Operating system

Lecture 1. Introduction to Operating system. OS architecture

Lecturer: Aidana Karibayeva

What is an operating system?

- An **operating system (OS)** is the interface between the user and the hardware

- OS provides standard services (an interface) which are implemented on the hardware, including:
  - Processes, CPU scheduling, memory management, file system, networking

  The OS coordinates multiple applications and users (multiple processes) in a fair and efficient manner

  The goal in OS development is to make the machine convenient to use (a software engineering problem) and efficient (a system and engineering problem)

OS – a software that controls and coordinates the computer hardware devices and runs other software and applications on a computer. It is the main part of system software and a computer will not function without it.

•Main functions of an operating system – booting the computer, managing system resources (CPU, memory, storage devices, printer, etc.), managing files, handling input and output, executing and providing services for application software, etc.

•Examples of operating system: Microsoft Windows, Apple iOS, Android OS, macOS, Linux, etc.

In brief, an operating system is the set of programs that controls a computer. Some examples of operating systems are UNIX, Mach, MS-DOS, MS Windows, Windows/NT, OS/2 and MacOS. • An operating system is an important part of almost every computer system. •

A computer system can be divided roughly into four components: the hardware, the operating system, the application programs and the users

History of OS

The 1940's - First Generations The earliest electronic digital computers had no operating systems. Machines of the time were so primitive that programs were often entered one bit at time on rows of mechanical switches (plug boards). Programming languages were unknown (not even assembly languages). Operating systems were unheard of.

The 1950's - Second Generation By the early 1950's, the routine had improved somewhat with the introduction of punch cards. The General Motors Research Laboratories implemented the first operating systems in early 1950's for their IBM 701. The system of the 50's generally ran one job at a time.

The 1960's - Third Generation The systems of the 1960's were also batch processing systems, but they were able to take better advantage of the computer's resources by running several jobs at once.

Fourth Generation With the development of LSI (Large Scale Integration) circuits, chips, operating system entered in the personal computer and the workstation age. Microprocessor technology evolved to the point that it becomes possible to build desktop computers as powerful as the mainframes of the 1970s.

Operating System Views

- Resource allocator
  - to allocate resources (software and hardware) of the computer system and manage them efficiently.
- Control program
  - Controls execution of user programs and operation of I/O devices.
- Kernel
  - The program that executes forever (everything else is an application with respect to the kernel).

Kernel Data Structures

The operating system must keep a lot of information about the current state of the system. As things happen within the system these data structures must be changed to reflect the current reality. For example, a new process might be created when a user logs onto the system. The kernel must create a data structure representing the new process and link it with the data structures representing all of the other processes in the system. Mostly these data structures exist in physical memory and are accessible only by the kernel and its subsystems. Data structures contain data and pointers, addresses of other data structures, or the addresses of routines.

Modern OS functionality

- Concurrency
  - Multiple processes active at once
  - Processes can communicate
  - Processes may require mutually exclusive access to some resource
  - CPU scheduling, resource management
- Memory management — allocate memory to processes, move processes between disk and memory
- File system — allocate space for storage of programs and data on disk
- Networks and distributed computing — allow computers to work together
- Security & protection

OS manages many kinds of activities:

- User programs
- System programs: printer spoolers, name servers, file servers, etc.

Each is encapsulated in a process

- A process includes the complete execution context (code, data, PC, registers, OS resources in use, etc.)
- A process is not a program

A process is one instance of a program in execution; many processes can be running the same program

Functions of OS

OS must:

- Create, delete, suspend, resume, and schedule processes
- Support inter-process communication and synchronization, handle deadlock
- Mechanics
    - Keep track of memory in use
    - Keep track of unused (“free”) memory
    - Protect memory space
    - Allocate, deallocate space for processes
    - Swap processes: memory <–> disk
- Policies
    - Decide when to load each process into memory
    - Decide how much memory space to allocate each process
    - Decide when a process should be removed from memory

Types of OS

Parallel operating systems

- Shared memory, shared clock
- Large number of tightly-coupled processors
- Appearance of single operating system

Distributed operating systems

- No shared memory, no shared clock
- Small number of loosely-coupled processors
- Appearance of single operating system is ideal goal, but not realized in practice
- May try to simulate a shared memory

Real-time operating systems

- Meet hard / soft real-time constraints on processing of data

File System Management

File System

- Disks (secondary storage) provide long-term storage, but are awkward to use directly
- File system provides files and various operations on files
  - A file is a long-term storage entity, a named collection of persistent information that can be read or written

A file system supports directories, which contain files and other directories – Name, size, date created, date last modified, owner, etc. n

OS must:

- Create and delete files and directories
- Manipulate files and directories
  - Read, write, extend, rename, copy, protect
- Provide general higher-level services

Backups, accounting, quotas

Operating systems everywhere









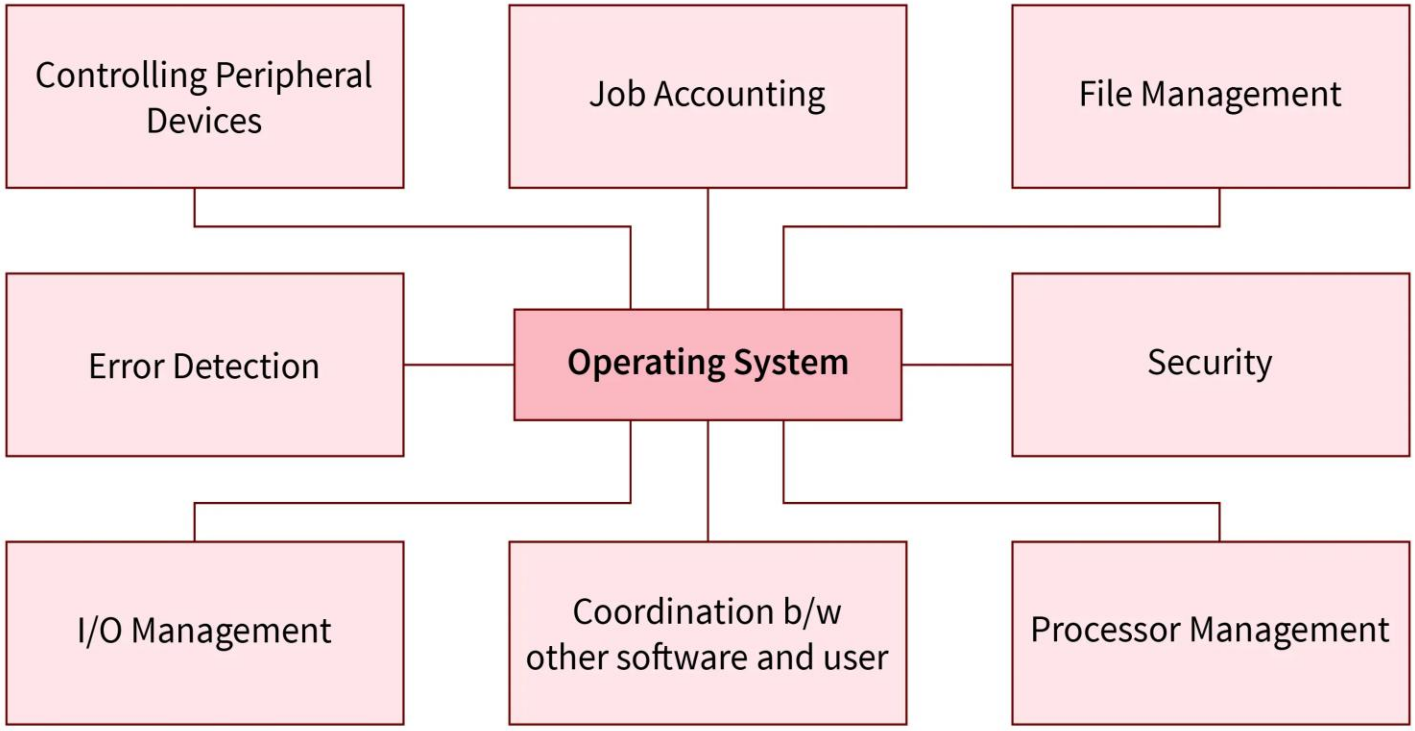


Terms of OS

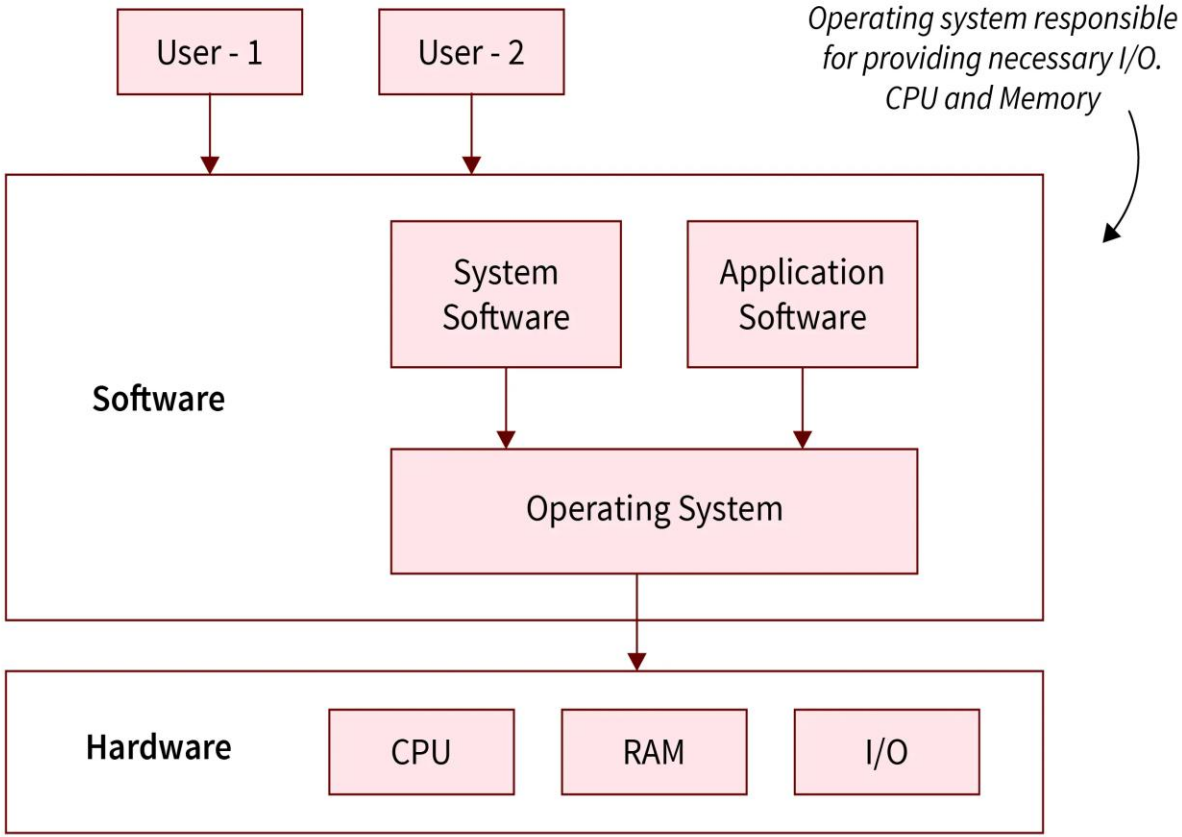
- The operating system gives an environment for executing programs by the users.
- **Kernel** is the most central part of the operating systems.
- **Software** running on any operating system can be system software and application software. The operating system as we know is an intermediary and its functionalities include file management, memory management, process management, handling input and output, and peripheral devices as well.

The architecture of an operating system

The architecture of an operating system consists of four major components hardware, kernel, shell, and application and we shall explore all of them in detail one by one.



SCALER Topics


User - 1
User - 2
Operating system responsible for providing necessary I/O. CPU and Memory
Software
System Software
Application Software
Operating System
Hardware
CPU
RAM
I/O

•Before explaining various architectures of the operating systems, let's explore a few terms first which are part of the operating system.

1) Application: The application represents the software that a user is running on an operating system it can be either system or application software eg slack, sublime text editor, etc.

2) Shell: The shell represents software that provides an interface for the user where it serves to launch or start some program for which the user gives instructions.

It can be of two types first is a command line and another is a graphical user interface for eg: MS-DOS Shell, PowerShell, csh, ksh, etc.

3) Kernel: Kernel represents the most central and crucial part of the operating system where it is used for resource management i.e. it provides necessary I/O, processor, and memory to the application processes through inter-process communication mechanisms and system calls. Let's understand the various types of architectures of the operating system.

Which among the given part of an OS provides an interface for the user where it serves to launch or start some program ?

A. Application
B. Shell
C. Kernel
D. None of the given

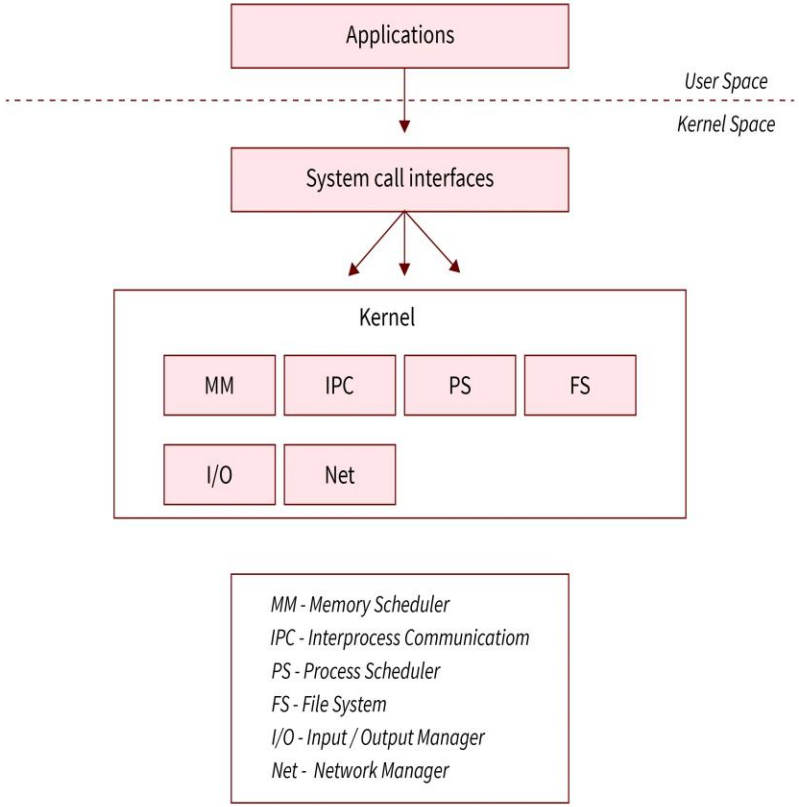
Types of Architectures of Operating System

•Architectures of operating systems can be of four types **monolithic, layered, microkernel, and hybrid.**

•Hybrid architecture is the combination of all architectures. There are major four major types of architectures of operating systems.

1) Monolithic Architecture

In monolithic architecture, each component of the operating system is contained in the kernel i.e. it is working in kernel space, and the components of the operating system communicate with each other using function calls.
Examples of this type of architecture are OS/360, VMX, and LINUX.



Monolithic Architecture

Advantages:

1.The main advantage of having a monolithic architecture of the operating system is that it provides CPU scheduling, memory management, memory management, etc through system calls.

2.In a single address space, the entire large process is running.

3.It is a single static binary file.

Disadvantages:

1.The main disadvantage is that all components are interdependent and when one of them fails the entire system fails.

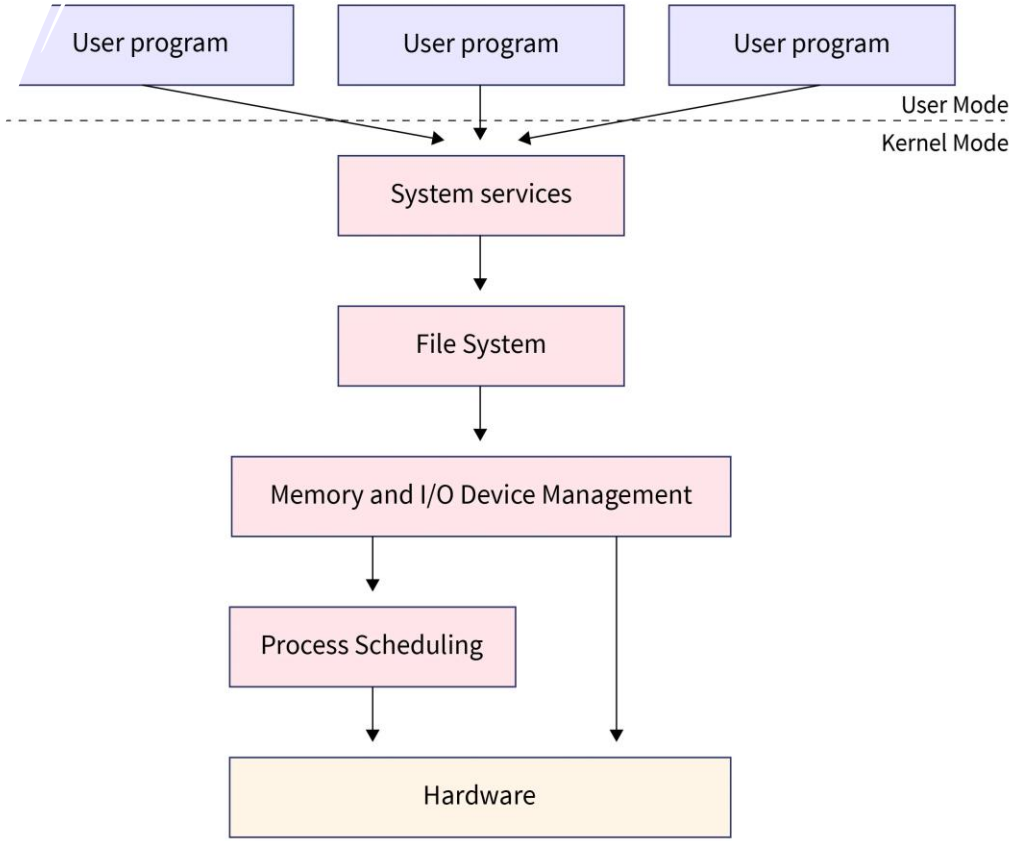
2.In case the user has to add a new service or functionality the entire operating system needs to be changed.

2) Layered architecture

In Layered architecture, components with similar functionalities are grouped to form a layer and in this way, total n+1 layers are constructed and counted from **0** to n where each layer has a different set of functionalities and services. Example: THE operating system, also Windows XP, and LINUX implements some level of layering.
The layers are implemented according to the following rule:
1.Each layer can communicate with all of its lower layers but not with its upper layer i.e. any **ith** layer can communicate with all layers from 0 to i-1 but not with the i+1th layer.
2.Each layer is designed in such a way that it will only need the functionalities that are present in itself or the layers below it.

Layered architecture

There are 6 layers in layered architecture as shown below:



SCALER Topics

Layered architecture

Let's explain the layers one by one

1) Hardware: This layer is the lowest layer in the layered operating system architecture, this layer is responsible for the coordination with peripheral devices such as keyboards, mice, scanners etc.

2) CPU scheduling: This layer is responsible for process scheduling, multiple queues are used for scheduling. Process entering the system are kept in the job queue while those which are ready to be executed are put into the ready queue. It manages the processes which are to be kept in the CPU and those which are to be kept out of the CPU.

3) Memory Management: This layer handles the aspect of memory management i.e. moving the processes from the secondary to primary memory for execution and vice-versa. There are memories like RAM and ROM. RAM is the memory where our processes run they are moved to the RAM for execution and when they exit they are removed from RAM.

Layered architecture

4) Process Management: This layer is responsible for managing the various processes i.e. assigning the CPU to those processes on a priority basis for their execution. Process management uses many scheduling algorithms for prioritizing the processes for execution such as the Round-Robin algorithm, FCFS(First Come First Serve), SJF(Shortest Job First), etc.

5) I/O Buffer: Buffering is the temporary storage of data and I/O Buffer means that the data input is first buffered before storing it in the secondary memory. All I/O devices have buffers attached to them for the temporary storage of the input data because it cannot be stored directly in the secondary storage as the speed of the I/O devices is slow as compared to the processor.

6) User Programs: This is the application layer of the layered architecture of the operating system, it deals with all the application programs running eg games, browsers, words, etc. It is the highest layer of layered architecture.

Layered architecture

Advantages:

1) Layered architecture of the operating system provides modularity because each layer is programmed to perform its own tasks only. 2) Since the layered architecture has independent components changing or updating one of them will not affect the other component or the entire operating system will not stop working, hence it is easy to debug and update. 3) The user can access the services of the hardware layer but cannot access the hardware layer itself because it is the innermost layer. 4) Each layer has its own functionalities and it is concerned with itself only and other layers are abstracted from it.

Disadvantages:

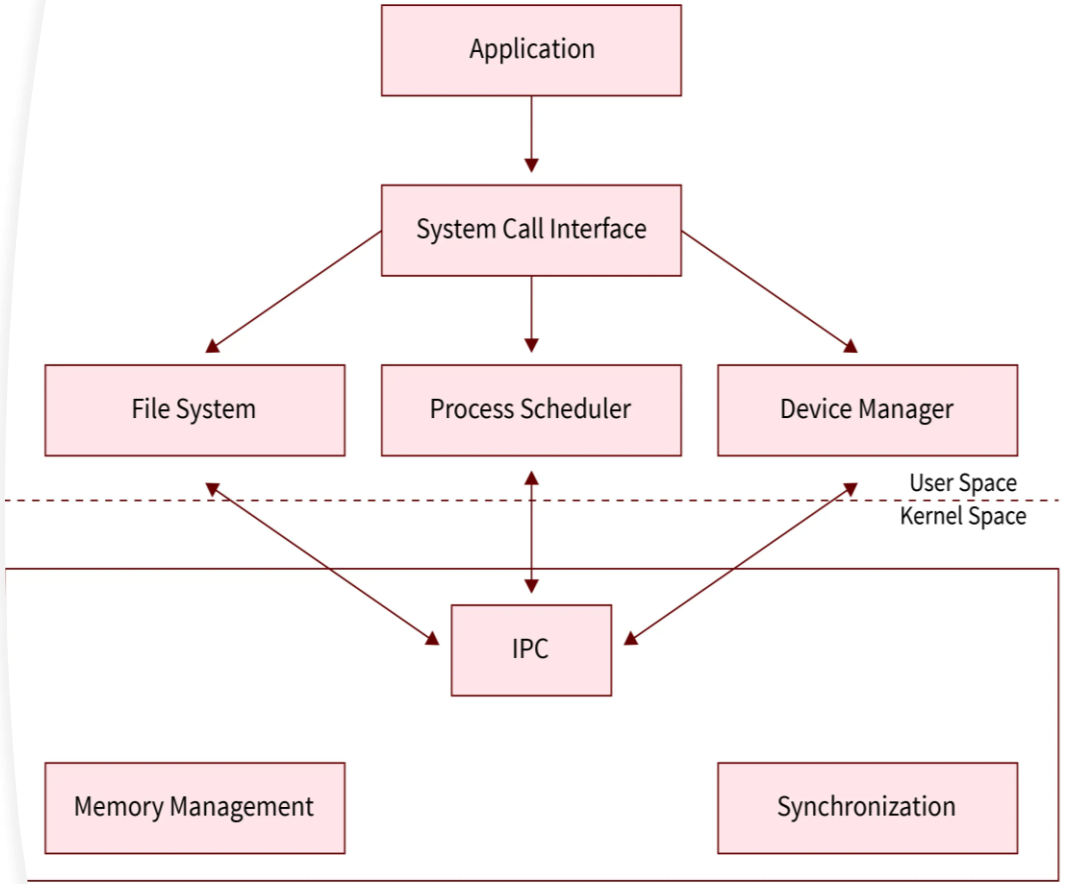
1.Layered architecture is complex in implementation because one layer may use the services of the other layer and therefore, the layer using the services of another layer must be put below the other one.

2.In a layered architecture, if one layer wants to communicate with another it has to send a request that goes through all layers in between which increases response time causing inefficiency in the system.

3) Microkernel Architecture

In this architecture, the components like process management, networking, file system interaction, and device management are executed outside the kernel while memory management and synchronization are executed inside the kernel. The processes inside the kernel have relatively high priority, the components possess high modularity hence even if one or more components fail the operating system keeps on working.

Example: Linux and Windows XP contain Modular components.



SCALER

Microkernel Architecture

Advantages:

1.Microkernel operating systems are modular and hence, disturbing one of the components will not affect the other component.

2.The architecture is compact and isolated and hence relatively efficient.

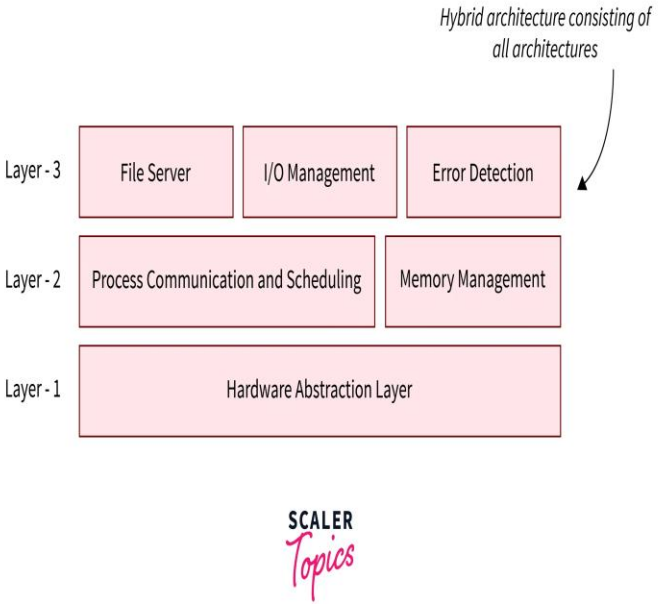
3.New features can be added without recompilation.

Disadvantages:

1.Implementing drivers as procedures requires a function call or context switch.

2.In microkernel architecture, providing services is costlier than monolithic operating systems.

4) Hybrid Architecture



Hybrid architecture as the name suggests consists of a hybrid of all the architectures explained so far and hence it has properties of all of those architectures which makes it highly useful in present-day operating systems.

The hybrid-architecture consists of three layers

1) Hardware abstraction layer: It is the interface between the kernel and hardware and is present at the lowest level.

2) Microkernel Layer: This is the old microkernel that we know and it consists of CPU scheduling, memory management, and inter-process communication.

3) Application Layer: It acts as an interface between the user and the microkernel. It contains functionalities like a file server, error detection, I/O device management, etc.

Example: Microsoft Windows NT kernel implements a hybrid architecture of the operating system.

Hybrid Architecture

Advantages:

1.Since it is a hybrid of other architectures it allows various architectures to provide their services respectively.

2.It is easy to manage because it uses a layered approach.

3.The number of layers is relatively lesser.

4.Security and protection are relatively improved.

Disadvantage:

1)Hybrid architecture of the operating system keeps certain services in the kernel space while moving less critical services to the user space.

In which among the given architecture executes different components of an operating system seperately?

A. Monolithic Architecture
B. Layered Architecture
C. Microkernel Architecture

Conclusion

•We conclude that the operating system has various architectures with which we can describe the functionality of various components.
•The components of the operating system are process management, memory management, I/O management, Error Detection, and controlling peripheral devices.
•These architectures include monolithic, layered, microkernel, and hybrid architectures classified on the basis of the structure of components.
•Hybrid architecture is the most efficient and useful architecture as it implements the functionalities of all other architectures.
•Hybrid architecture is better in terms of security as well.

Thanks for your attention!