

Microkernel in Operating System

In this article, you will learn about the microkernel with its architecture, advantages and disadvantages. But before discussing the microkernel, you must know about the kernel.

What is a kernel?

It is the central component of an OS that handles system resources. It also acts as a bridge between the computer's application and hardware. It is one of the initial programs that is loaded when the computer boots up. When an OS is loaded, the kernel is the first component that loads into memory and rests there until the OS is shut down. It is in charge of various activities, including task management, disk management, and memory management.

What is Microkernel?

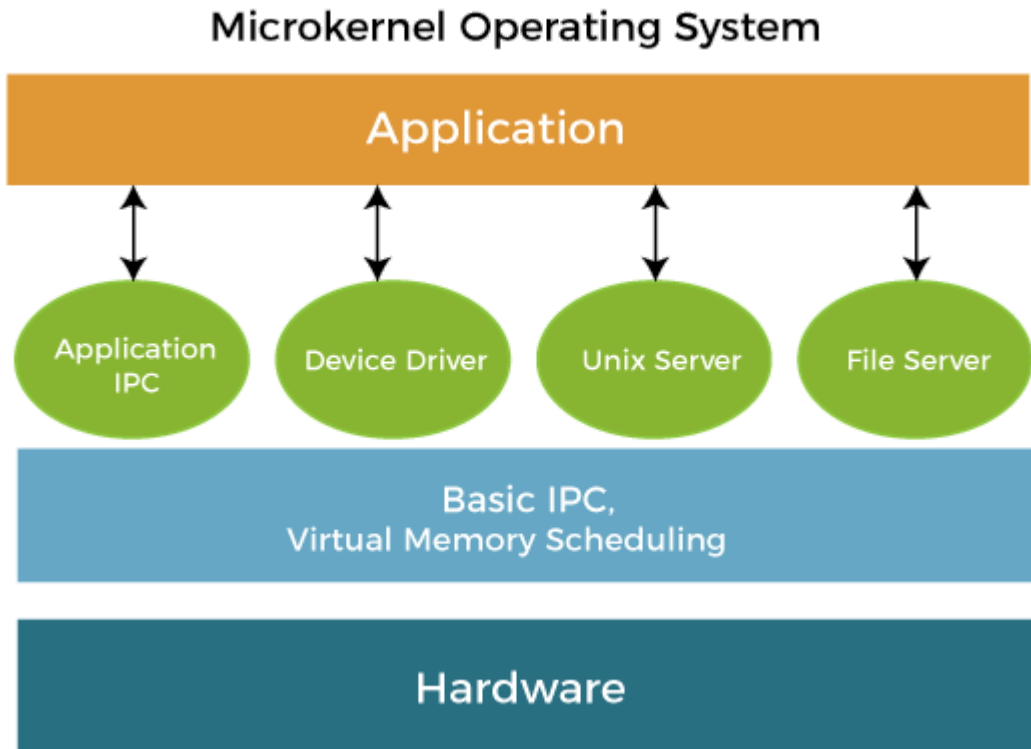
The microkernel is one of the kernel's classifications. Being a kernel, it handles all system resources. On the other hand, the user and kernel services in a microkernel are implemented in distinct address spaces. **User services** are kept in **user address space**, while **kernel services** are kept in **kernel address space**. It aids to reduce the kernel and OS's size.

It provides a minimal amount of process and memory management services. The interaction between the client application and services running in user address space is established via message passing that helps to reduce the speed of microkernel execution. The OS is unaffected because kernel and user services are isolated, so if any of the user services fails, the kernel service is unaffected. It is extendable because new services are added to the user address space, hence requiring no changes in kernel space. It's also lightweight, secure, and reliable.

Microkernels and their user environments are typically used in C++ or C languages with a little assembly. On the other hand, other implementation programming languages may be possible with some high-level code.

Architecture of Microkernel

A microkernel is a minimum needed of software required to implement an operating system correctly. Memory, process scheduling methods, and fundamental inter-process communication are all included.



In the above figure, the microkernel includes basic needs like process scheduling mechanisms, memory, and interprocess communication. It is the only program that executes at the privileged level, i.e., kernel mode. The OS's other functions are moved from the kernel-mode and execute in the user mode.

The microkernel ensures that the code may be easily controlled because the services are split in the user space. It means some code runs in the kernel mode, resulting in improved security and stability.

Since the kernel is the most crucial OS component, it is responsible for the essential services. As a result, under this design, only the most significant services are present inside the kernel in this architecture. In contrast, the rest operating system services are available inside the system application software. As a result, users can interact with such unnecessary services within the system application. The microkernel is entirely responsible for the operating system's most significant services, which are as follows:

1. **Inter-Process Communication**
2. **Memory Management**
3. **CPU Scheduling**

Inter-Process Communication

Interprocess communication refers to how processes interact with one another. A process has several threads. In the kernel space, threads of any process interact with one another. Messages are sent and received across threads using ports. At the kernel level, there are several ports like process port, exceptional port, bootstrap port, and registered port. All of these ports interact with user-space processes.

Memory Management

Memory management is the process of allocating space in main memory for processes. However, there is also the creation of virtual memory for processes. Virtual memory means that if a process has a bigger size than the main memory, it is partitioned into portions and stored. After that, one by one, every part of the process is stored in the main memory until the CPU executes it.

CPU Scheduling

CPU scheduling refers to which process the CPU will execute next. All processes are queued and executed one at a time. Every process has a level of priority, and the process with the highest priority is performed out first. CPU scheduling aids in optimizing CPU utilization. In addition, resources are being used more efficiently. It also minimizes the waiting time. Waiting time shows that a process takes less time in the queue and that resources are allocated to the process more quickly. CPU scheduling also reduces response and turnaround times.

Components of Microkernel

A microkernel contains only the system's basic functions. A component is only included in the microkernel if putting it outside would disrupt the system's operation. The user mode should be used for all other non-essential components. The minimum functionalities needed in the microkernel are as follows:

1. In the microkernel, processor scheduling algorithms are also required.
Process and thread schedulers are included.
2. Address spaces and other memory management mechanisms should be incorporated in the microkernel. Memory protection features are also included.
3. Inter-process communication (IPC) is used to manage servers that execute their own address spaces.

Advantages and Disadvantages of Microkernel

Various advantages and disadvantages of the microkernel are as follows:

Advantages

1. Microkernels are secure since only those parts are added, which might disturb the system's functionality.
2. Microkernels are modular, and the various modules may be swapped, reloaded, and modified without affecting the kernel.
3. Microkernel architecture is compact and isolated, so it may perform better.
4. The system expansion is more accessible, so it may be introduced to the system application without disrupting the kernel.
5. When compared to monolithic systems, microkernels have fewer system crashes. Furthermore, due to the modular structure of microkernels, any crashes that do occur are simply handled.
6. The microkernel interface helps in enforcing a more modular system structure.
7. Server failure is treated the same as any other user program failure.
8. It adds new features without recompiling.

Disadvantages

1. When the drivers are implemented as procedures, a context switch or a function call is needed.
2. In a microkernel system, providing services are more costly than in a traditional monolithic system.
3. The performance of a microkernel system might be indifferent and cause issues.