

What is Kernel in Operating System and what are the various types of Kernel?

A Kernel is a computer program that is the heart and core of an Operating System. Since the Operating System has control over the system so, the Kernel also has control over everything in the system. It is the most important part of an Operating System. Whenever a system starts, the Kernel is the first program that is loaded after the bootloader because the Kernel has to handle the rest of the thing of the system for the Operating System. The Kernel remains in the memory until the Operating System is shut-down.

The Kernel is responsible for low-level tasks such as disk management, memory management, task management, etc. It provides an interface between the user and the hardware components of the system. When a process makes a request to the Kernel, then it is called System Call.

A Kernel is provided with a protected Kernel Space which is a separate area of memory and this area is not accessible by other application programs. So, the code of the Kernel is loaded into this protected Kernel Space. Apart from this, the memory used by other applications is called the User Space. As these are two different spaces in the memory, so communication between them is a bit slower.

Functions of a Kernel

Following are the functions of a Kernel:

- **Access Computer resource:** A Kernel can access various computer resources like the CPU, I/O devices and other resources. It acts as a bridge between the user and the resources of the system.

- **Resource Management:** It is the duty of a Kernel to share the resources between various process in such a way that there is uniform access to the resources by every process.
- **Memory Management:** Every process needs some memory space. So, memory must be allocated and deallocated for its execution. All these memory management is done by a Kernel.
- **Device Management:** The peripheral devices connected in the system are used by the processes. So, the allocation of these devices is managed by the Kernel.

Kernel Mode and User Mode

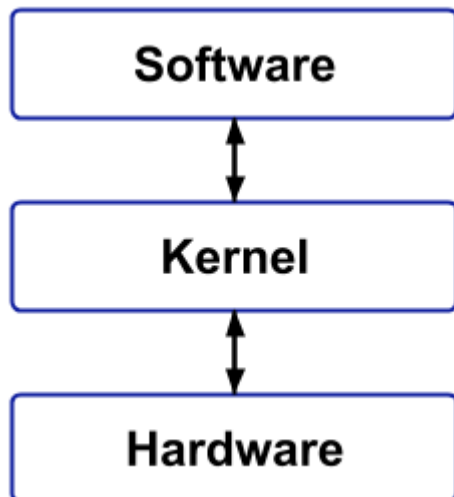
There are certain instructions that need to be executed by Kernel only. So, the CPU executes these instructions in the Kernel Mode only. For example, memory management should be done in Kernel-Mode only. While in the User Mode, the CPU executes the processes that are given by the user in the User Space.

Types of Kernel

In general, there are five types of Kernel. They are:

1. Monolithic Kernels

Monolithic Kernels are those Kernels where the user services and the kernel services are implemented in the same memory space i.e. different memory for user services and kernel services are not used in this case. By doing so, the size of the Kernel is increased and this, in turn, increases the size of the Operating System. As there is no separate User Space and Kernel Space, so the execution of the process will be faster in Monolithic Kernels.



Advantages:

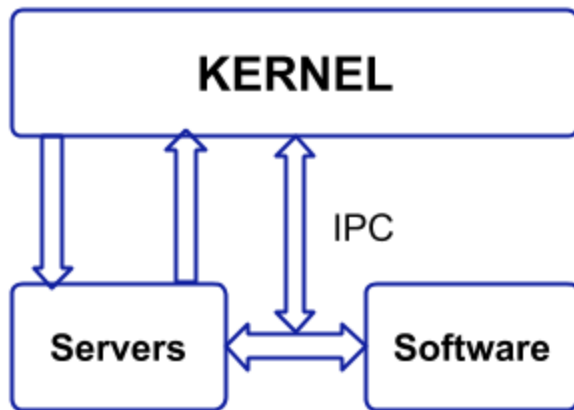
- It provides CPU scheduling, memory scheduling, file management through System calls only.
- Execution of the process is fast because there is no separate memory space for user and kernel.

Disadvantages:

- If any service fails, then it leads to system failure.
- If new services are to be added then the entire Operating System needs to be modified.

2. Microkernel

A Microkernel is different from Monolithic kernel because in a Microkernel, the user services and kernel services are implemented into different spaces i.e. we use User Space and Kernel Space in case of Microkernels. As we are using User Space and Kernel Space separately, so it reduces the size of the Kernel and this, in turn, reduces the size of Operating System.



As we are using different spaces for user services and kernel service, so the communication between application and services is done with the help of message parsing and this, in turn, reduces the speed of execution.

Advantages:

- If new services are to be added then it can be easily added.

Disadvantages:

- Since we are using User Space and Kernel Space separately, so the communication between these can reduce the overall execution time.

3. Hybrid Kernel

A Hybrid Kernel is a combination of both Monolithic Kernel and Microkernel. It makes the use of the speed of Monolithic Kernel and the modularity of Microkernel.

Hybrid kernels are micro kernels that have some "non-essential" code in kernel-space in order for the code to run more quickly than it would be in user-space. So, some services such as network stack or filesystem are run in Kernel space to reduce the performance overhead, but still, it runs kernel code as servers in the user-space.

4. Nanokernel

In a Nanokernel, as the name suggests, the whole code of the kernel is very small i.e. the code executing in the privileged mode of the hardware is very small. The term nanokernel is used to describe a kernel that supports a nanosecond clock resolution.

5. Exokernel

Exokernel is an Operating System kernel that is developed by the MIT parallel and the Distributed Operating Systems group. Here in this type of kernel, the resource protection is separated from the management and this, in turn, results in allowing us to perform application-specific customization.

In the Exokernel, the idea is not to implement all the abstractions. But the idea is to impose as few abstractions as possible and by doing so the abstraction should be used only when needed. So, no force abstraction will be there in Exokernel and this is the feature that makes it different from a Monolithic Kernel and Microkernel. But the drawback of this is the complex design. The design of the Exokernel is very complex.