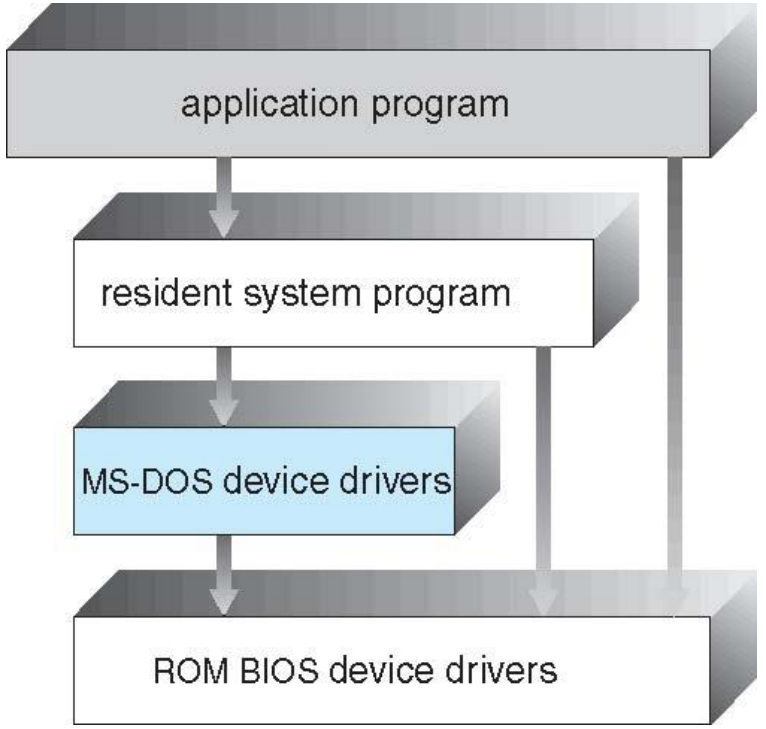
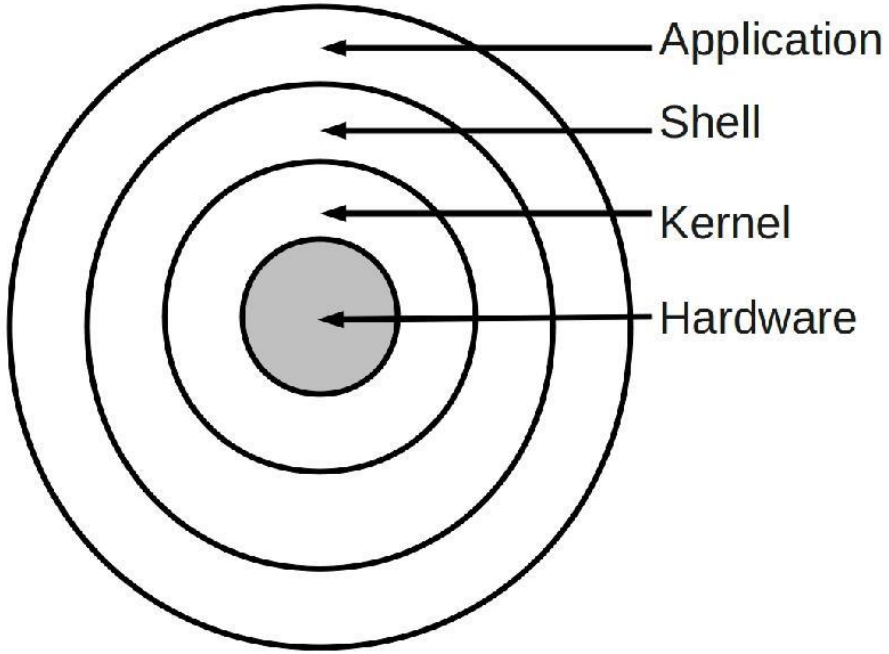
OPERATING SYSTEM: CSET209

SIMPLE STRUCTURE -- MS-DOS

- MS-DOS – written to provide the most functionality in the least space
  - Not divided into modules
  - Its interfaces and levels of functionality are not well separated



OPERATING SYSTEM ARCHITECTURE OF TRADITIONAL UNIX



- An operating system is a program that acts as an interface between a user of a computer and the computer resources. The purpose of an operating system is to provide an environment in which a user may execute programs.
- **Hardware:** The hardware consists of the memory, CPU, arithmetic-logic unit, various bulk storage devices, I/O, peripheral devices and other physical devices.

OPERATING SYSTEM ARCHITECTURE

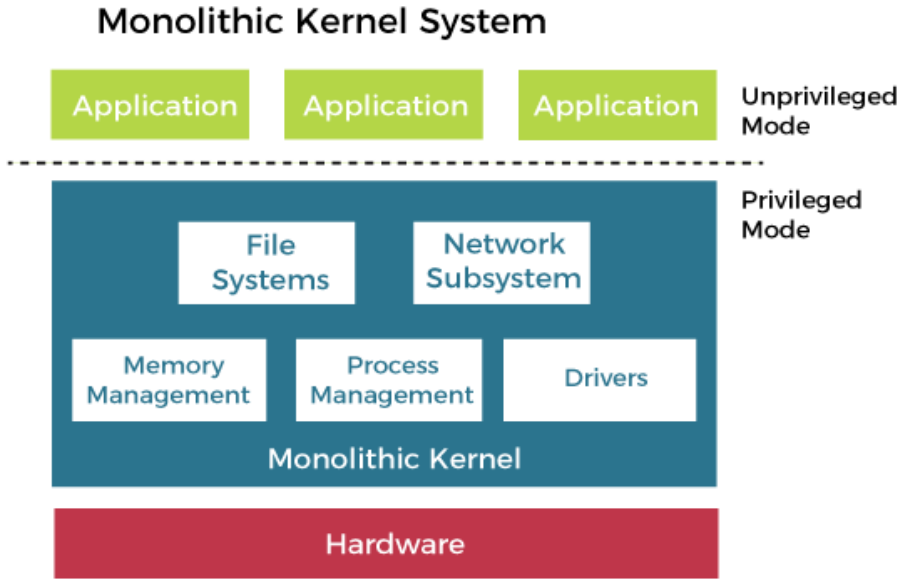
- **Kernel:**
  - In computing, the kernel is the central component of most computer operating systems;
  - The kernel provides the file system, CPU scheduling, memory management, and other operating-system functions.
- **Shell:**
  - A shell is a piece of software that provides an interface for users to access to the **services of a kernel.** E.g. Bash (Bourne again shell) in Ubuntu.
  - The name shell originates from shells being an outer layer of interface between the user and the innards of the operating system (the kernel)
  - Operating system shells generally fall into one of two categories: command-line and graphical

MONOLITHIC STRUCTURE OF OPERATING SYSTEM

- The monolithic operating system is a very basic operating system in which file management, memory management, device management, and process management are directly controlled within the kernel.
- Operating systems that use monolithic architecture were first time used in the 1970s.
- The monolithic operating system is also known as the monolithic kernel.

MONOLITHIC SYSTEM ARCHITECTURE

- A monolithic design of the operating system architecture makes no special accommodation for the special nature of the operating system.
- Although the design follows the separation of concerns, no attempt is made to restrict the privileges granted to the individual parts of the operating system.
- The entire operating system executes with maximum privileges. The communication overhead inside the monolithic operating system is the same as that of any other software, considered relatively low.



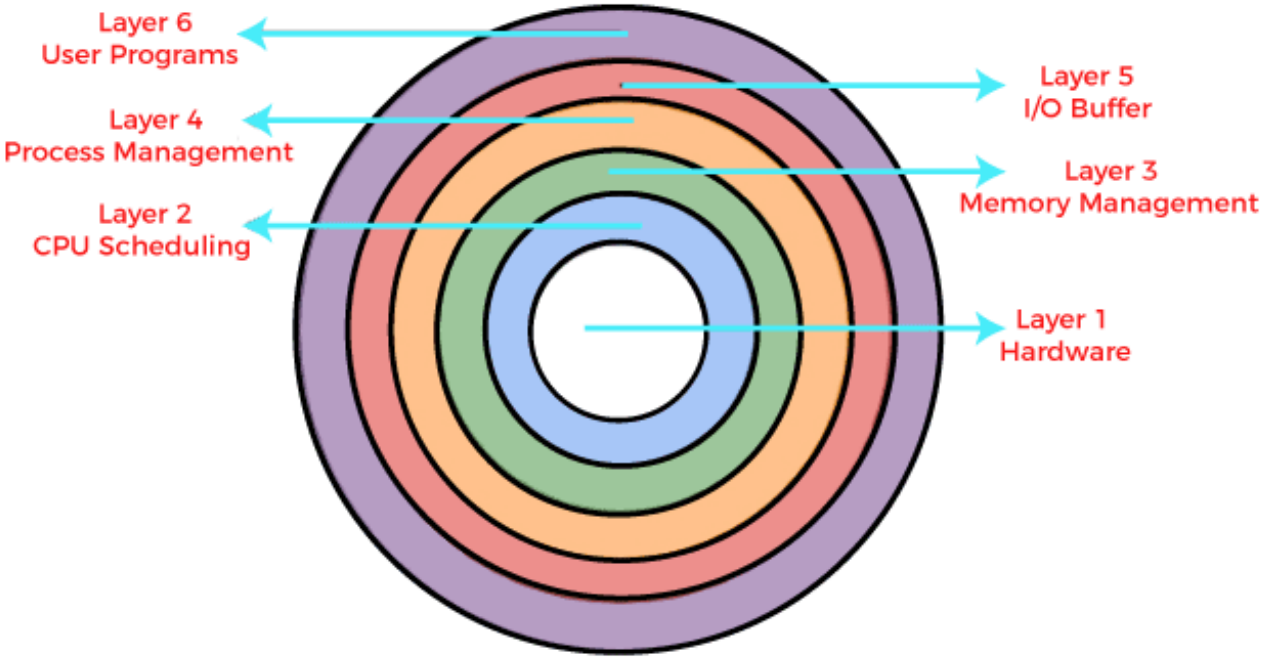
ADVANTAGES OF MONOLITHIC KERNEL

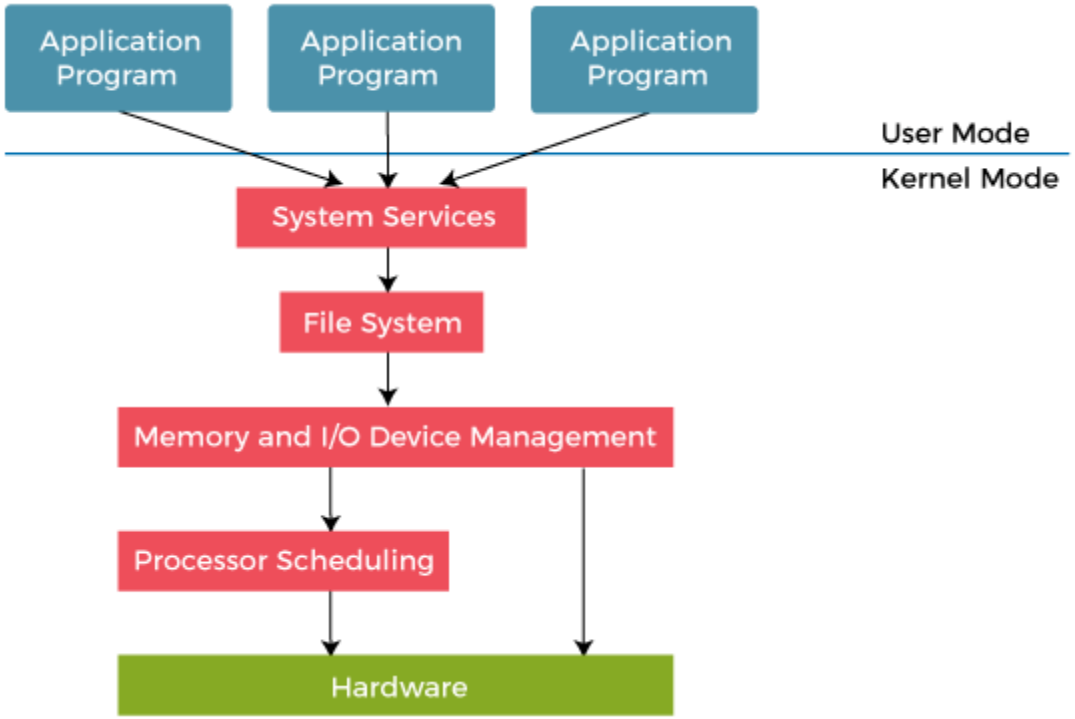
- The execution of the monolithic kernel is quite fast as the services such as memory management, file management, process scheduling etc. are implemented under the same address space.
- A process runs completely in a single address space in the monolithic kernel.
- The monolithic kernel is a static single binary file.

DISADVANTAGES OF MONOLITHIC KERNEL

- If any service fails in the monolithic kernel, it leads to the failure of the entire system.
- To add any new service, the entire operating system needs to be modified by the user.

LAYERED OS ARCHITECTURE





LAYERED OS ARCHITECTURE

- It was created to improve the pre-existing structures like the Monolithic structure (UNIX) and the Simple structure (MS-DOS).
- The outermost layer is User Interface layer and the innermost layer is the Hardware layer.
- A typical operating-system layer—say, layer M—consists of data structures and a set of routines that can be invoked by higher-level layers.
- Example – The Windows NT operating system.
- The modularity of layered operating systems allows the implementation of each layer to be modified without requiring any modification to adjacent layers.

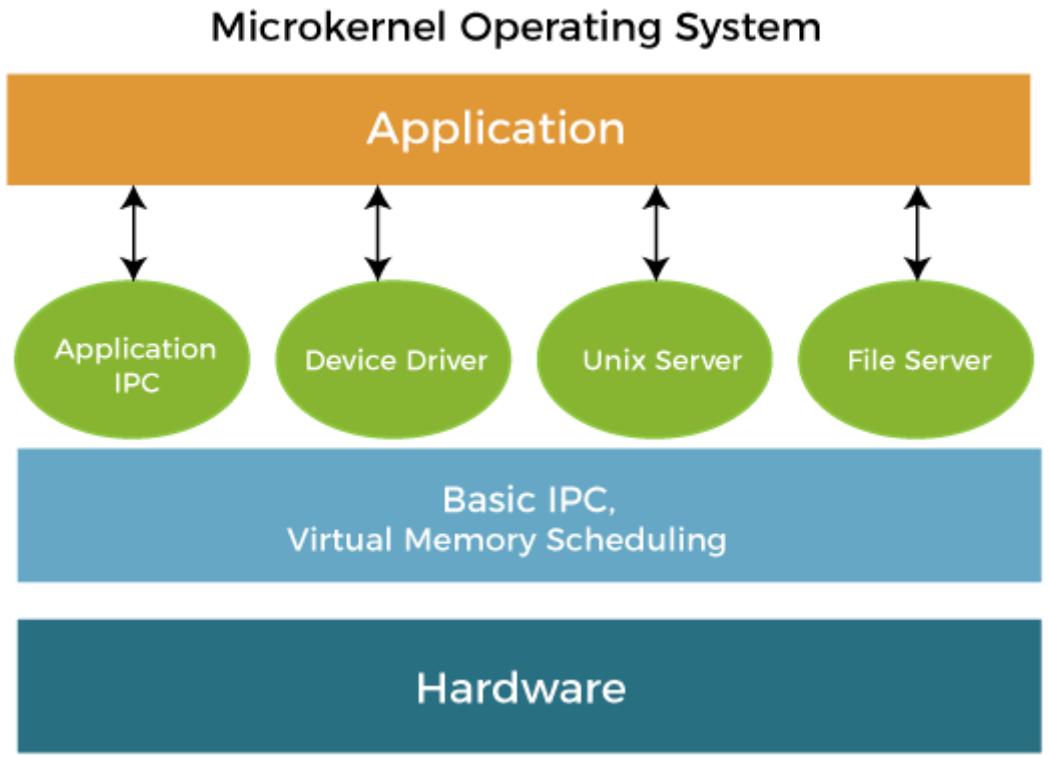
ADVANTAGES: LAYERED OS ARCHITECTURE

- **Modularity :** This design promotes modularity as each layer performs only the tasks it is scheduled to perform.
- **Easy debugging :** As the layers are discrete so it is very easy to debug. Suppose an error occurs in the CPU scheduling layer, so the developer can only search that particular layer to debug, unlike the Monolithic system in which all the services are present together.
- **Easy update :** A modification made in a particular layer will not affect the other layers.
- **No direct access to hardware :** The hardware layer is the innermost layer present in the design. So a user can use the services of hardware but cannot directly modify or access it, unlike the Simple system in which the user had direct access to the hardware.
- **Abstraction :** Every layer is concerned with its own functions. So the functions and implementations of the other layers are abstract to it

DISADVANTAGE: LAYERED OS ARCHITECTURE

- **Complex and careful implementation:** As a layer can access the services of the layers below it, so the arrangement of the layers must be done carefully. For example, the device driver for the backing store (disk space used by virtual-memory algorithms) must be at a lower level than the memory-management routines, because memory management requires the ability to use the backing store.
- **Slower in execution:** If a layer wants to interact with another layer, it requests to travel through all the layers present between the two interacting layers. Thus it increases response time, unlike the Monolithic system, which is faster than this. Thus an increase in the number of layers may lead to a very inefficient design.
- **Functionality:** It is not always possible to divide the functionalities. Many times, they are interrelated and can't be separated.

MICROKERNEL OS ARCHITECTURE



This method structures the operating system by removing all nonessential components from the kernel and implementing them as system and user-level programs.

MICROKERNEL OS ARCHITECTURE

- A microkernel architecture includes only a very small number of services within the kernel in an attempt to keep it small and scalable.
- The services typically include low-level memory management, inter-process communication and basic process synchronisation to enable processes to cooperate.
- In microkernel designs, most operating system components, such as process management and device management, execute outside the kernel with a lower level of system access.
- Microkernels are highly modular, making them extensible, portable and scalable. Operating system components outside the kernel can fail without causing the operating system to fall over..

ADVANTAGES: MICROKERNEL OS ARCHITECTURE

- **The microkernel also provides more security and reliability, since most services are running as user rather than kernel processes.**
- Microkernels are modular, and the various modules may be swapped, reloaded, and modified without affecting the kernel.
- **It makes extending the operating system easier. All new services are added to user space and consequently do not require modification of the kernel.**
- 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.
- The microkernel interface helps in enforcing a more modular system structure.
- Server failure is treated the same as any other user program failure.

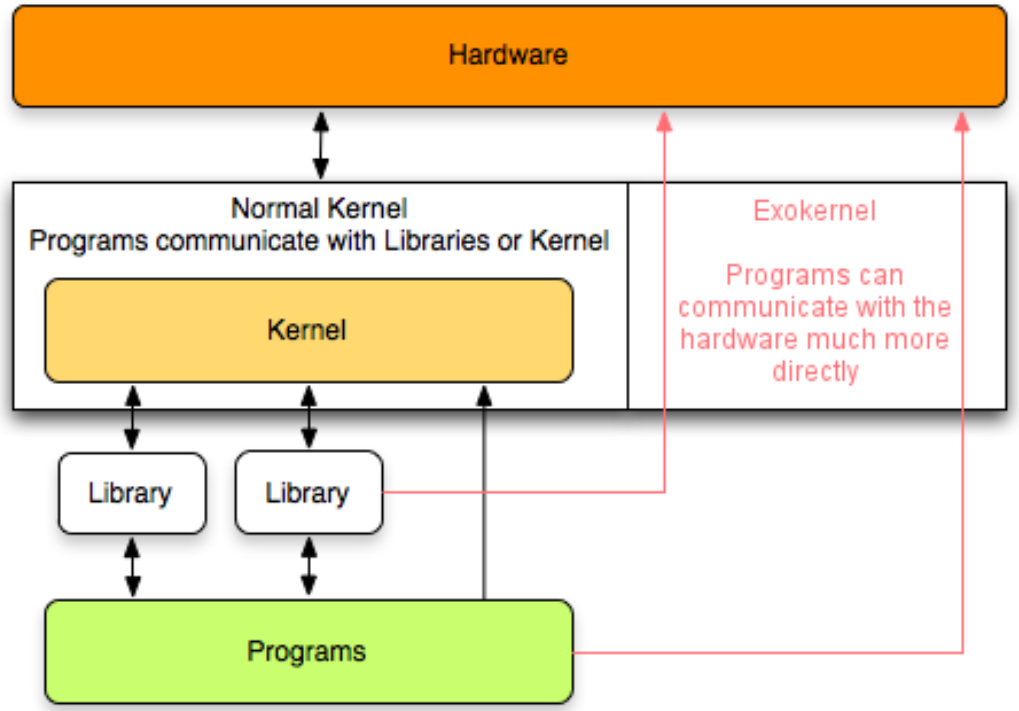
DISADVANTAGES: MICROKERNEL OS ARCHITECTURE

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

Terms	Monolithic Kernel	microkernel
Definition	A monolithic kernel is a type of kernel in operating systems where the entire operating system works in the kernel space.	A microkernel is a kernel type that provides low-level address space management, thread management, and interprocess communication to implement an operating system.
Address space	In a monolithic kernel, both user services and kernel services are kept in the same address space.	In microkernel user services and kernel, services are kept in separate address spaces.
Size	The monolithic kernel is larger than the microkernel.	The microkernel is smaller in size.
Execution	It has fast execution.	It has slow execution.
OS services	In a monolithic kernel system, the kernel contains the OS services.	In a microkernel-based system, the OS services and kernel are separated.
Extendible	The monolithic kernel is quite	The microkernel is easily extendible.
Code	Less coding is required to write a monolithic kernel.	A microkernel is required more coding.
Example	Linux, FreeBSD, OpenBSD, NetBSD, Microsoft Windows (95, 98, Me), Solaries, HP-UX, DOS, OpenVMS, XTS-400, etc.	QNX, Symbian, L4L.inux, Singularity, K42, Mac OS X, Integrity, PikeOS, HURD, Minix, and Coyotos.

EXOKERNEL

- Exokernel is an operating system developed at the MIT that provides application-level management of hardware resources.
- This architecture is designed to separate resource protection from management to facilitate application-specific customization.



PRINCIPLES OF EXOKERNELS

1. Separate protection and management : Resource management is restricted to functions necessary for protection.

2. Expose allocation : Applications allocate resources explicitly.

3. Expose name : Exokernels use physical names wherever possible.

4. Expose revocation : Exokernels let applications to choose which instance of a resource to give up.

5. Expose information : Exokernels expose all system information and collect data that applications cannot easily derive locally.

ADVANTAGES OF EXOKERNELS

1. Significant performance increase.
2. Applications can make more efficient and intelligent use of hardware resources by being aware of resource availability, revocation and allocation.
3. Ease development and testing of new operating system ideas. (New scheduling techniques, memory management methods, etc)

DISADVANTAGES OF EXOKERNELS

1. Complexity in design of exokernel interfaces.
2. Less consistency.

VIRTUAL MACHINES

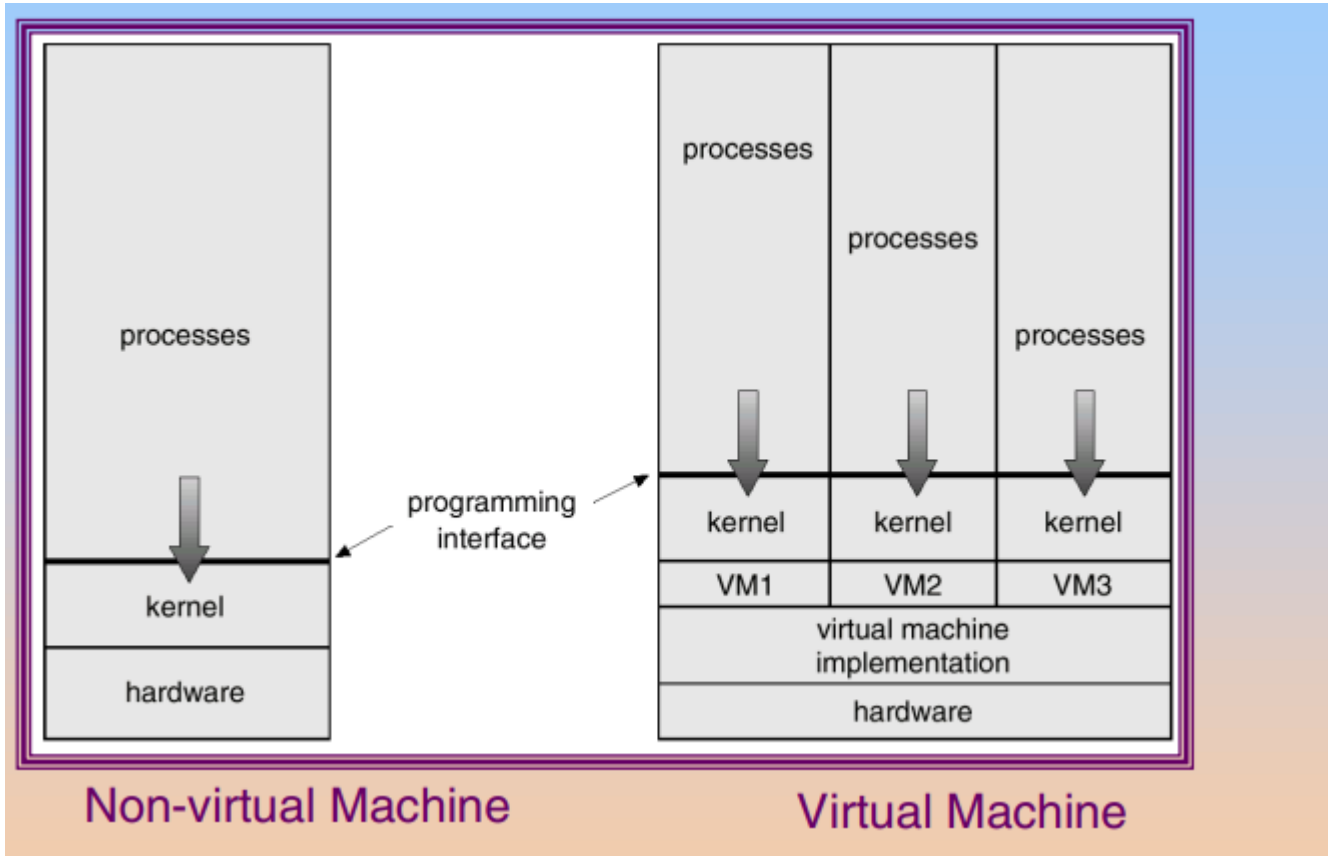
- Virtualization is a technology that allows operating systems to run as applications within other operating systems..
- With virtualization, in contrast, an operating system runs within another operating system.
- Running multiple virtual machines allowed (and still allows) many users to run tasks on a system designed for a single user.
- The VMM runs the guest operating systems, manages their resource use, and protects each guest from the others.
- Purpose:

  ✦ for exploration or to run applications written for operating systems other than the native host

  ✦ Companies writing software for multiple operating systems can use virtualization to run all of those operating systems on a single physical server for development, testing, and debugging.

  ✦ Cloud computing uses virtualization.

VIRTUAL MACHINES



ADVANTAGES/DISADVANTAGES OF VIRTUAL MACHINES

- The virtual-machine concept provides complete protection of system resources since each virtual machine is isolated from all other virtual machines. This isolation, however, permits no direct sharing of resources.
- A virtual-machine system is a perfect vehicle for operating-systems research and development. System development is done on the virtual machine, instead of on a physical machine and so does not disrupt normal system operation.
- The virtual machine concept is difficult to implement due to the effort required to provide an exact duplicate to the underlying machine.

THANK YOU
?