

Module - 2

History of Operating System

An operating system is a type of software that acts as an interface between the user and the hardware. It is responsible for handling various critical functions of the computer and utilizing resources very efficiently so the operating system is also known as a resource manager. The operating system also acts like a government because just as the government has authority over everything, similarly the operating system has authority over all resources. Various tasks that are handled by OS are file management, task management, garbage management, memory management, process management, disk management, I/O management, peripherals management, etc.

Function of Operating System

- Memory management
- Process management
- File management
- Device Management
- Deadlock Prevention
- Input Output device management
- Act as a resource manager
- Time Management

Generation of Operating System

1. The First Generation (1940 to early 1950s)

In 1940, an operating system was not included in the creation of the first electrical computer. Early computer users had complete control over the device and wrote programs in pure machine language for every task. During the computer generation, a programmer can merely execute and solve basic mathematical calculations. An operating system is not needed for these computations.

2. The Second Generation (1955 – 1965)

GMOSIS, the first operating system (OS) was developed in the early 1950s. For the IBM Computer, General Motors has created the operating system. Because it gathers all related jobs into groups or batches and then submits them to the operating system using a punch card to finish all of them, the second-generation operating system was built on a single- stream batch processing system.

3. The Third Generation (1965 – 1980)

- Because it gathers all similar jobs into groups or batches and then submits them to the second generation operating system using a punch card to finish all jobs in a machine, the second-generation operating system was based on a single stream [batch processing system](#).
- [Operating system](#) designers were able to create a new operating system in the late 1960s that was capable of [multiprogramming](#)—the simultaneous execution of several tasks in a single computer program.
- In order to create operating systems that enable a CPU to be active at all times by carrying out multiple jobs on a computer at once, multiprogramming has to be introduced. With the release of the DEC PDP-1 in 1961, the third generation of minicomputers saw a new phase of growth and development.

4. The Fourth Generation (1980 – Present Day)

- The fourth generation of personal computers is the result of these PDPs. The Generation IV (1980–Present) The evolution of the personal computer is linked to the fourth generation of operating systems. Nonetheless, the third-generation minicomputers and the personal computer have many similarities. At that time, minicomputers were only slightly more expensive than personal computers, which were highly expensive.
- The development of Microsoft and the Windows operating system was a significant influence in the creation of personal computers. In 1975, Microsoft developed the first Windows operating system. Bill Gates and Paul Allen had the idea to advance personal computers after releasing the Microsoft Windows OS. As a result, the [MS-DOS](#) was

released in 1981, but users found it extremely challenging to decipher its complex commands.

- Windows is now the most widely used and well-liked operating system available. Following then, Windows released a number of operating systems, including Windows 95, Windows 98, [Windows XP](#), and Windows 7, the most recent operating system. The majority of Windows users are currently running Windows 10. Apple is another well-known operating system in addition to Windows.

Types of Operating System

Operating Systems have evolved in past years. It went through several changes before getting its original form. These changes in the operating system are known as the **evolution of operating systems**. OS improve itself with the invention of new technology. Basically , OS added the feature of new technology and making itself more powerful. Let us see the evolution of operating system year-wise in detail:

- No OS – (0s to 1940s)
- Batch Processing Systems -(1940s to 1950s)
- Multiprogramming Systems -(1950s to 1960s)
- Time-Sharing Systems -(1960s to 1970s)
- Introduction of GUI -(1970s to 1980s)
- Networked Systems – (1980s to 1990s)
- Mobile Operating Systems – (Late 1990s to Early 2000s)
- AI Integration – (2010s to ongoing)

1. No OS – (0s to 1940s)

As we know that before 1940s, there was no use of OS . Earlier, people are lacking OS in their computer system so they had to manually type instructions for each tasks in machine language(0-1 based language) . And at that time , it was very hard for users to implement even a simple task. And it was very time consuming and also not user-friendly . Because not everyone had that much level of understanding to understand the machine language and it required a deep understanding.

2. Batch Processing Systems -(1940s to 1950s)

With the growth of time, batch processing system came into the market .Now Users had facility to write their programs on punch cards and load it to the computer operator. And then operator make different batches of similar types of jobs and then serve the different

batch(group of jobs) one by one to the CPU .CPU first executes jobs of one batch and then jump to the jobs of other batch in a sequence manner.

3. Multiprogramming Systems -(1950s to 1960s)

[Multiprogramming](#) was the first operating system where actual revolution began. It provide user facility to load the multiple program into the memory and provide a specific portion of memory to each program. When one program is waiting for any I/O operations (which take much time) at that time the OS give permission to CPU to switch from previous program to other program(which is first in ready queue) for continuous execution of program with [interrupt](#).

4. Time-Sharing Systems -(1960s to 1970s)

Time-sharing systems is extended version of multiprogramming system. Here one extra feature was added to avoid the use of CPU for long time by any single program and give access of CPU to every program after a certain interval of time. Basically OS switches from one program to another program after a certain interval of time so that every program can get access of CPU and complete their work.

5. Introduction of GUI -(1970s to 1980s)

With the growth of time, Graphical User Interfaces (GUIs) came. First time OS became more user-friendly and changed the way of people to interact with computer. GUI provides computer system visual elements which made user's interaction with computer more comfortable and user-friendly. User can just click on visual elements rather than typing commands. Here are some feature of GUI in Microsoft's windows icons, menus and windows.

6. Networked Systems – (1980s to 1990s)

At 1980s,the craze of computer networks at it's peak .A special type of Operating Systems needed to manage the network communication . The OS like Novell NetWare and Windows NT were developed to manage network communication which provide users facility to work in collaborative environment and made file sharing and remote access very easy.

7. Mobile Operating Systems – (Late 1990s to Early 2000s)

Invention of smartphones create a big revolution in software industry, To handle the operation of smartphones , a special type of operating systems were developed. Some of them are : iOS and Android etc. These operating systems were optimized with the time and became more powerful.

8. AI Integration – (2010s to ongoing)

With the growth of time, [Artificial intelligence](#) came into picture. Operating system integrates features of AI technology like Siri, Google Assistant, and Alexa and became more powerful and efficient in many ways. These AI features with operating system create a entire new feature like voice commands, predictive text, and personalized recommendations.

Note: The above mentioned OS basically tells how the OS evolved with the time by adding new features but it doesn't mean that only new generation OS are in use and previously OS system are not in use, according to the need, all these OS are still used in software industry.

Operating System - Architecture

An operating system allows the user application programs to interact with the system hardware. Since the operating system is such a complex structure, its architecture plays an important role in its usage. Each component of the Operating System Architecture should be well defined with clear inputs, outputs and functions.

Import Terms

In operating system Architecture, we've two major terms which defines the major components of the operating systems.

- **Kernal** – Kernal is the central component of an operating system architecture in most of the implementation. A kernal is responsible for all major operations and interaction with the hardware. A kernal manages memory, processor, input/output devices and provides interface to application programs to interact with hardware components.
- **Shell** – Shell is an interface of an operating system. It can be command line interface or a graphical user interface. User interacts with an operating system using shell. Application programs can also use shell interface to interact with underlying operating system.
- **System Softwares** – System softwares are the programs which interact with Kernal and provides interface for security managment, memory management and other low level activities.

- **Application Programs** – Application softwares/Programs are the one using which a user interacts with the operating system. For example a word processor to create a document and save it on the file system, a notepad to create notes etc.

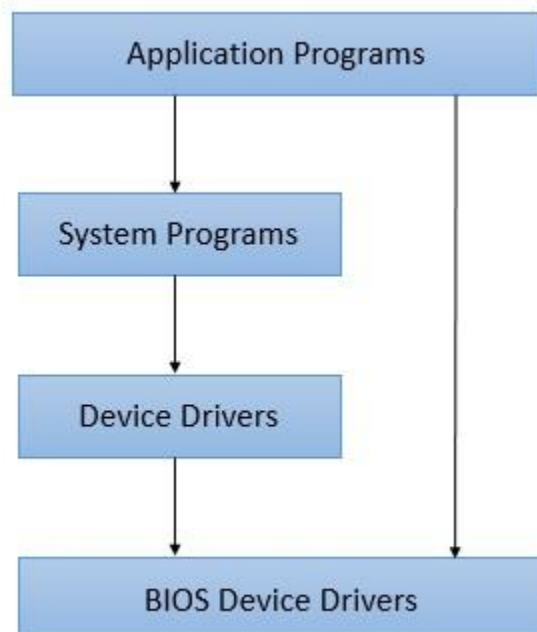
Popular Architectures

Following are various popular implementations of Operating System architectures.

- Simple Architecture
- Monolith Architecture
- Micro-Kernel Architecture
- Exo-Kernel Architecture
- Layered Architecture
- Modular Architecture
- Virtual Machine Architecture

Simple Architecture

There are many operating systems that have a rather simple structure. These started as small systems and rapidly expanded much further than their scope. A common example of this is MS-DOS. It was designed simply for a niche amount for people. There was no indication that it would become so popular.



Few operating systems have a simple yet powerful architecture, for example, MS-DOS. That would lead to greater control over the computer system and its various applications. The simple architecture allows the programmers to hide information as required and implement internal routines as they see fit without changing the outer specifications.

Advantages

Following are advantages of a simple operating system architecture.

- **Easy Development** - In simple operation system, being very few interfaces, development is easy especially when only limited functionalities are to be delivered.
- **Better Performance** - Such a system, as have few layers and directly interacts with hardware, can provide a better performance as compared to other types of operating systems.

Disadvantages

Following are disadvantages of a simple operating system architecture.

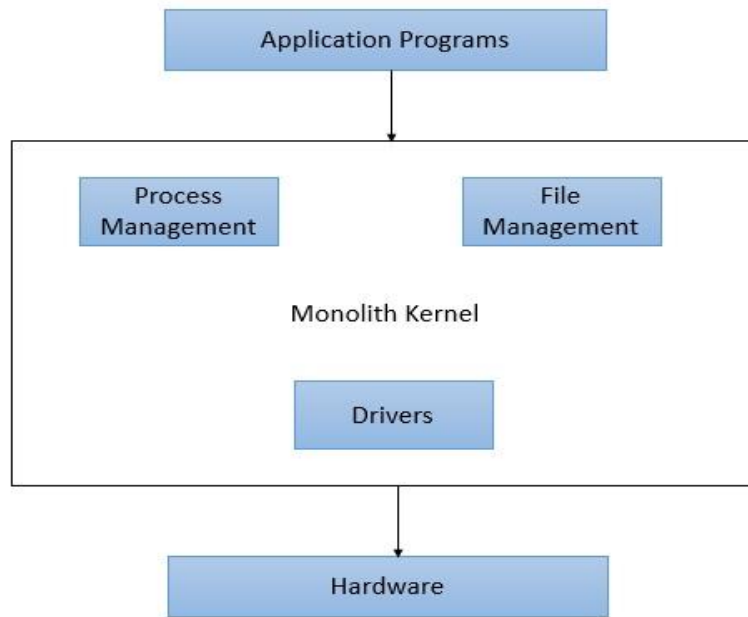
- **Frequent System Failures** - Being poorly designed, such a system is not robust. If one program fails, entire operating system crashes. Thus system failures are quite frequent in simple operating systems.
- **Poor Maintainability** - As all layers of operating systems are tightly coupled, change in one layer can impact other layers heavily and making code unmanageable over a period of time.



Monolith Architecture

In monolith architecture operating system, a central piece of code called kernel is responsible for all major operations of an operating system. Such operations include file management, memory management, device management and so on. The kernel is the main component of an operating system and it provides all the services of an operating system to the application programs and system programs.

The kernel has access to all the resources and it acts as an interface with application programs and the underlying hardware. A monolithic kernel architecture promotes timesharing, multiprogramming model and was used in old banking systems.



Advantages

Following are advantages of a monolith operating system architecture.

- **Easy Development** - As kernel is the only layer to develop with all major functionalities, it is easier to design and develop.
- **Performance** - As Kernel is responsible for memory management, other operations and have direct access to the hardware, it performs better.

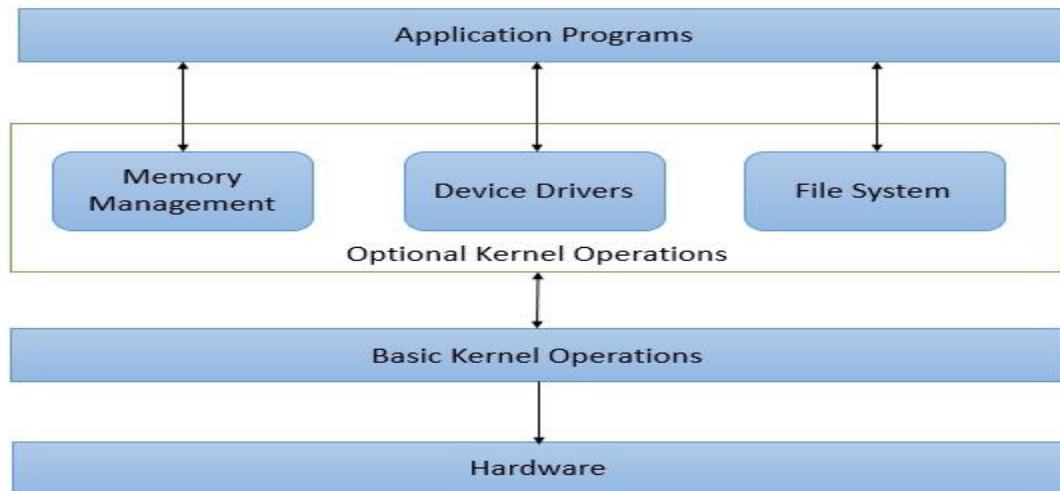
Disadvantages

Following are disadvantages of a monolith operating system architecture.

- **Crash Prone** - As Kernel is responsible for all functions, if one function fails entire operating system fails.
- **Difficult to enhance** - It is very difficult to add a new service without impacting other services of a monolith operating system.

Micro-Kernel Architecture

As in case monolith architecture, there was single kernel, in micro-kernel, we have multiple kernels each one specilized in particular service. Each microkernel is developed independent to the other one and makes system more stable. If one kernel fails the operating sytem will keep working with other kernel's functionalities.



Advantages

Following are advantages of a microkernel operating system architecture.

- **Reliable and Stable** - As multiple kernels are working simultaneously, chances of failure of operating system is very less. If one functionality is down, operating system can still provide other functionalities using stable kernels.
- **Maintainability** - Being small sized kernels, code size is maintainable. One can enhance a microkernel code base without impacting other microkernel code base.

Disadvantages

Following are disadvantages of a microkernel operating system architecture.

- **Complex to Design** - Such a microkernel based architecture is difficult to design.
- **Performance Degradation** - Multi kernel, Multi-modular communication may hamper the performance as compared to monolith architecture.



Exo-Kernel Architecture

Exo-Kernal Architecture operating system was designed and developed at MIT. The aim of this design was to keep Kernel size minimal while allowing the application programs to manage hardware resources directly. The purpose of removing abstraction of operating system for hardware resources was to enable application programmer to write high performance code while exo-kernel handles other operations.

Advantages

Following are advantages of a exo-kernel operating system architecture.

- **High Performance** - As application program can allocate memory, a better designed code can make optimal use and perform better.
- **Application Control** - As resource management is not secured by operating system, application program has more control over system resources and can write custom operations on system resources.

Disadvantages

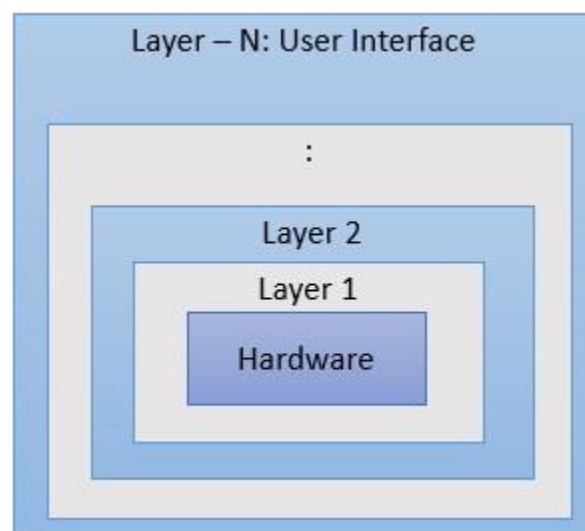
Following are disadvantages of a exo-kernel operating system architecture.

- **Unreliable and Unsafe** - As security is in application program level, a poorly written code can ruin the system.
- **Complex Design** - Exo-Kernel designing is complicated.

Layered Architecture

One way to achieve modularity in the operating system is the layered approach. In this, the bottom layer is the hardware and the topmost layer is the user interface.

An image demonstrating the layered approach is as follows –



As seen from the image, each upper layer is built on the bottom layer. All the layers hide some structures, operations etc from their upper layers.

One problem with the layered architecture is that each layer needs to be carefully defined. This is necessary because the upper layers can only use the functionalities of the layers below them.

Advantages

Following are advantages of a layered operating system architecture.

- **High Customizable** - Being layered, each layer implementation can be customized easily. A new functionality can be added without impacting other modules as well.
- **Verifiable** - Being modular, each layer can be verified and debugged easily.

Disadvantages

Following are disadvantages of a layered operating system Architecture.

- **Less Performant** - A layered structured operating system is less performant as compared to basic structured operating system.
- **Complex designing** - Each layer is to planned carefully as each layer communicates with lower layer only and a good design process is required to create a layered operating system.

Modular Architecture

Modular architecture operating system works on the similar principle as a monolith but with better design. A central kernel is responsible for all major operations of operating system. This kernel has set of core functionality and other services are loaded as modules dynamically to the kernel at boot time or at runtime. Sun Solaris OS is one of the example of Modular structured operating system.

Advantages

Following are advantages of a modular operating system architecture.

- **High Customizable** - Being modular, each module implementation can be customized easily. A new functionality can be added without impacting other modules as well.

- **Verifiable** - Being modular, each layer can be verified and debugged easily.

Disadvantages

Following are disadvantages of a modular operating system architecture.

- **Less Performant** - A modular architecture operating system is less performant as compared to basic structured operating system.
- **Complex designing** - Each module is to planned carefully as each module communicates with kernal. A communication API is to be devised to facilitate the communication.

Virtual Machine Architecture

In this kind of architecture, hardware like CPU, memory, hard disks are abstracted into virtual machines. User can use them with actually configure them using execution contexts. Virtual machine takes a good amount of disk space and is to be provisioned. Multiple virtual machines can be created on a single physical machine.

Advantages

Following are advantages of a virtual machine based operating system architecture.

- **High Customizable** - Being virtual, functionality are easily accessible, can be customized on need basis.
- **Secure** - Being virtual, and no direct hardware access, such systems are highly secured.

Disadvantages

Following are disadvantages of a virtual machine based operating system architecture.

- **Less Performant** - A virtual structured operating system is less performant as compared to modular structured operating system.
- **Complex designing** - Each virtual component of the machine is to planned carefully as each component is to abstract underlying hardware.