

This article discusses a variety of operating system implementation structures, including those listed below, as well as how and why they function. Additionally, the operating system structure is defined.

What is a System Structure for an Operating System?

A system structure for an operating system is like the blueprint of how an OS is organized and how its different parts interact with each other. 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.

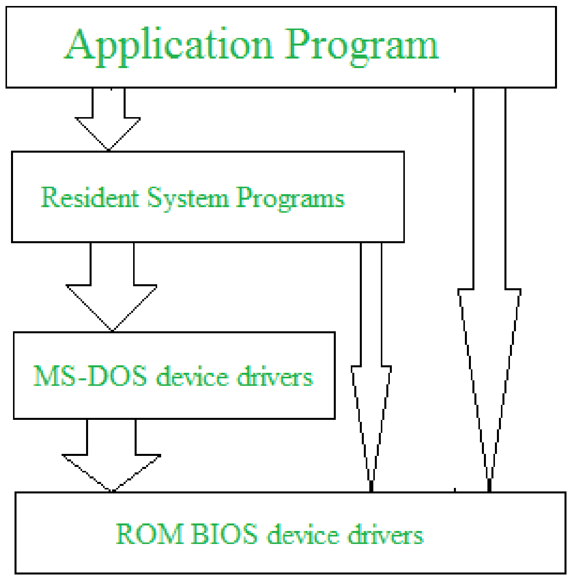
Types of Operating Systems Structures

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

- Simple/Monolithic Structure
- Micro-Kernel Structure
- Hybrid-Kernel Structure
- Exo-Kernel Structure
- Layered Structure
- Modular Structure
- Virtual Machines

Simple/Monolithic structure

Such 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/Monolithic 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/Monolithic Structure

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

Micro-Kernel Structure

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 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 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.

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.

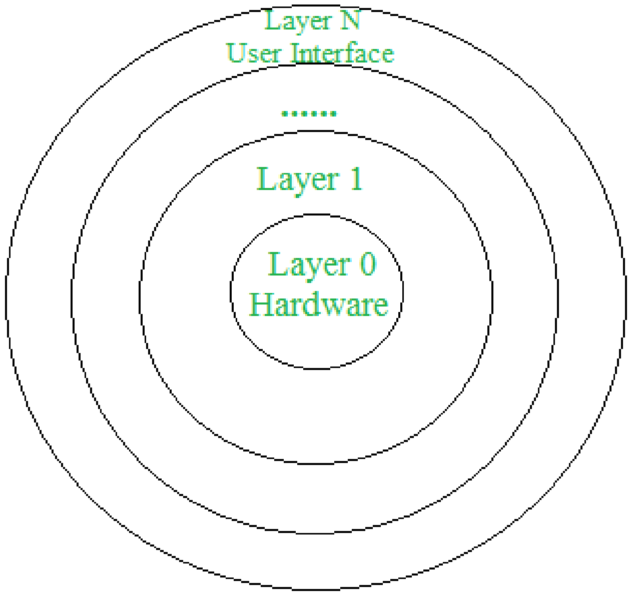
Disadvantages of Exo-Kernel

- 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

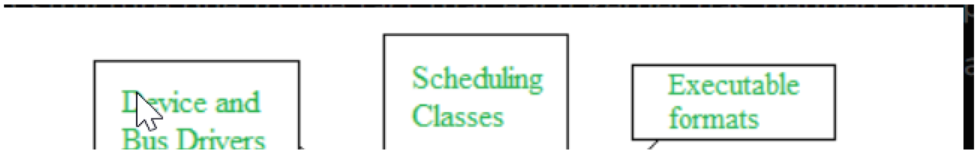
- Layering makes it easier to enhance the operating system, as the implementation of a layer can be changed easily without affecting the other layers.
- It is very easy to perform debugging and system verification.

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.

Modular Structure

It is considered as the best approach for an OS. It involves designing of 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 runtime 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 a layered structure as a module can call any other module. For example Solaris OS is organized as shown in the figure.



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.

Conclusion

Operating systems can be built using various structures, each affecting how components interact within the kernel. These structures include

Simple/Monolithic, Micro-Kernel, Hybrid-Kernel, Exo-Kernel, Layered, Modular, and Virtual Machines. The structure chosen influences the ease of use, modification, and overall efficiency. By breaking down the OS into manageable parts, developers can create more adaptable and robust systems. Understanding these structures helps in designing operating systems that meet specific needs while maintaining functionality and performance.

Frequently Asked Questions on Operating System Structures – FAQs

What is a Microkernel?

> *A microkernel is an operating system structure that aims to keep the kernel as small and lightweight as possible. It provides only essential services, such as process scheduling and inter-process communication, while moving most non-essential services, like device drivers, into user space.*

What is a Hybrid Kernel?

> *A hybrid kernel combines features of both monolithic and microkernels. It includes a small kernel that handles essential services and basic hardware interactions, while additional services, such as file systems and device drivers, run in kernel mode but outside the core kernel.*

What is a Virtualization-based Kernel?

A virtualization-based kernel, also known as a hypervisor, is an operating system structure that enables the execution of multiple operating systems concurrently on the same hardware. It provides virtualized environments for guest operating systems and manages their interactions with the underlying hardware.

mohi... 126

Previous Article
System Programs in Operating System

Next Article
History of Operating System

Similar Reads

Operating Systems | Input Output Systems | Question 5

Which of the following is major part of time taken when accessing data on the disk? (A) Settle time (B) Rotational latency (C) Seek time (D) Waiting...

1 min read

Structures of Directory in Operating System

A directory is a container that is used to contain folders and files. It organizes files and folders in a hierarchical manner. In other words,...

8 min read

Operating Systems - GATE CSE Previous Year Questions

Solving GATE Previous Year's Questions (PYQs) not only clears the concepts but also helps to gain flexibility, speed, accuracy, and...

4 min read

Operating Systems | Set 2

Following questions have been asked in GATE CS exam. 1. Consider a machine with 64 MB physical memory and a 32-bit virtual address space. ...

4 min read

Operating Systems | Set 4

Following questions have been asked in GATE CS exam. 1. Using a larger block size in a fixed block size file system leads to (GATE CS 2003) a)...

3 min read

View More Articles

Article Tags : Operating Systems

All Cheat Sheets

NLP Tutorial
Deep Learning Tutorial

Web Technologies

HTML
CSS
JavaScript
TypeScript
ReactJS
NextJS
Bootstrap
Web Design

Python Tutorial

Python Programming Examples
Python Projects
Python Tkinter
Web Scraping
OpenCV Tutorial
Python Interview Question
Django

Computer Science

Operating Systems
Computer Network
Database Management System
Software Engineering
Digital Logic Design
Engineering Maths
Software Development
Software Testing

DevOps

Git
Linux
AWS
Docker
Kubernetes
Azure
GCP
DevOps Roadmap

System Design

High Level Design
Low Level Design
UML Diagrams
Interview Guide
Design Patterns
OOAD
System Design Bootcamp
Interview Questions

Inteview Preparation

Competitive Programming
Top DS or Algo for CP
Company-Wise Recruitment Process
Company-Wise Preparation
Aptitude Preparation
Puzzles

School Subjects

Mathematics
Physics
Chemistry
Biology
Social Science
English Grammar
Commerce
World GK

GeeksforGeeks Videos

DSA
Python
Java
C++
Web Development
Data Science
CS Subjects