BCA
Fourth Semester
" Operating System"

Prepared by :Narayan Dhamala

Unit II
Introduction to Operating System

operating system structure:

An operating system might have many structure. According to the structure of the operating system; operating systems can be classified into many categories.
Some of the main structures used in operating systems are:

1. **Monolithic architecture of operating system**

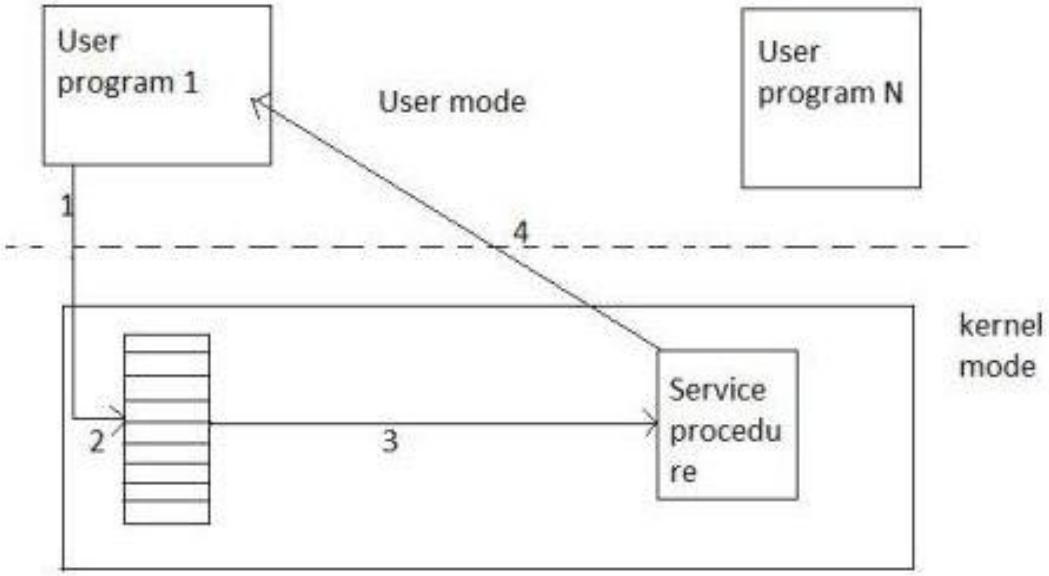


fig:- monolithic structure of os

Unit II

monolithic sturucture of operating system

It is the oldest architecture used for developing operating system. Operating system resides on kernel for anyone to execute. System call is involved i.e. Switching from user mode to kernel mode and transfer control to operating system shown as event 1. Many CPU has two modes, kernel mode, for the operating system in which all instruction are allowed and user mode for user program in which I/O devices and certain other instruction are not allowed. Two operating system then examines the parameter of the call to determine which system call is to be carried out shown in event 2. Next, the operating system index's into a table that contains procedure that carries out system call. This operation is shown in events. Finally, it is called when the work has been completed and the system call is finished, control is given back to the user mode as shown in event 4.

2. **Layered Architecture of operating system**

The layered Architecture of operating system was developed in 60's in this approach; the operating system is broken up into number of layers. The bottom layer (layer 0) is the hardware layer and the highest layer (layer n) is the user interface layer as shown in the figure.

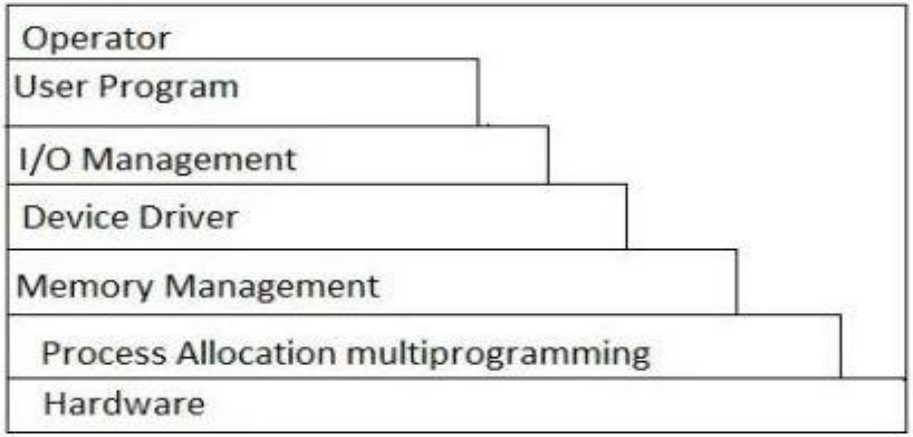


fig:- layered Architecture

layered architecture

The layered are selected such that each user functions and services of only lower level layer. The first layer can be debugged wit out any concern for the rest of the system. It user basic hardware to implement this function once the first layer is debugged., it's correct functioning can be assumed while the second layer is debugged & soon . If an error is found during the debugged of particular layer, the layer must be on that layer, because the layer below it already debugged. Because of this design of the system is simplified when operating system is broken up into layer.

Os/2 operating system is example of layered architecture of operating system another example is earlier version of Windows NT.

The main disadvantage of this architecture is that it requires an appropriate definition of the various layers & a careful planning of the proper placement of the layer.

3. Virtual memory architecture of operating system

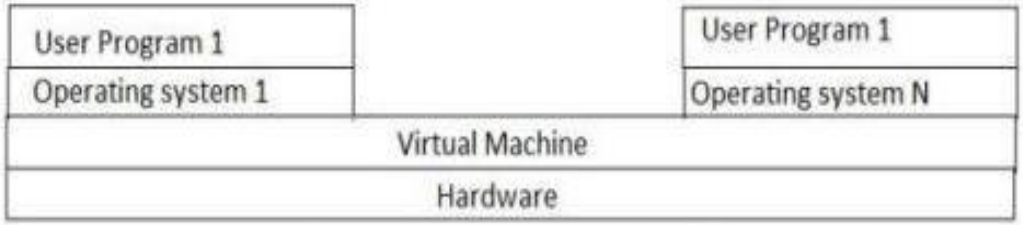


fig:- virtual memory architecture of os

virtual memory architecture

Virtual machine is an illusion of a real machine. It is created by a real machine operating system, which make a single real machine appears to be several real machine. The architecture of virtual machine is shown above. The best example of virtual machine architecture is IBM 370 computer. In this system each user can choose a different operating system. Actually, virtual machine can run several operating systems at once, each of them on its virtual machine.

Its multiprogramming shares the resource of a single machine in different manner.

The concepts of virtual machine are:-

a. Control program (cp):- cp creates the environment in which virtual machine can execute. It gives to each user facilities of real machine such as processor, storage I/0 devices.

B. conversation monitor system (cons):- cons is a system application having features of developing program. It contains editor, language translator, and various application packages.

c. Remote spooling communication system (RSCS):- provide virtual machine with the ability to transmit and receive file in distributed system.

d. IPCS (interactive problem control system):- it is used to fix the virtual machine software problems.

4. client/server architecture of operating system

A trend in modern operating system is to move maximum code into the higher level and remove as much as possible from operating system, minimising the work of the kernel. The basic approach is to implement most of the operating system functions in user processes to request a service, such as request to read a particular file, user send a request to the server process, server checks the parameter and finds whether it is valid or not, after that server does the work and send back the answer to client server model works on request- response technique i.e. Client always send request to the side in order to perform the task, and on the other side, server gates complementing that request send back response. The figure below shows client server architecture.

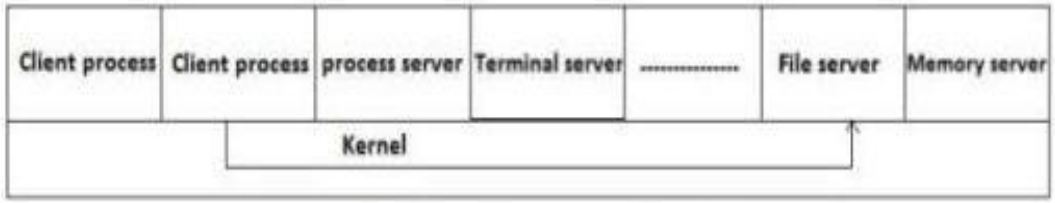


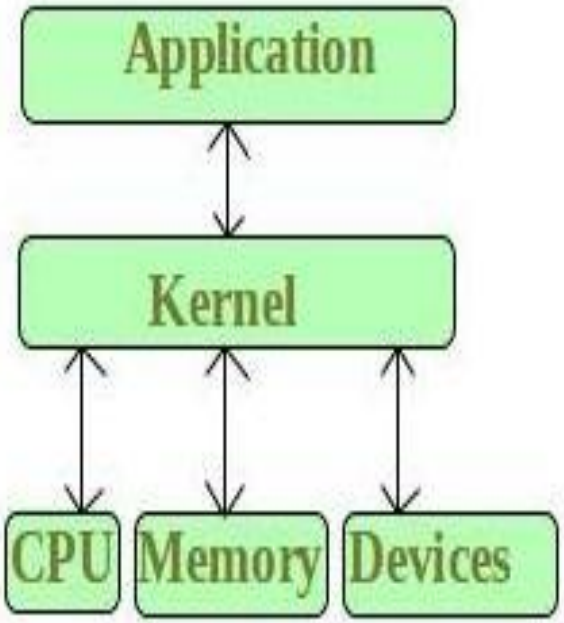
fig:- The client- server model

client server model

In this model, the main task of the kernel is to handle all the communication between the client and the server by splitting the operating system into number of ports, each of which only handle some specific task. I.e. file server, process server, terminal server and memory service.

Another advantage of the client-server model is it's adaptability to user in distributed system. If the client communicates with the server by sending it the message, the client need not know whether it was send a ..... Is the network to a server on a remote machine? As in case of client, same thing happen and occurs in client side that is a request was send and a reply come back.

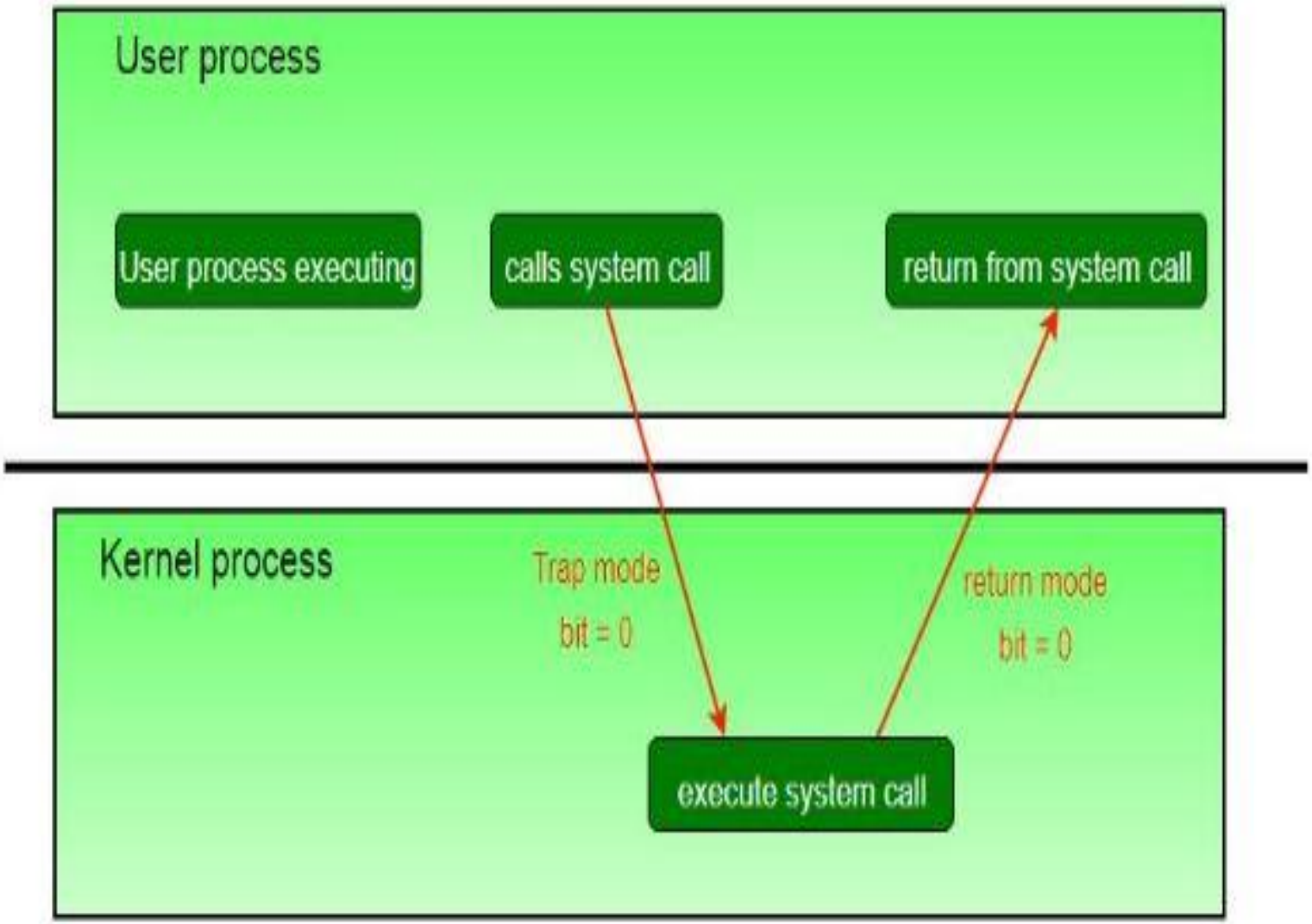
Kernel is the core part of an operating system which manages system resources. It also acts like a bridge between application and hardware of the computer. It is one of the first programs loaded on start-up (after the Bootloader).



Kernel Mode and User Mode of CPU Operation

The CPU can execute certain instruction only when it is in the kernel mode. These instruction are called privilege instruction. They allow implementation of special operation whose execution by the user program could interface with the functioning of operating system or activity of another user program. For example, instruction for managing memory protection.

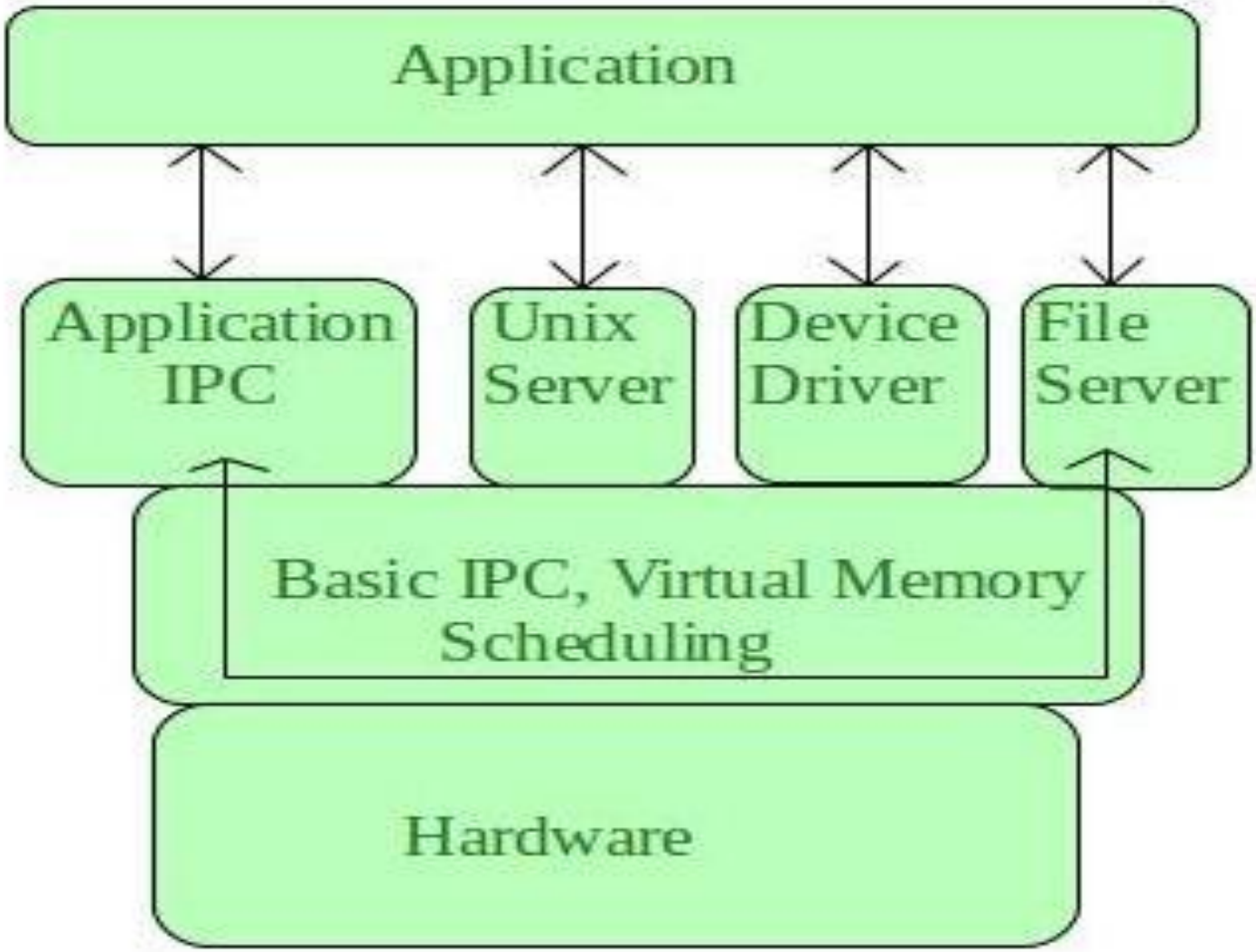
- The operating system puts the CPU in kernel mode when it is executing in the kernel so, that kernel can execute some special operation.
- The operating system puts the CPU in user mode when a user program is in execution so, that user program cannot interface with the operating system program.
- User-level instruction does not require special privilege. Example are ADD,PUSH,etc.


User process
User process executing
calls system call
return from system call
Kernel process
Trap mode
bit = 0
return mode
bit = 0
execute system call

Types of Kernel : (a) Micro kernel

What is Microkernel?

Microkernel is one of the classification of the kernel. Being a kernel it manages all system resources. But in a microkernel, the **user services** and **kernel services** are implemented in different address space. The user services are kept in **user address space**, and kernel services are kept under **kernel address space**, thus also reduces the size of kernel and size of operating system as well.


Application
Application IPC
Unix Server
Device Driver
File Server
Basic IPC, Virtual Memory Scheduling
Hardware

It provides minimal services of process and memory management. The communication between client program/application and services running in user address space is established through message passing, reducing the speed of execution microkernel. The Operating System **remains unaffected** as user services and kernel services are isolated so if any user service fails it does not affect kernel service. Thus it adds to one of the advantages in a microkernel. It is easily **extendable** i.e. if any new services are to be added they are added to user address space and hence requires no modification in kernel space. It is also portable, secure and reliable.

Advantages of Microkernel –

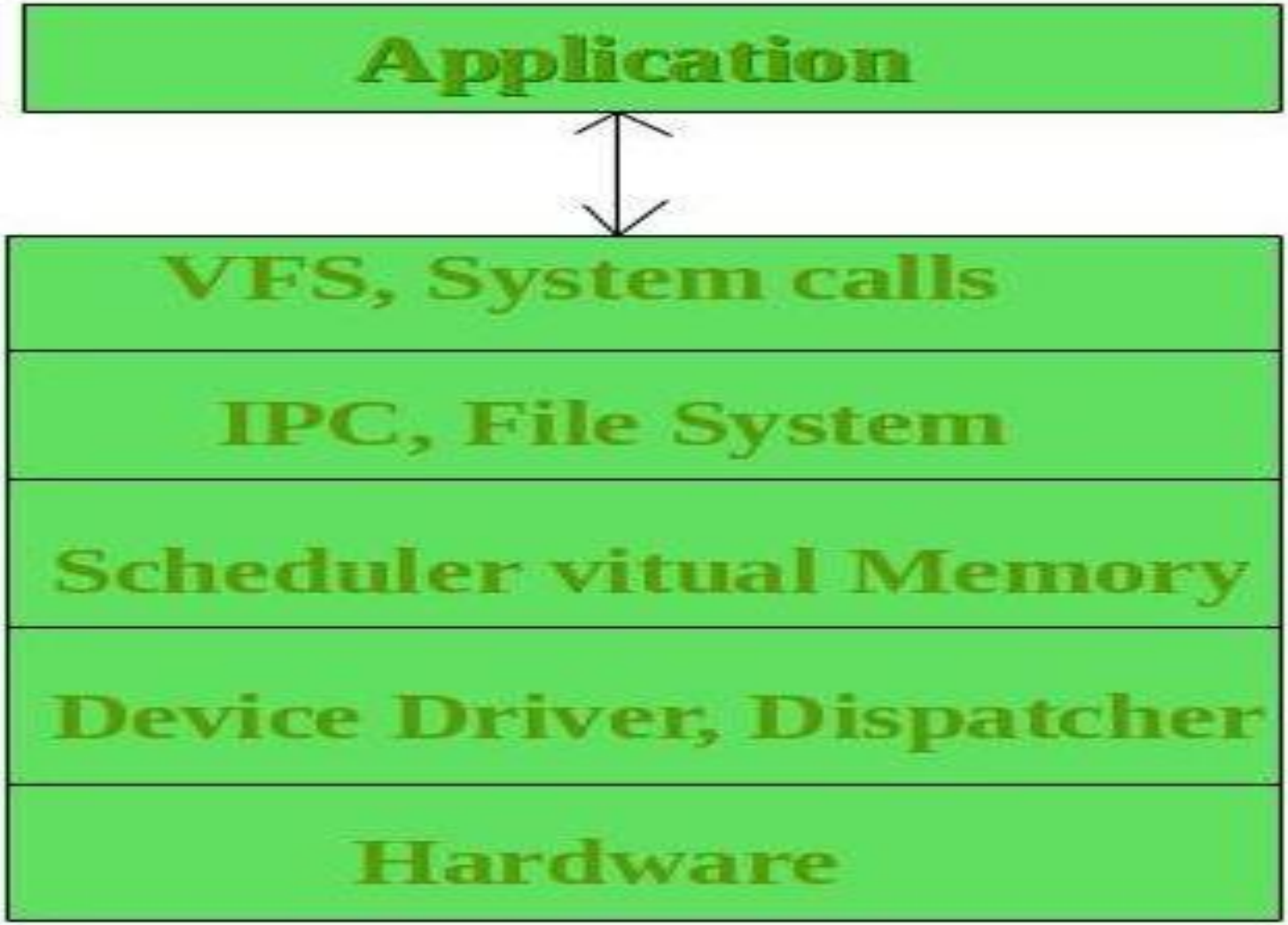
- The architecture of this kernel is small and isolated hence it can function better.
- Expansion of the system is easier, it is simply added in the system application without disturbing the kernel.

Eclipse IDE is a good example of Microkernel Architecture.

Types of Kernel: (b) Monolithic Kernel

Apart from microkernel, **Monolithic Kernel** is another classification of Kernel. Like microkernel this one also manages system resources between application and hardware, but **user services** and **kernel services** are implemented under same address space. It increases the size of the kernel, thus increases size of operating system as well.

This kernel provides CPU scheduling, memory management, file management and other operating system functions through system calls. As both services are implemented under same address space, this makes operating system execution faster.



Monolithic Kernel

Advantages of Monolithic Kernel –

- One of the major advantage of having monolithic kernel is that it provides CPU scheduling, memory management, file management and other operating system functions through system calls.
- The other one is that it is a single large process running entirely in a single address space.
- It is a single static binary file. Example of some Monolithic Kernel based OSs are: Unix, Linux, Open VMS, XTS-400, z/TPF.

Disadvantages of Monolithic Kernel –

- One of the major disadvantage of monolithic kernel is that, if anyone service fails it leads to entire system failure.
- If user has to add any new service. User needs to modify entire operating system.

Basis for Comparison	Microkernel	Monolithic Kernel
Size	Microkernel is smaller in size	It is larger than microkernel
Execution	Slow Execution	Fast Execution
Extendible	It is easily extendible	It is hard to extend
Security	If a service crashes, it does effects on working on the microkernel	If a service crashes, the whole system crashes in monolithic kernel.
Code	To write a microkernel more code is required	To write a monolithic kernel less code is required
Example	QNX, Symbian, L4Linux etc.	Linux,BSDs(FreeBSD,OpenBSD,NetBSD)etc.

Types of Kernel: c) Exo-kernel

- Exo-kernel is a type of operating system that provides application-level management of hardware resources.
- The exokernel architecture is designed to separate resource protection from management to facilitate application-specific customization.
- Exokernels are typically small in size because of their limited operability.
- The main goal of an exokernel is to ensure that there is no forced abstraction, which is what makes an exokernel different from micro- and monolithic kernels.

Benefits of the exokernel operating system

- Improved performance of applications
- More efficient use of hardware resources through precise resource allocation and revocation
- Easier development and testing of new operating systems
- Each user-space application is allowed to apply its own optimized memory management

Drawbacks of the exokernel operating system

- Reduced consistency
- Complex design of exokernel interfaces

Shell

- A shell is software that provides an interface for an operating system's users to provide access to the kernel's services.
- An OS starts a shell for each user when the user logs in or opens a terminal or console window.
- It is named a shell because it is the outermost layer around the operating system kernel.

A kernel is a program that:

- Controls all computer operations.
- Coordinates all executing utilities
- Ensures that executing utilities do not interfere with each other or consume all system
- resources.
- Schedules and manages all system processes.

- By interfacing with a kernel, a shell provides a way for a user to execute utilities and programs.

Shell Contd…

- The shell also provides a user environment that you can customize using initialization files.
- These files contain settings for user environment characteristics, such as:

❑Search paths for finding commands.

❑Default permissions on new files.

❑Values for variables that other programs use.

❑Values that you can customize.

END