Operating Systems

Revya Naik V
Assistant Professor
Computer science and engineering
RGUKT Basar

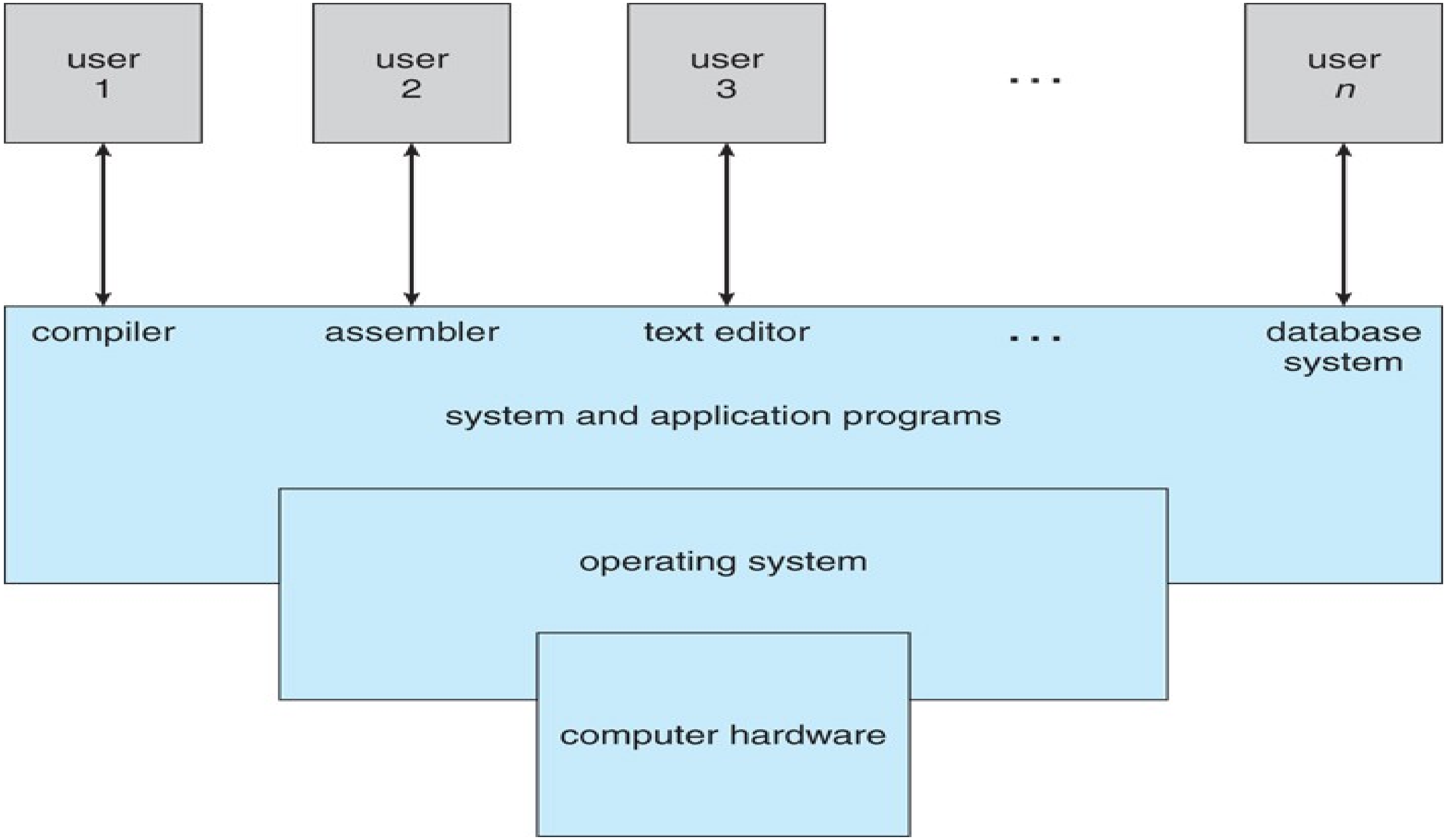
Operating system

- An operating system is a program that manages the computer hardware. It also provides a basis for application programs and acts as an intermediary between the computer user and the computer hardware.
- The main goal of the Operating System is to make the computer environment more convenient to use and Secondary goal is to use the resources in the most efficient manner.
- The main task an operating system carries out is the allocation of resources and services, such as the allocation of memory, devices, processors, and information.
- The operating system also includes programs to manage these resources, such as a traffic controller, a scheduler, memory management, I/O programs, and a file system.

Functions of OS

- **Processor management**:The OS decides the order in each processes have access to the processor and how much time required for each process this function is called process scheduling.
- The OS keep track of process status and process allocation and de-allocation of main memory.
- **Device management**: An OS manages device communication via their respective drivers.
- **File management**: A file system is organized into directories for efficient or easy navigation and usage. The OS provides file access and edit permissions.
- **Memory management**: The OS manages main memory allocation and deallocation to processes.
- **Error detection**: The OS constantly monitors systems to detect errors.
- **Security** : The OS uses password to protect user data and anauthorized access.

The components of computer system



The components of computer system

Hardware: The CPU, memory and I/O devices provides the basic computing resourses for the system.

Application programs: These resources are used to solve user's computing problems. The OS controls the hardware and coordinates its use among the various application programs for the various users.

Operating system:OS is similar to a government, it performs no useful function by itself. It provides an environment within which other programs can do useful work.

Generation of Operating Systems

First Generation: Vacuum Tubes and Plugbords(1945-55)

Computers were not constructed until the Second World War. No programming languages and no OS. In the 1950s, punch cards were introduced; programs were written on cards and read into the system.

Second generation: Transisters and batch OS (1955-65)

System reads jobs from the tape and executes, after that writes output on second tape.

Third generation: Multiprogramming OS (1965-80)

Multiple jobs were loaded into RAM, if particular job or process executed then immediately start another process or job (none preemptive). Keeping CPU always busy.

Fourth Generation: Multitasking or time sharing OS (1980-present)

Network and distributed operating systems, using this OS remote login (login one system to another for copying and accessing files).

Types of Operating systems

Batch OS: Operator takes similar jobs having the same requirement and group them into batches.

Time sharing OS: Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of CPU as they use a single system.

Distributed OS: Interconnected computers communicate with each other using a shared communication network. The user can access files or software that are not actually present on his system.

- Independent systems possess thier own memory unit and CPU . These aare referrre to as loosly coupled sysrems or DS.
- Failure of one will not affect the other network commnucation.
- Failure of main network it will affect the communications.

Clustering OS: two or more computers joined together with LAN. High availability

Real time OS :The time interval required to process and repond to inputs is very small. Medical systems, weapons system, scientific systems, robotic systems etc.

- **Hard RTOS** time constraints are very strict and even the shortest posssible delay is not acceptable. Ex; In cars airbags os, rocket launching os etc
- . **Soft RTOS** : time constriants is less strict
- **Embedded OS**: used in embedded devices.

Services of OS

Program execution: OS manages how a program is going to be executed. Program saved in memory then compiled and executed. OS handles deadlock (no two processes executed at the same time).

I/O operations: the OS manages i/o operations and establishes communication between users and hardware. Device drivers softwares are managed by OS.

File management : The OS grants file permissions(read only, read-write). OS stores files in HardDisk, USB drive etc.

Services of OS

Error handling: OS detects errors during process executions, devices errors etc.

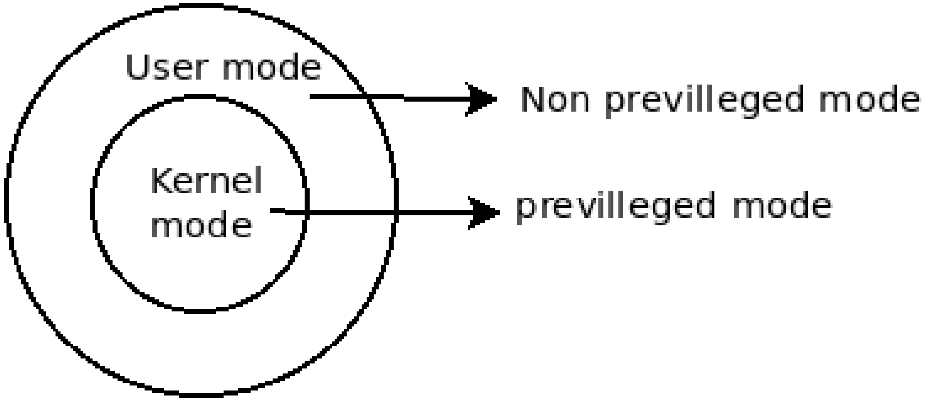
Resource management: CPU, memory and I/O devices are managed by OS.

Interprocesses communication: Inter process communication managed by OS.

Protection and security : OS uses username and password for protecting user data and system resources.

System calls

- System calls provide an interface to the services made available by an operating sytem.

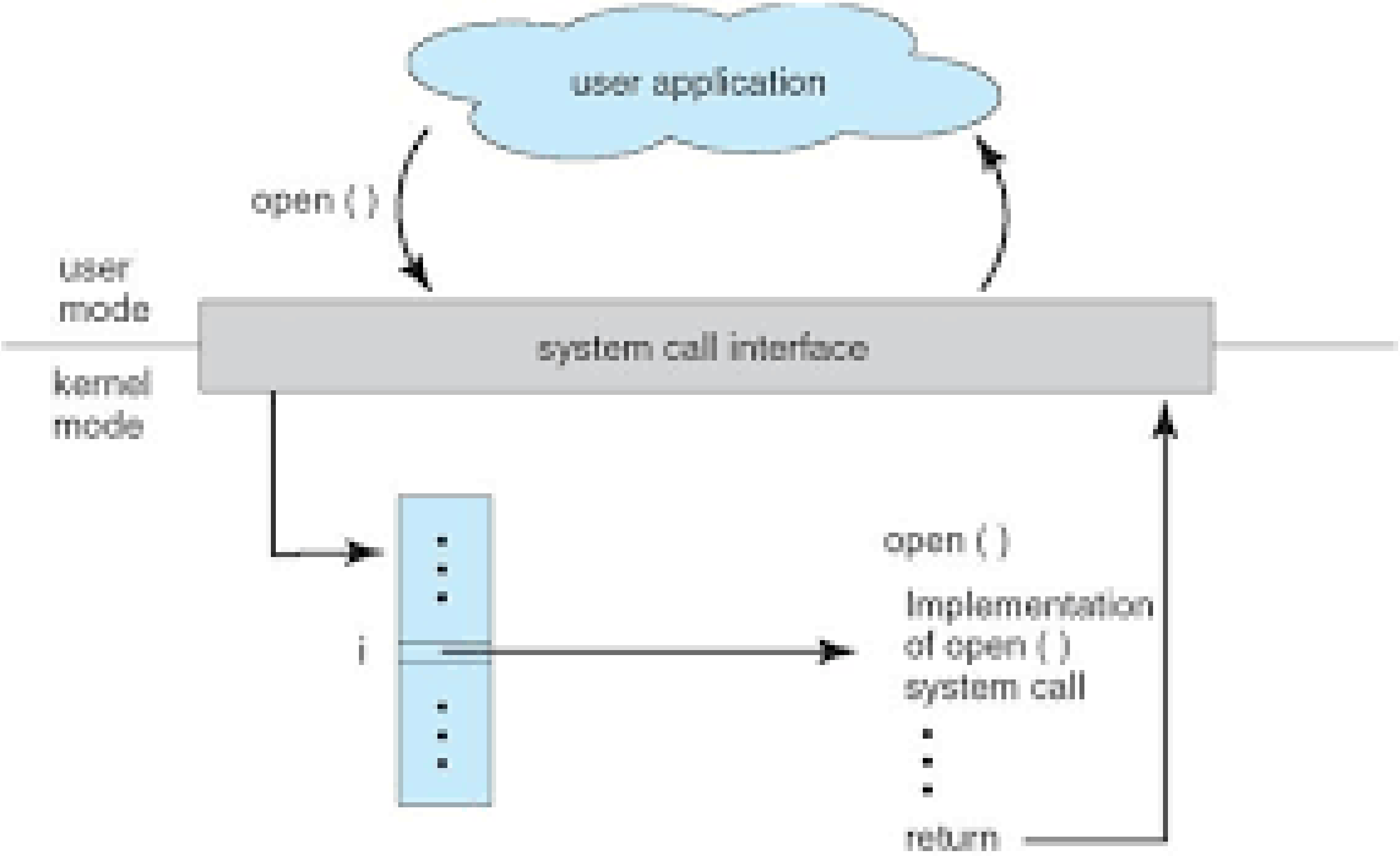


- If the program is executing in user mode then program may not access directly resources like memory, cpu. Error occurs in user mode then system not crashes.
- When program switches from user mode to kernel mode and vice versa is called context switching.
- Kernel mode is nothing but previlleged mode, kernel mode program can directly access cpu, memory. If error occurs in kernel mode program then system crashes.

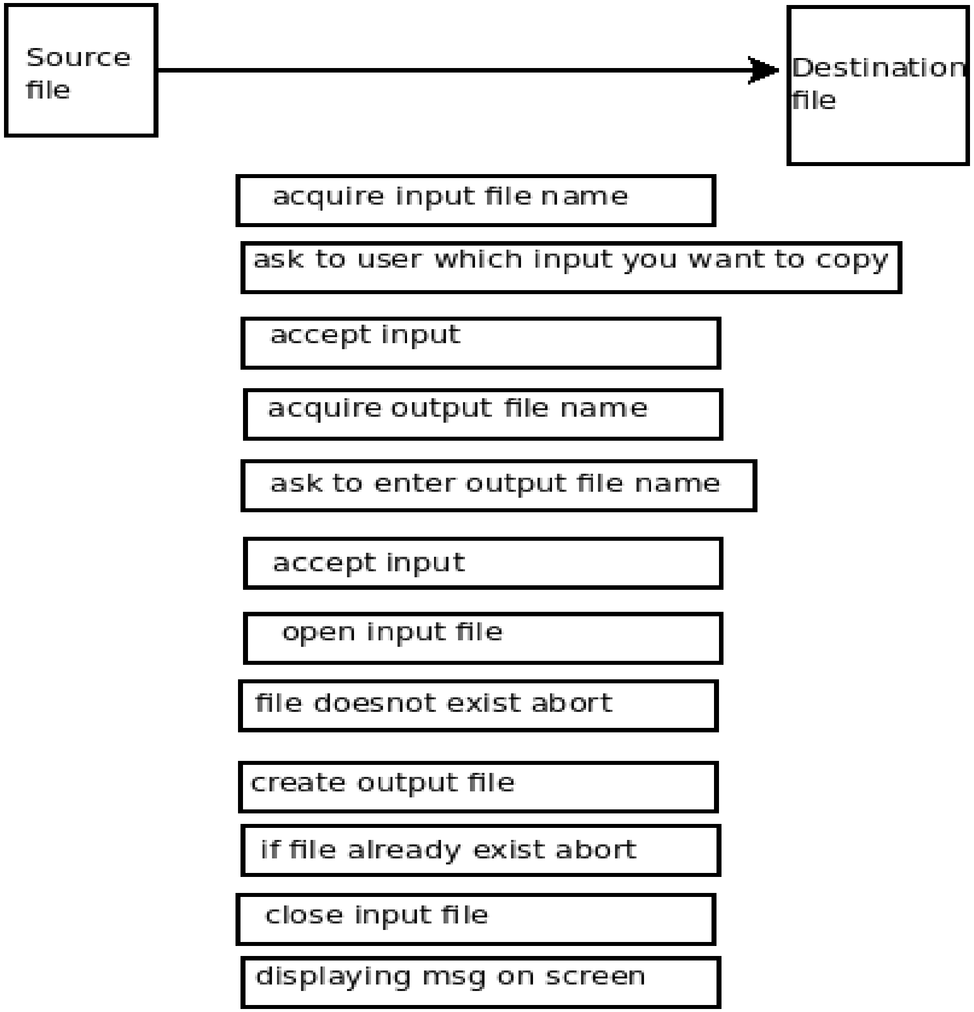
System calls

- Switches from user mode to kernel and a call made by program to access certain resources that call is known as system call.
- System calls made by program when it needs to access certain resources.
- System call is the programmatic way in which a computer program request a service from the kernel of the operating system.
- They are many system calls are executed per second. System calls are executed always.

System calls



System call



Source file

Destination file

acquire input file name

ask to user which input you want to copy

accept input

acquire output file name

ask to enter output file name

accept input

open input file

file doesnot exist abort

create output file

if file already exist abort

close input file

displaying msg on screen

copy from one file to another

Types of system calls

- **Process control**

  end, abort, load execute, create process, terminate process, get process attributes, set process attributs, wait for time, allocate and free memory.

- **File manipulation**

  create file, delete file,open, close, read, write, file reposition, get file attributes, set file attributes.

- **Information maintaince**

  get time or date, set time or date, get process, file or device, set process, file or device, get system data, set system data.

- **Divice management**

  request device, release device, read, write, reposition, logically attach or detach device.

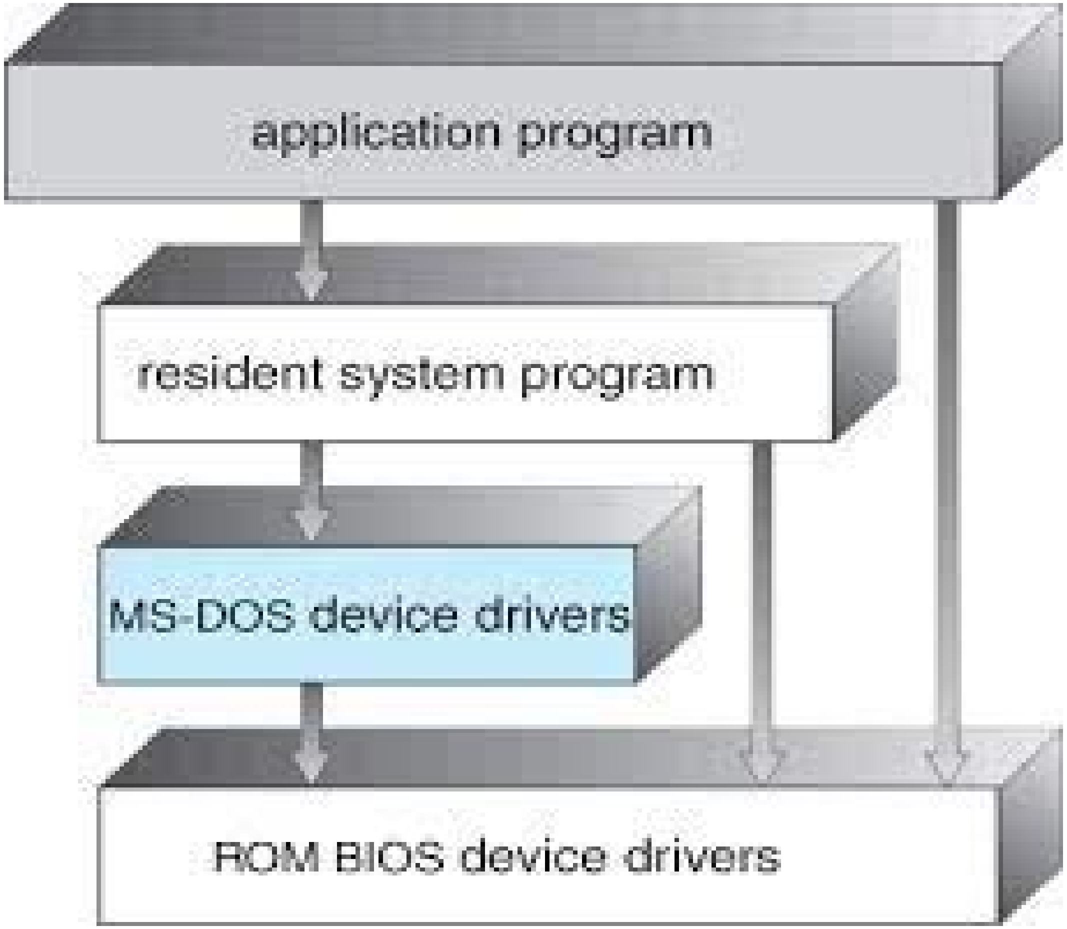
- **Communications**

  create, delete communication connections, send, recieve messages, transfer status information, attach or detach remote devices.

Fork() and exec() system calls

- Fork() system call use to create separate, duplicate process.
- If fork() system call called n times then total 2^n process created, one parent process, remaining 2^n-1 are child process.
- Exec() system call replace same existing process id.

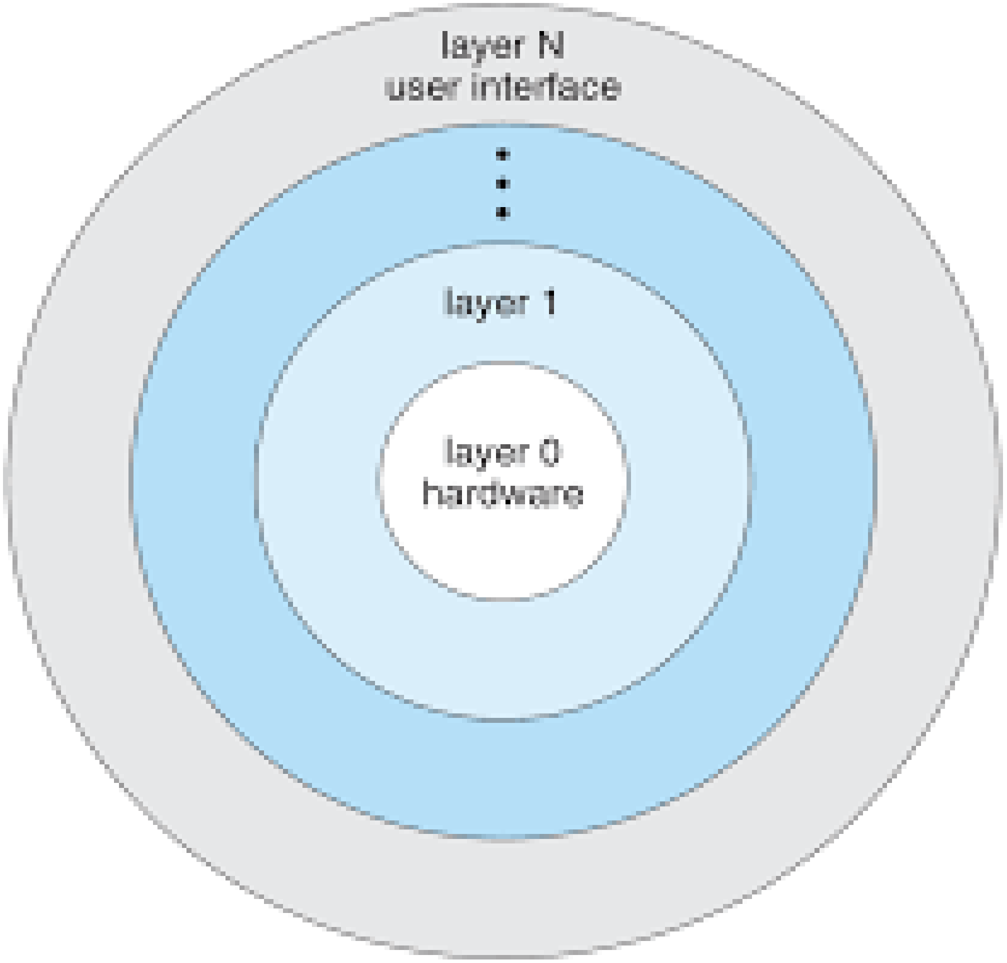
Structures of OS (Simple structure)



Simple structure of OS

- In simple structure OS levels of functionality are not well separated.
- Application programs can access directly BIOS hardware.
- If the user program fails then entire system crashes.
- It was designed for Intel 8088, no dual mode and no hardware protection.

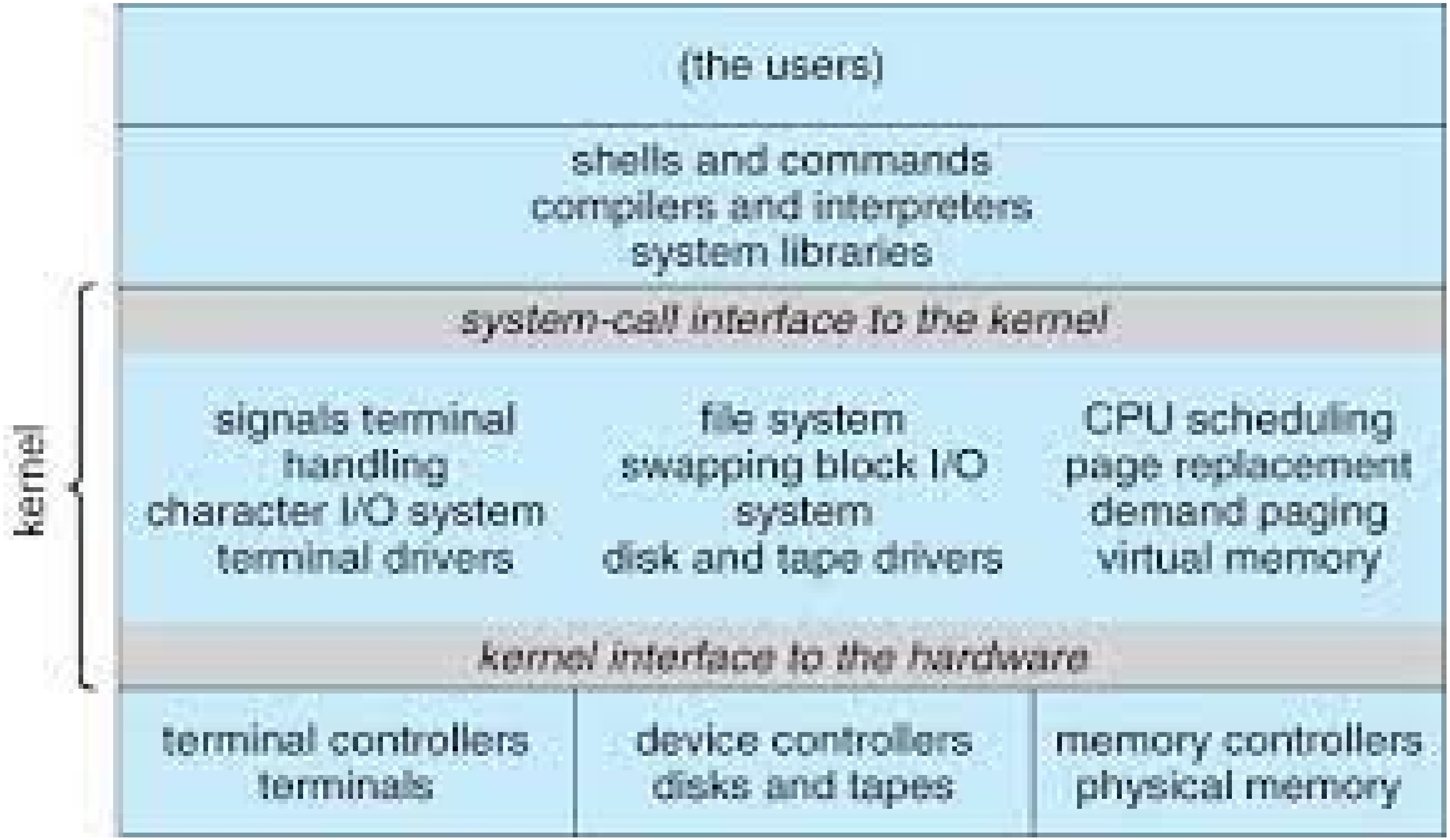
Layered operating system



Layered operating system

- In which OS is broken into a number of layers. Bottom layer (layer 0) is the hardware; the highest (layer N) is the user interface.
- The main advantage of the layered approach is simplicity of construction and debugging.
- Layer wise debugging if error is found during debugging of a particular layer, the error must be on that layer, because the below layers already debugged.
- The backing storage device at lower layer.

Monolithic structure of OS (unix)



(the users)

shells and commands
compilers and interpreters
system libraries

system-call interface to the kernel

signals terminal handling
character I/O system
terminal drivers

file system
swapping block I/O system
disk and tape drivers

CPU scheduling
page replacement
demand paging
virtual memory

kernel interface to the hardware

terminal controllers
terminals

device controllers
disks and tapes

memory controllers
physical memory

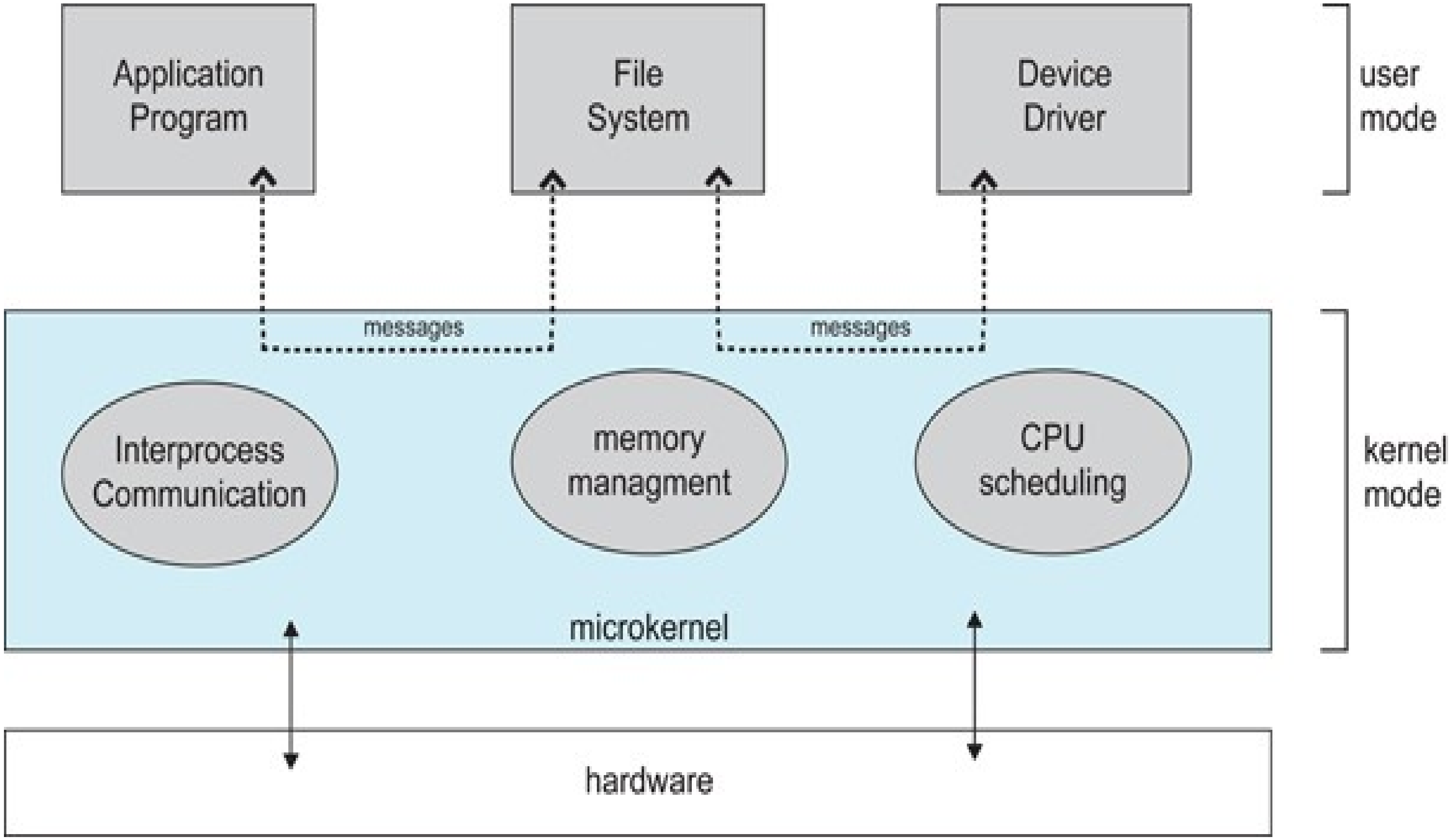
Monolithic kernel

- Everything below the system call interface and above the physical hardware is the kernel.
- The kernel provides the file system, CPU scheduling, memory management, and other OS functions through system calls.
- It was difficult to implement and maintain. If the error is occurs in any of the program then entire system crashes.

Microkernels

- Unix kernel became large and difficult to manage.
- Microkernel provide minimal process and memory management.
- The main function of the microkernel is to provide a communication facility between client program and the various services that are also running in user space.
- If client program wishes to access a file, it must interact with the file server.
- If any services added into user mode then no need to update kernel.
- It is more secure because most of services are running in user mode. If a service fails, rest of the OS untouched.

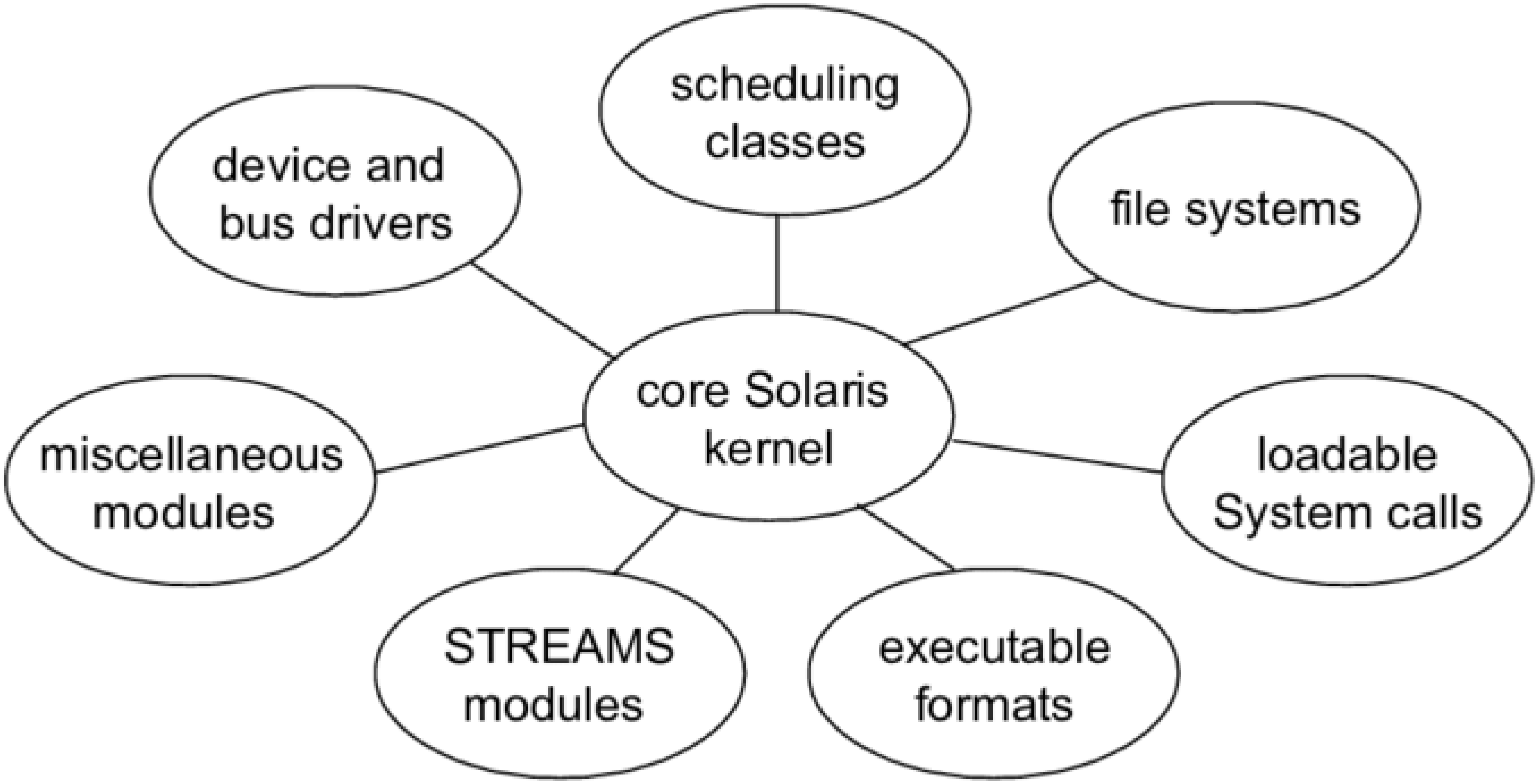
Microkernel operating system



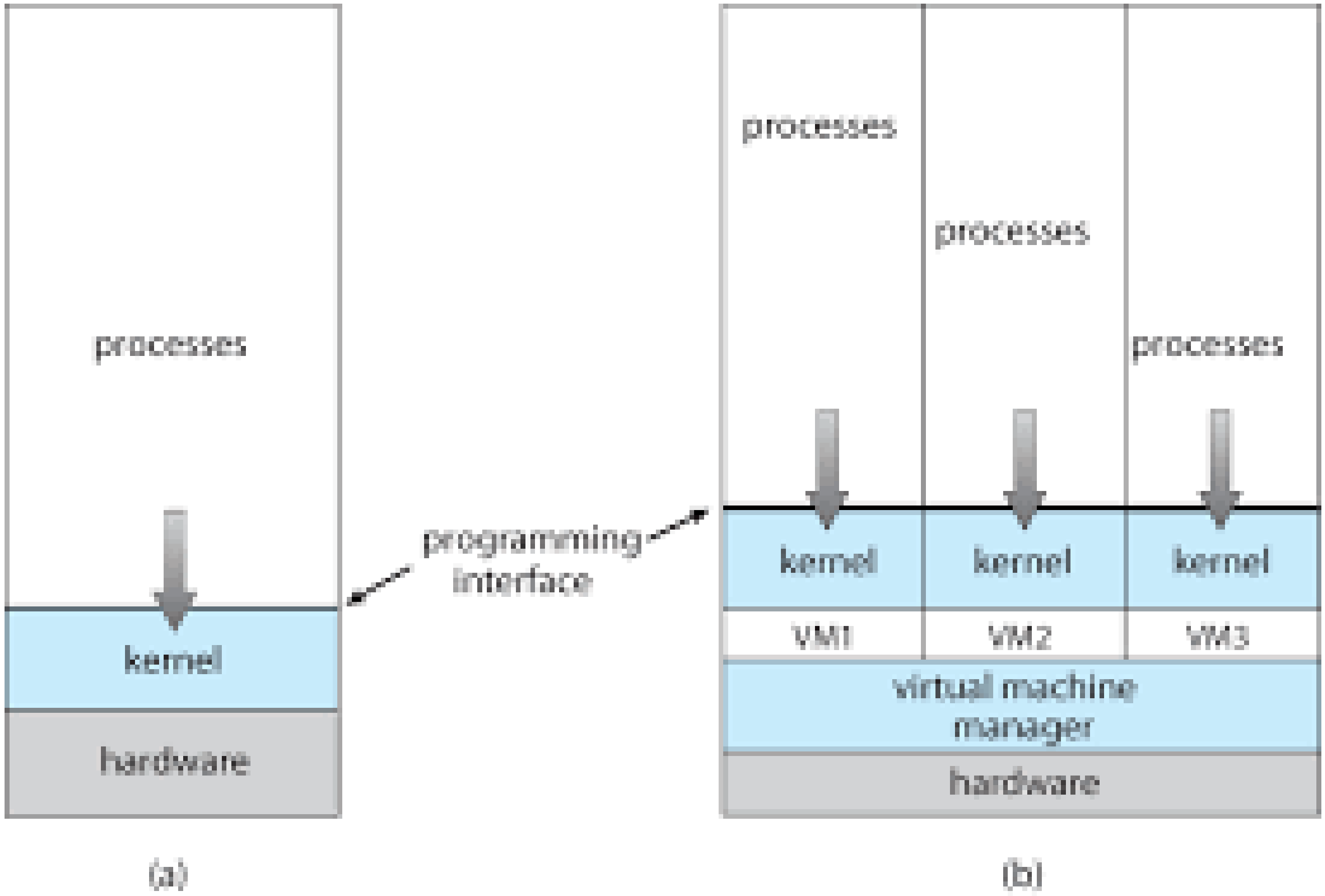
Modular kernel

- Object oriented programming techniques used to create a modular kernel. It uses dynamically loadable strategy and is common in modern implementation of Unix, such as solaris, linux, and mac OS.
- It is more flexible than a layered system in that any module can call any other module.
- The approach is like microkernel approach in that the primary module has only core functions and knowledge of how to load and communicate with other modules; but it is more efficient.

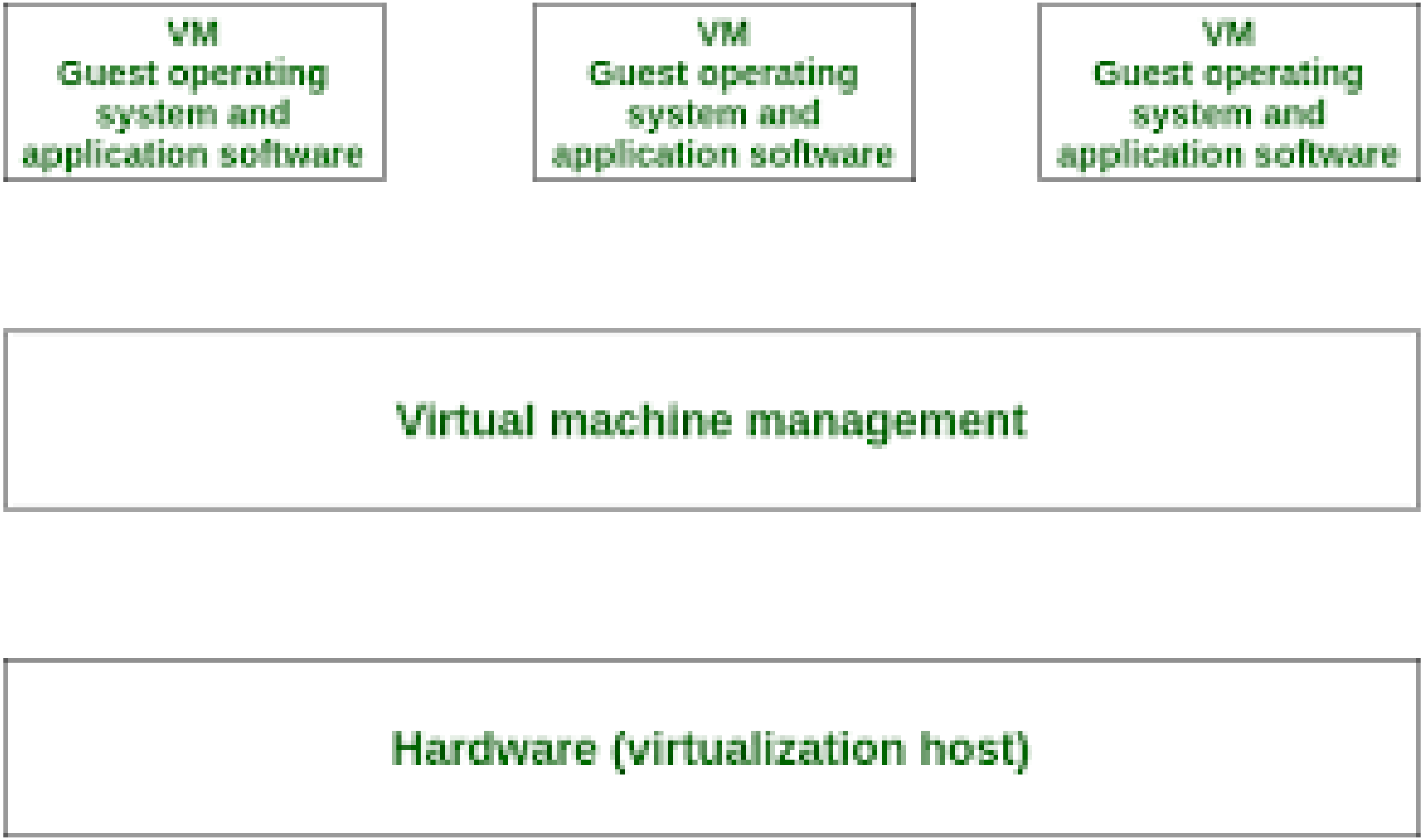
Modular kernel



Virtual machines



Virtual machines



VM
Guest operating system and application software

VM
Guest operating system and application software

VM
Guest operating system and application software

Virtual machine management

Hardware (virtualization host)

Virtual machines

- Divided a mainframe into multiple virtual machines, each running its own operating system.
- A major difficultly with the VM virtual machine approach involved disk systems. For all virtual machines common HardDisk.