**OS STRUCTURE**

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.

1. Simple/Monolithic Structure
2. Micro-Kernel Structure
3. Hybrid-Kernel Structure
4. Exo-Kernel Structure
5. Layered Structure
6. Modular Structure
7. Virtual Machines

1.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.



**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 Micro-kernel structure

 It makes the operating system portable to various platforms.

 As microkernels are small so these can be tested effectively

## 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.

## 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.

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.

## **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](https://www.geeksforgeeks.org/introduction-to-unix-system/) is an example of this structure.

 

## Modular structure or approach

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.