CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

UNIVERSITY INSTITUTEOF ENGINEERING

Bachelor of Engineering (Computer Science & Engineering)

Operating System (CST-328)

Introduction to Operating System

Font size 24

DISCOVER . LEARN . EMPOWER

Lecture 1

Introduction to the Operating System

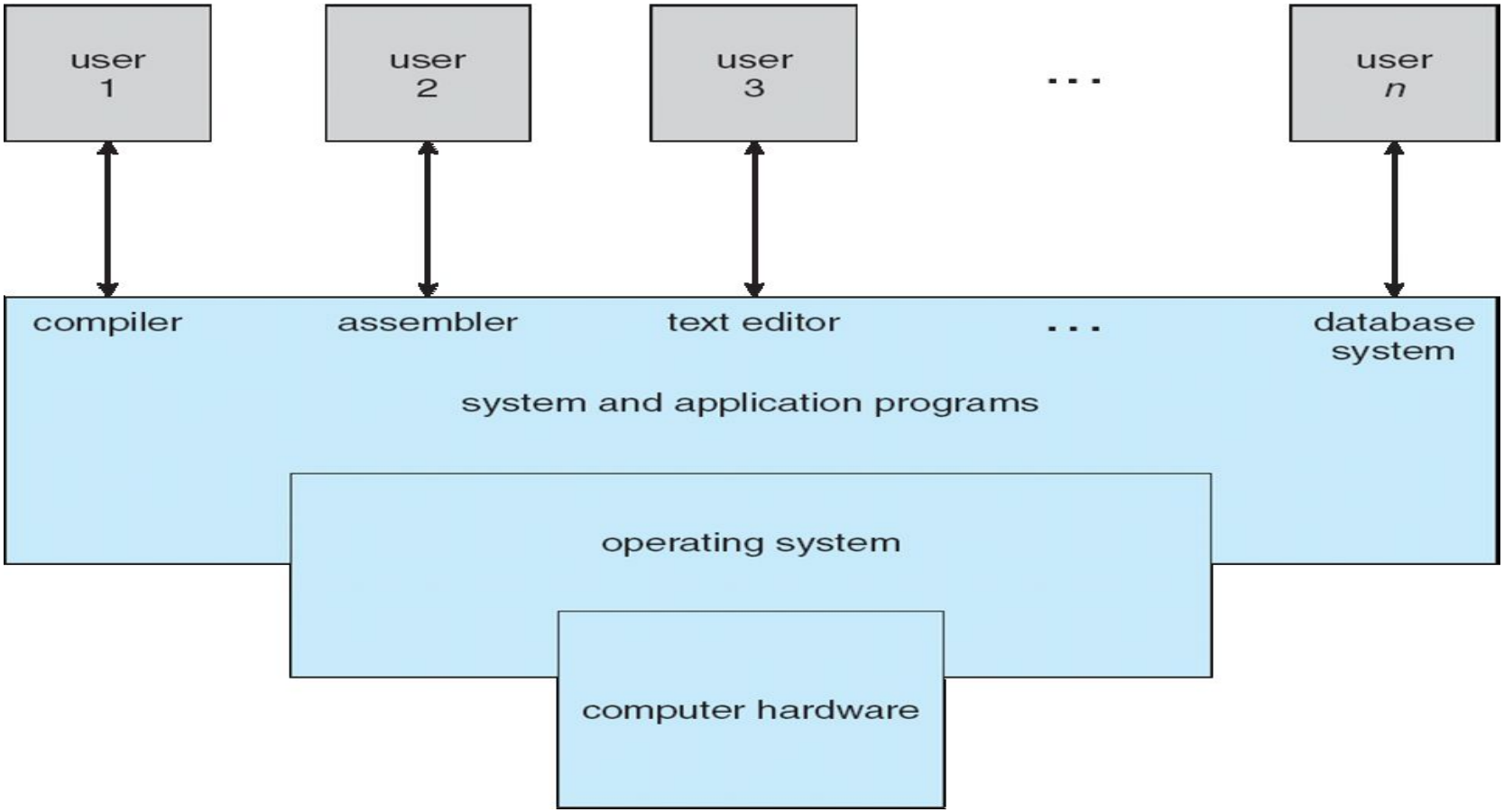
Why do we need an OS?

- Allows you to hide details of hardware by creating an abstraction
- Easy to use with a GUI
- Offers an environment in which a user may execute programs/applications
- The operating system must make sure that the computer system convenient to use
- Operating System acts as an intermediary among applications and the hardware components
- It provides the computer system resources with easy to use format
- Acts as an intermediate between all hardware's and software's of the system

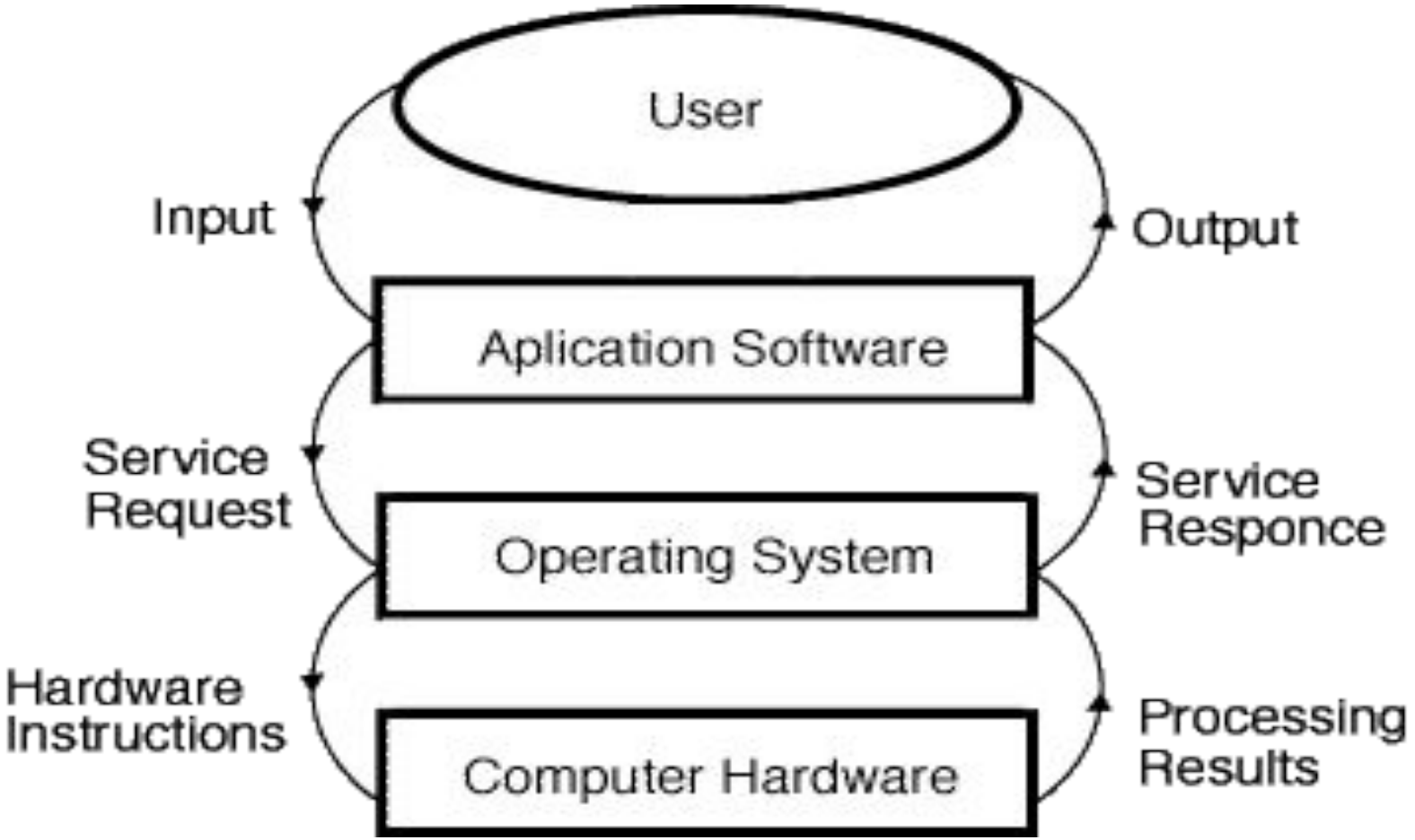
Computer System Components

1. **Hardware –** provides basic computing resources (CPU, Memory, I/O devices, Communication).
2. **Operating System –** controls and coordinates use of the hardware among various application programs for various users.
3. **System & Application Programs –** ways in which the system resources are used to solve computing problems of the users (Word processors, Compilers, Web browsers, Database systems, Video games).
4. **Users –** (People, Machines, other computers).

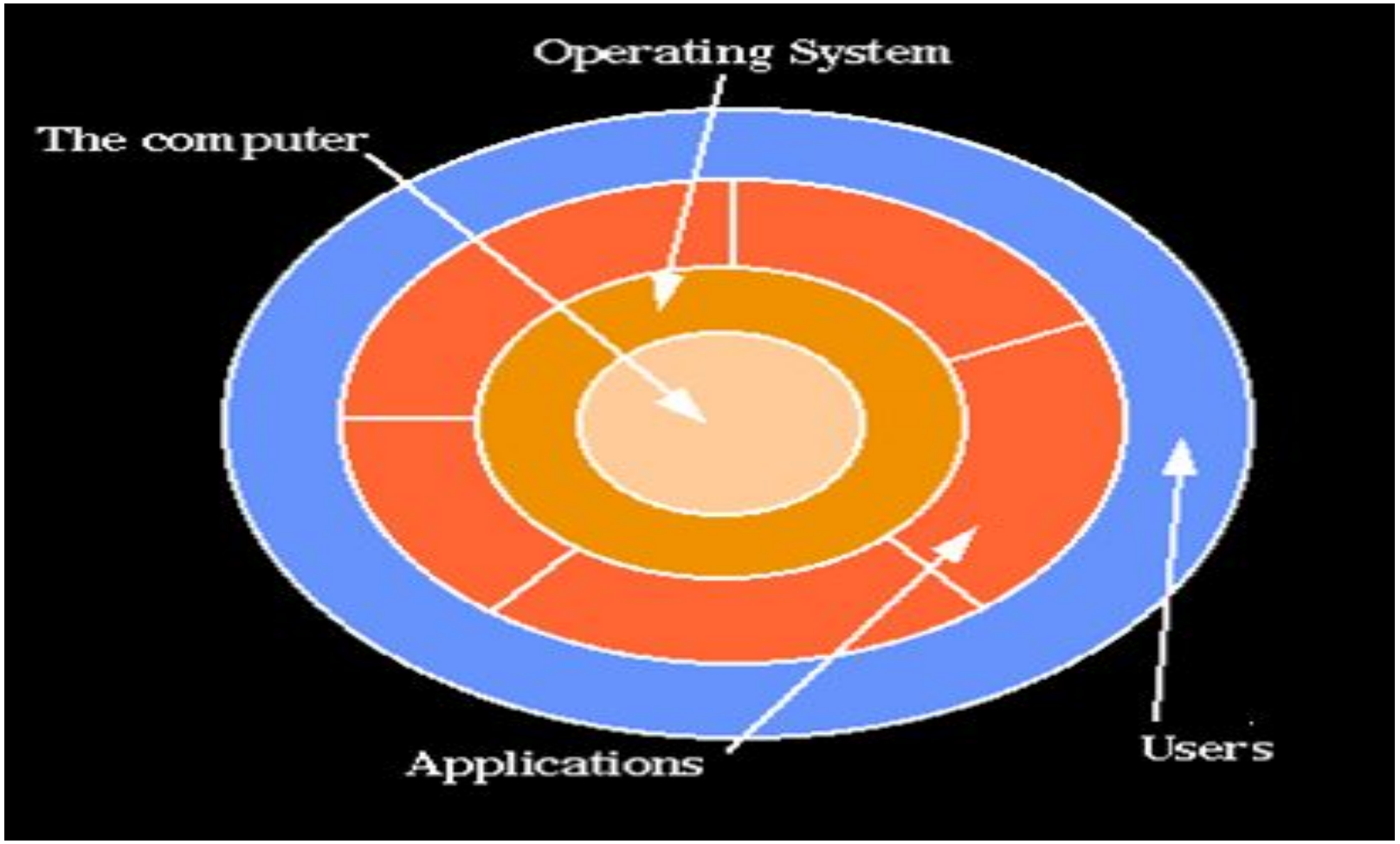
Static View of System Components



Dynamic View of System Components



Hierarchical view of computer system



What is OS?

✔Operating System is a software, which makes a computer to actually work.

✔ It is the software the enables all the programs we use.

✔ The OS organizes and controls the hardware.

✔ OS acts as an interface between the application programs and the machine hardware.

Examples: Windows, Linux, Unix and Mac OS, etc.

- **Operating system goals:**
  - Execute user programs and make user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

Responsibilities of an Operating System

There are three basic responsibilities (in literature):

1. Resource Manager – manages and allocates resources.
2. Control program – controls the execution of user programs and operations of I/O devices.
3. Command Executer – Provides an environment for running user commands.

- But one more modern view: the Operating System as a Virtual Machine.

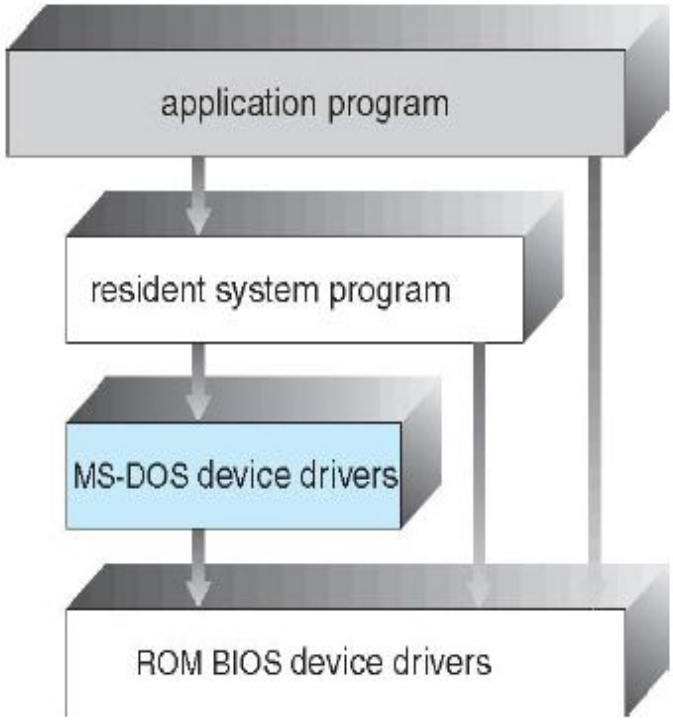
Operating-System Structure

- General-purpose OS is very large program

- Various ways to structure ones
  - Simple structure – MS-DOS
  - More complex -- UNIX
  - Layered – an abstraction
  - Microkernel -Mach

Simple Structure -- MS-DOS

- MS-DOS – written to provide the most functionality in the least space Not divided into modules
- Limited in H/W functionality
- Although MS-DOS has some structure, its interfaces and levels of functionality are not well separated



Non Simple Structure -- UNIX

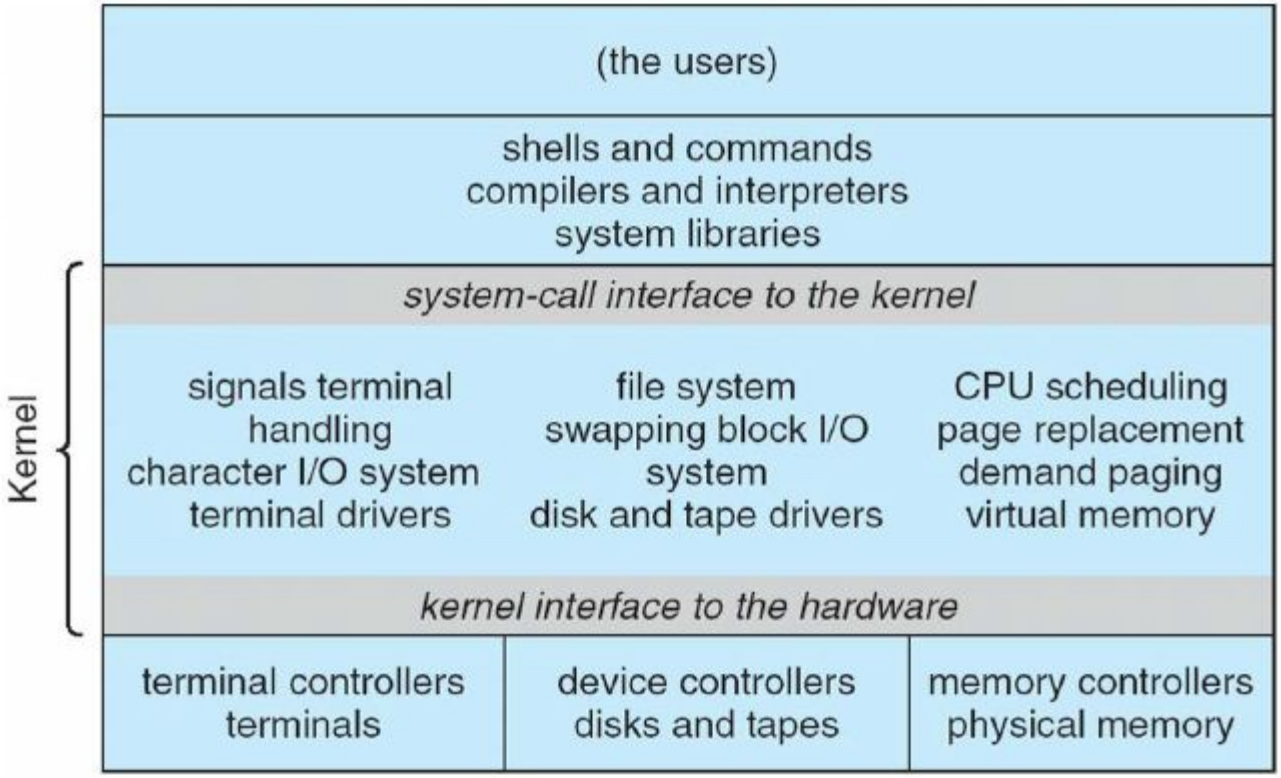
- UNIX – limited by hardware functionality, the original UNIX operating system had limited structuring.

The UNIX OS consists of two separable parts

- Systems programs
  - The kernel Consists of everything below the system-call interface and above the physical hardware
  - Provides the file management, CPU scheduling, memory management, and other operating-system functions; a large number of functions for one level

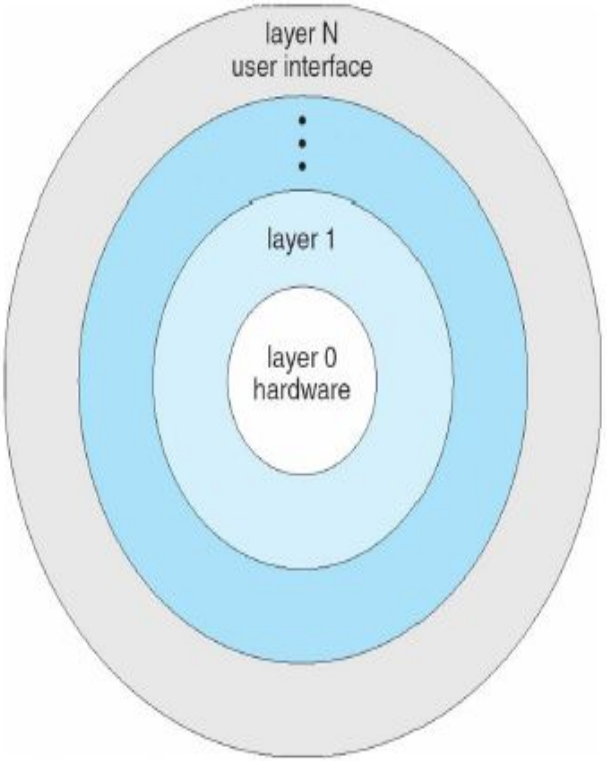
Non Simple Structure -- UNIX

Beyond simple but not fully layered



Layered Approach

- The operating system is divided into a number of layers (levels), each built on top of lower layers.
- The bottom layer (layer 0), is the hardware; the highest (layer N) is the user interface.
- Main advantage of layered approach is Modularity.
- With modularity, layers are selected such that each uses functions (operations) and services of only lower-level layers.
- The major difficulty with layered approach is to divide the layers carefully, because a layer can use only those layers which are below it.



Microkernel System Structure

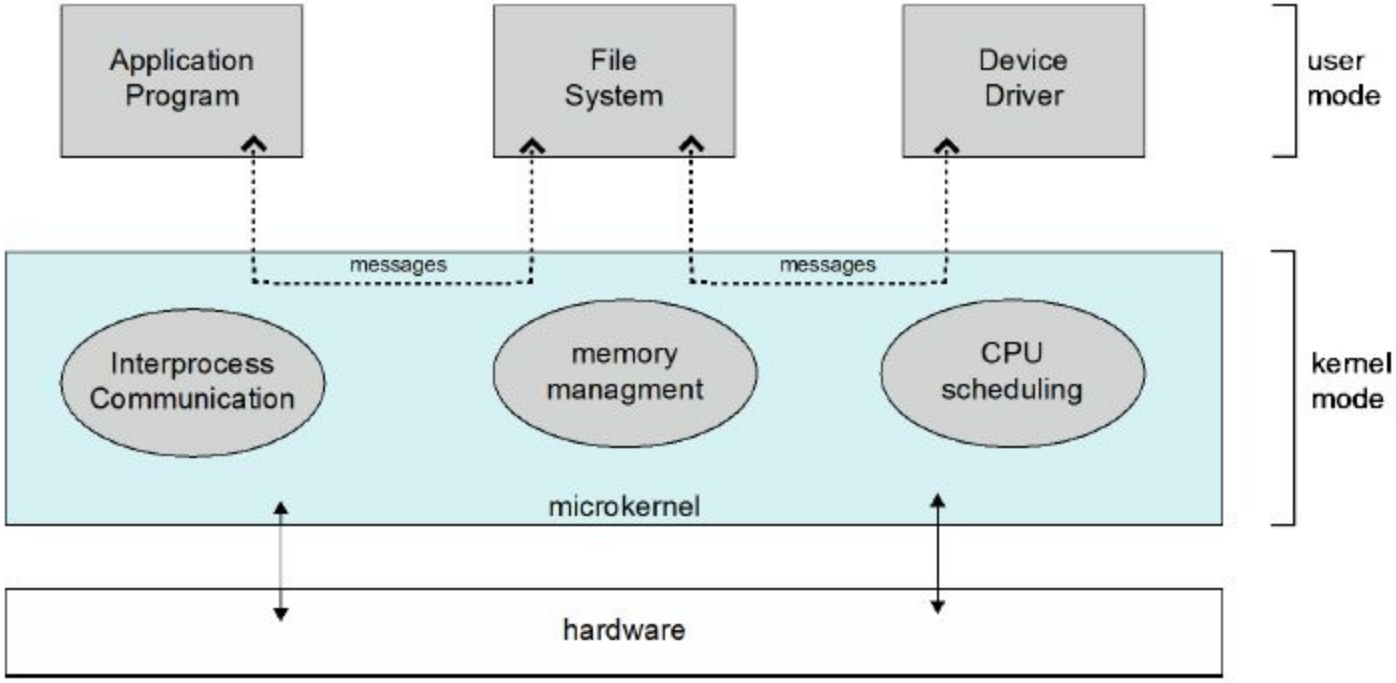
- Moves as much from the kernel into user space
- **Mach** example of **microkernel**
  - Mac OS X kernel (**Darwin**) partly based on Mach

- Communication takes place between user modules using **message passing**
- **Benefits**:
  - ✔ Easier to extend a microkernel
  - ✔ Easier to port the operating system to new architectures
  - ✔ More reliable (less code is running in kernel mode)
  - ✔ More secure

- **Detriments:** Performance overhead of user space to kernel space communication

Microkernel System Structure

Kernel is a computer program that manages I/O requests from software and translates them into data processing instructions for CPU and other electronic components of computer.



What is a Kernel?

- The kernel is a computer program at the core of a computer's operating system with complete control over everything in the system. It is an integral part of any operating system.

Features of Kernel

- Low-level scheduling of processes
- Inter-process communication
- Process synchronization
- Context switching

User Interfaces to Operating Systems

- **Command interpreter or shell**

  Text-driven, command-response interface style

- **Graphical user interface**

  Menu-driven and/or direct manipulation interface style

Applications of Operating System

- Security
- Control over system performance
- Job accounting
- Error detecting aids
- Coordination between other softwares and users

Conclusion

This lecture makes the student familiar with basics of operating systems like OS Definition, need of OS, OS structure, kernel, applications of OS etc.

References

https://www.includehelp.com/c-programming-questions/

https://www.studytonight.com/operating-system/

https://computing.llnl.gov/tutorials/

https://www.tutorialspoint.com/operating_system/index.htm#:~:text=An%20operating%20system%20(OS)%20is,software%20in%20a%20computer%20system.

https://www.javatpoint.com/os-tutorial

https://www.guru99.com/operating-system-tutorial.html

https://www.geeksforgeeks.org/operating-systems/