DATE / / /

Module 1 - Overview of operating systems-

* Part 1.1 - → Introduction to OS ✓
  → Interaction of OS and hardware ✓
  → Goals of OS ✓
  → Basic functions of OS ✓
  → OS services ✓
  → System calls ✓
  → Types of System calls ✓

* Part 1.2 - → Types of OS ✓
  → Batch, Multiprogramming ✓
  → Time-Sharing, Parallel, Distributed and Real-Time OS- ✓

* Part 1.3 → Structure of O.S ✓
  → Monolitnic, Layered, Virtualization ✓
  → Virtual Machines ✓
  → MicroKernels ✓

* Part 1.4 - → Modern UNIX Systems ✓

* Introduction to O.S.

→ An operating system (O.S) is a software program that acts as an intermediary between computer hardware and the user

→ It manages various hardware and software components while also providing a user-friendly environment for users to interact with the applications.

Here are some ways in which O.S acts as an intermediate.

① Hardware Abstraction - Abstracts/Hides the underlying hardware details into a consistent and standardized format.

② Device Drivers - OS provides device drivers which are software components that facilitate communication between OS and hardware devices.

Some other functions will be discussed further like Memory Management, Process Management, System calls etc.

DATE / / /

* Main objectives / Goals of O.S.

① Efficiency - It is to ensure efficient utilization of hardware resources. Which involves managing processes, memory and I/O operations. in such a way that it minimizes resource wastage and maximizes system performances.

② Consistency :- O.S should provide a consistent and predictable environment for both users and applications. This consistency ensures that similar operations behave the same way across different applications.

③ Ability to Evolve :- An O.S should be designed with the ability to evolve over time. As technology advances and hardware architectures change - O.S should be adaptable to those new environments and requirements.

x x

* Basic Functions of O.S.

① Process Management :- The O.S should be able to manipulate processes within a computer which include:

a) Process creation - The O.S creates and initializes new processes. A process ~~cog~~ represents an executing program and its associated resources.

b) Process Scheduling - The O.S manages the execution of processes on the C.P.U, allocating time slices to each process and determining the order in which process runs.

c) Process termination - When a process completes its execution, or encounters an error, the O.S terminates the process, freeing up the resources.

② Memory Management - OS should manage the memory resources. This includes

a) - Memory Allocation - OS allocates memory spaces to processes based on their requirements managing available memory to prevent conflicts.

b) Memory Protection - O.S enforces memory protection mechanisms to prevent one process to access other process's memory.

PAGE No.
DATE / /

c) Memory Recycling :- When a process is complete, O.S reclaims its memory

③ File System Management. - O.S manages file manipulation within a system.

ⓐ File Creation and deletion - O.S provides tools for creating and deleting files managing their metadata and storage locations.

b) File Reading and writing - O.S facilitates reading data from and writing data to files abstracting low-level storage operation.

c) File permissions and access control. - O.S enforces permissions to control who can access, modify or delete files.

④ Device Management

a) Device Recognition :- O.S identifies and initializes hardware devices such as printers, disk drives and network interfaces.

b) Device Drivers : O.S provides device drives that allow software applications to interact with hardware devices.

PAGE No.
DATE / /

* O.S services.

1. Program Development : This service involves tools provided by the operating system to assist programmers in creating software applications.

Includes application like text editors, code editors, IDE's etc.

2. Program Execution :- O.S is responsible for loading and executing programs in memory. It allocates memory space for programs to run and manages the execution of multiple processes concurrently.

3. Access I/O devices :- O.S provides a standardised interface for applications to interact with input and output (I/O) devices such as Keyboard, mouse, monitors, printers network interfaces etc.

4. Controlled Access to Files :- O.S manages file systems, which include organizing and storing files on storage devices.

Also enforces Access control mechanisms to ensure only authorized users and processes can access, modify or delete files.

5. System Access :- This service involves user authentication and user account management users must log in with valid credential before gaining access to the system.

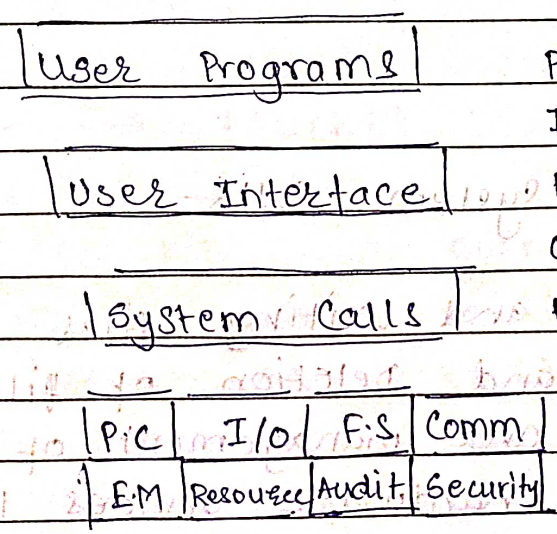
controls user privileges and permissions, determining which resources and operation each user can access.

---

* System calls.

→ A system call is a mechanism that provides the interface b/w a process and the O.S.

→ Its a programmatic method in which a computer program requests a service from the Kernel.



P.C → Program Control.
I/O → Input/output
F.S → File System
Comm → Communications.
E.M → Error Management

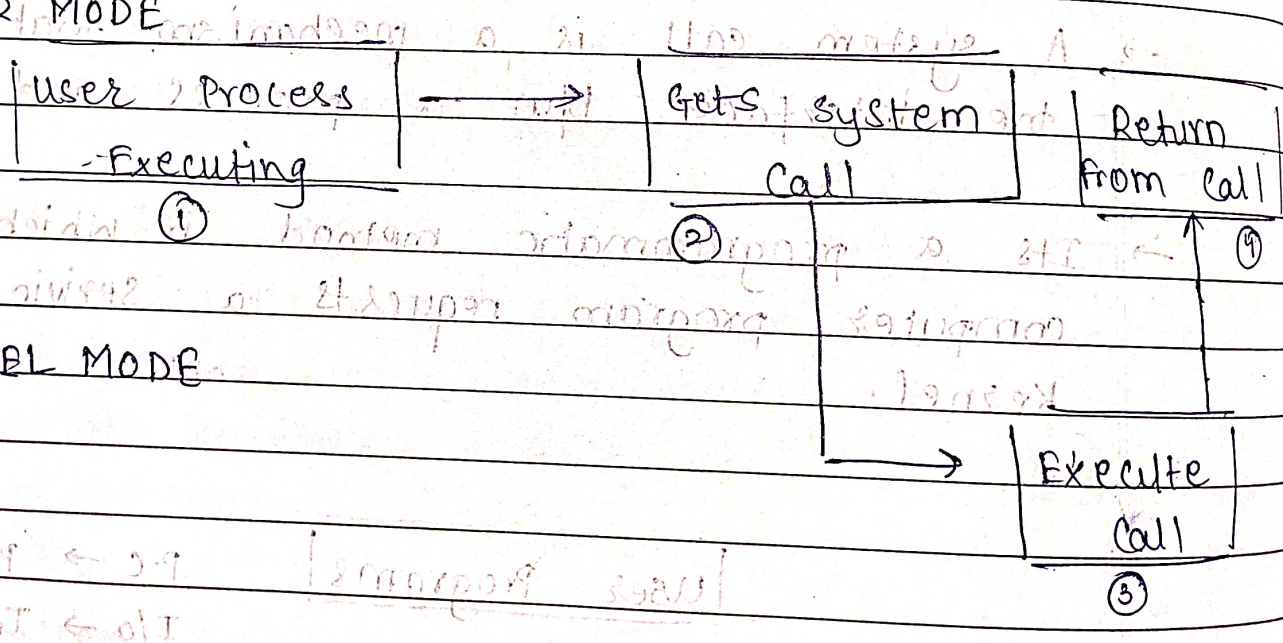
DATE / /

* Examples.

→ If we need to write a program code to read data from one file, copy that data into another file.

→ First info which the program requires is the name of both files.

① First call provides a prompting message.
② Second, to read from the Keyboard, the character which define the two files.



* Use of System Calls

① Reading and writing files
② creation and deletion of files
③ Creation and management of new processes.
④ Accessing hardware devices like scanners

* Types of System calls:-

① Process control :- Performs tasks like process creation and process deletion etc.

Functions

ⓐ End and Abort → { exit() }

ⓑ Load and Execute → { exec() }

ⓒ create process → { fork() }

ⓓ Wait and signal Event → { wait() }

② File Management :- File management system calls handle file manipulation jobs like creating a file, reading and writing, etc.

Functions

ⓐ Create File → touch()
ⓑ Delete File →
ⓒ open and close File → open() / close()
ⓓ Read and Write → read() / write()
ⓔ Get and set → get() / set().

③ Device Management :- used for device manipulation like reading from device, buffers, writing into device buffers etc.

Functions

ⓐ Request and release device
ⓑ Logically attach and detach devices.
ⓒ Get and set device attributes.

④ Information Maintainence :- It handles info and its transfer b/w the O.S and user program

Functions

ⓐ Get / set Date and Time
ⓑ Get process / device attributes

⑤ Communication :- Used to inter process communications

Functions

ⓐ create, delete communication connections
ⓑ send / Recieve Message
ⓒ Help O.S transfer status info
ⓓ Attach / Detach remote devices.

DATE / / /

Part 1.2.

* Types of O.S

① Batch operating System

→ Designed to process a sequence of similar jobs or tasks without manual intervention.

→ Users submit jobs in batches and the operating system executes them one after another.

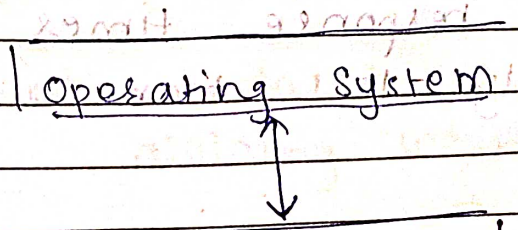
→ Lack interactivity but efficient for handling repetitive tasks.

→ No interaction b/w user and computer

→ Monitor is always in the main memory ready to execute operations.

* Advantages

① No interaction b/w user and computer.
② No mechanism to prioritize tasks.



② Multi-Programming Batch O.S.

→ Single User cannot Keep CPU and I/O devices busy at all times.

→ Multi-programming organizes jobs (code and data) so CPU always has one to exeute

→ A subset of jobs is always kept in the memory

→ One job is selected using job-scheduling

→ As job waits for the resource, such as I/O operation to complete the operating system takes advantage of idle CPU time and exeutes another job.

③ Time - Sharing O.S

→ Rapidly switches the CPU b/w multiple jobs often within a fraction of seconds.

→ Faster response times to user inputs usually less than a second.

→ Each user is given at least one process that is loaded into the memory. This represents user's active session, enabling them to perform tasks.

→ Manages the multiple processes using complex CPU scheduling algorithms. Aims to balance CPU time allocation.

→ If active process exceed physical memory, the swapping technique is used which allows moving processes in and out of the main memory.

→ Implements the concept of virtual memory, allowing execution of processes even when they are not loaded.

(4) Parallel O.S.

→ Utilizes multiple processors in a single computer system to execute tasks concurrently.

→ Executes multiple tasks and threads simultaneously, dividing workload among processors.

→ It is complex due to issues like data synchronization, load balancing etc.

* Distributed O.S.

→ Distributed O.S manages a network of interconnected computers or nodes as a single unified system. It enables users and applications to access resources located across networks.

→ Distributed systems allow resource sharing such as files, printers and processing power, across geographically dispersed machine.

→ Uses communication mechanisms like message-passing and remote procedure calls, are crucial in distributed system.

→ Incorporate fault tolerance mechanisms to ensure system remains operational.

PAGE No.
DATE / /

⑥ Real - time O.S :

→ An RTOS is designed to meet strict timing requirement in applications where response times need to be quick

→ Provides deterministic behavior by guaranteeing that tasks will be completed within the timing constraint.

→ Uses priority - based or time - driven scheduling algorithms to ensure that high - priority tasks are executed before lower - priority ones.

→ In Hard R.T, missing deadlines can be catastrophic, but in soft R.T, missing deadline can degrade the system.

—— x —— x ——

Part 1.3.

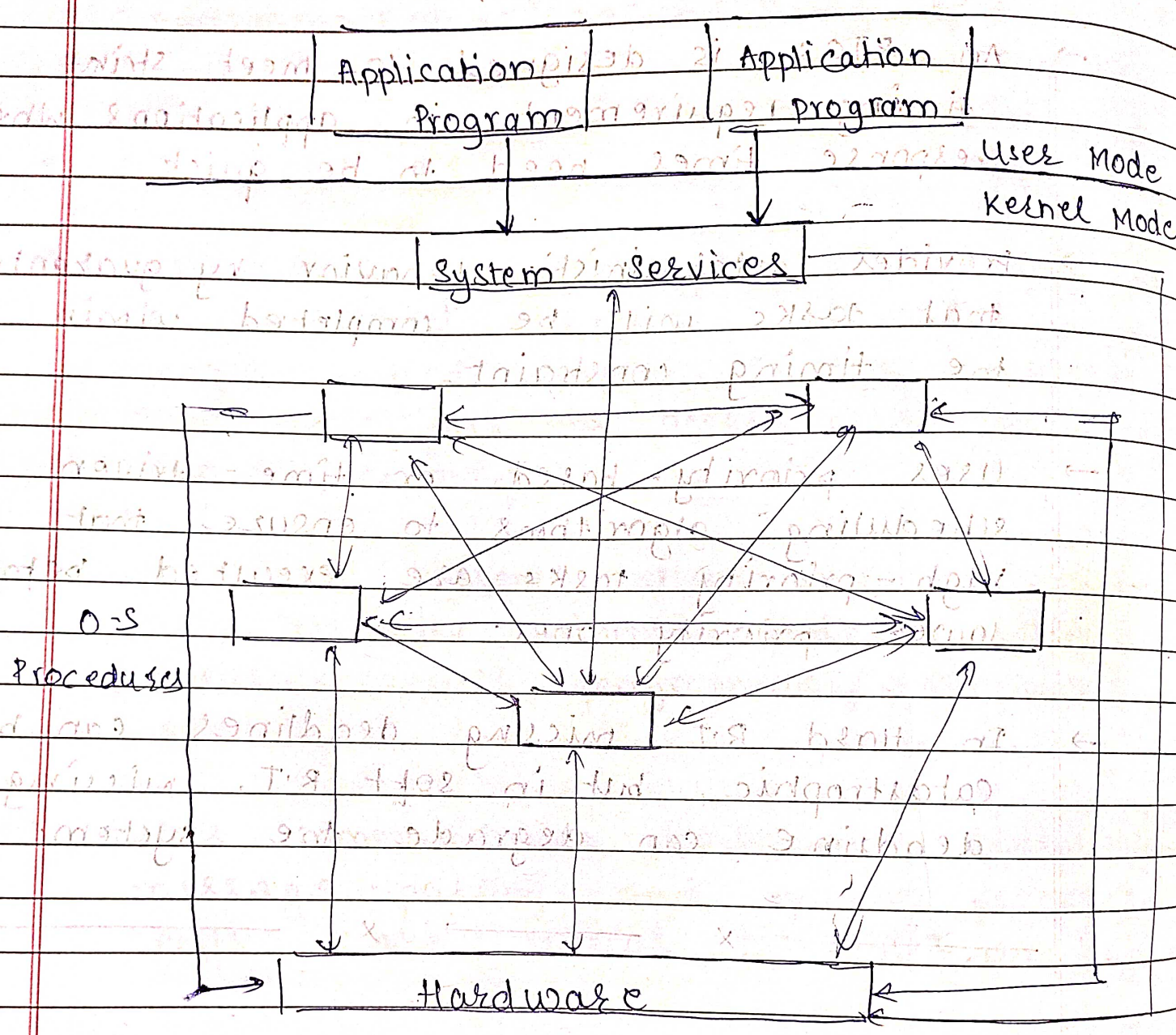
* Structure of Operating Systems

① Monolithic O.S
② Layered O.S
③ Micro - Kernel O.S.

DATE / /

Monolithic

* ~~Overall~~ Structure of O.S



→ Monolithic O.S is characterized by having all ~~of~~ of its components including device drivers, file system and process management, in a single executable binary.

→ Both user-level applications and O.S core share same address space.

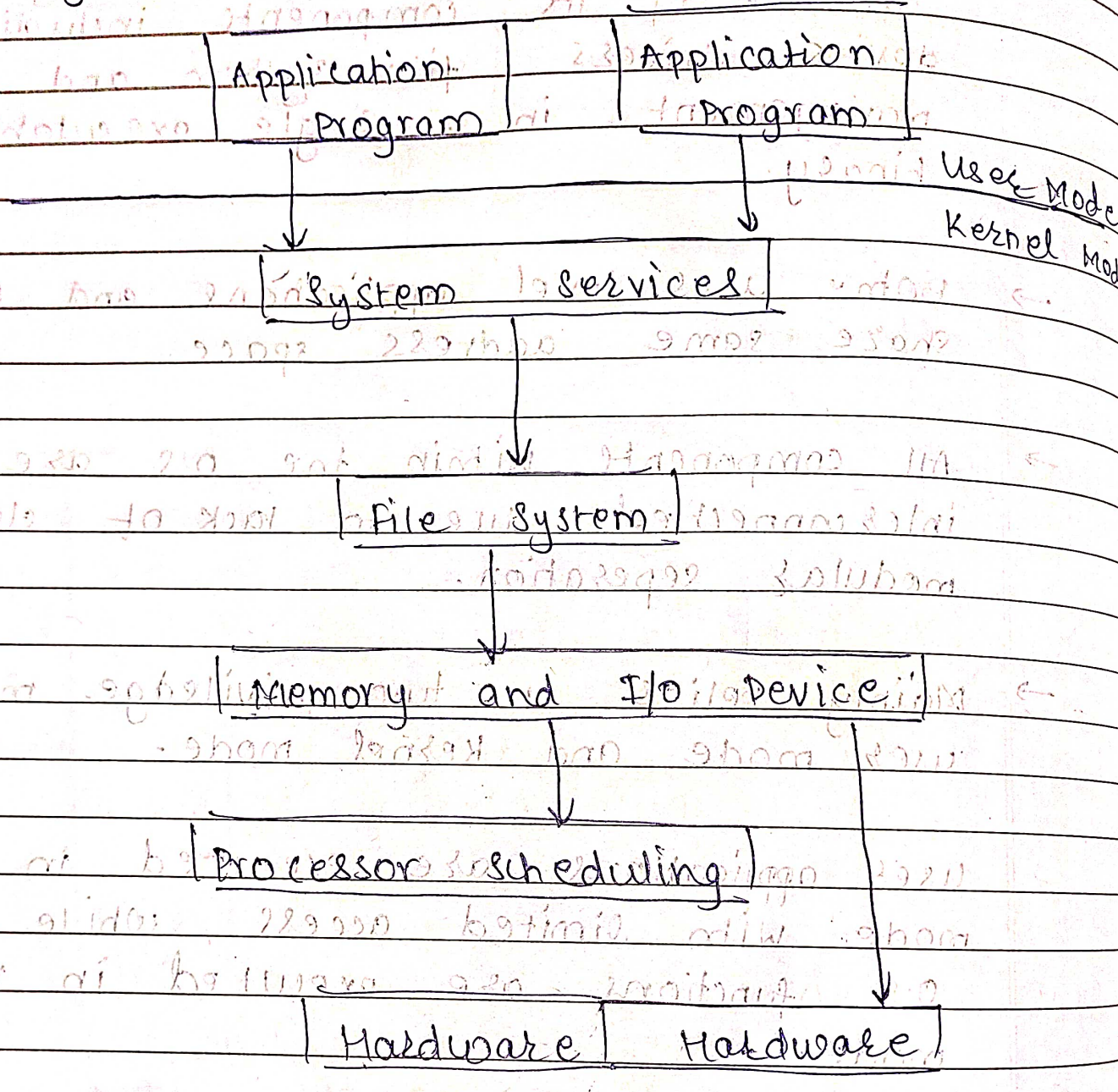
→ All components within the O.S are tightly interconnected due to lack of clear modular seperation.

→ Mainly relies on two priviledge modes, user mode and kernel mode.

User applications are executed in user mode with limited access, while critical O.S functions are executed in kernel mode.

→ Lack of modularity make monolithic O.S efficient, but system grows more complex.

② Layered O.S structure



DATE / /

→ The Layered O.S consists multiple modules which have their own functions and cover their own aspect of system functionality.

→ These layers work together in a coordinated manner to provide an efficient management resources.

Some imp layers are

① Application layer - Here user application and programs run. It's shielded from the complexities of hardware interactions

② services Layer - Provides 3 dedicated layers which abstract the underlying hardware

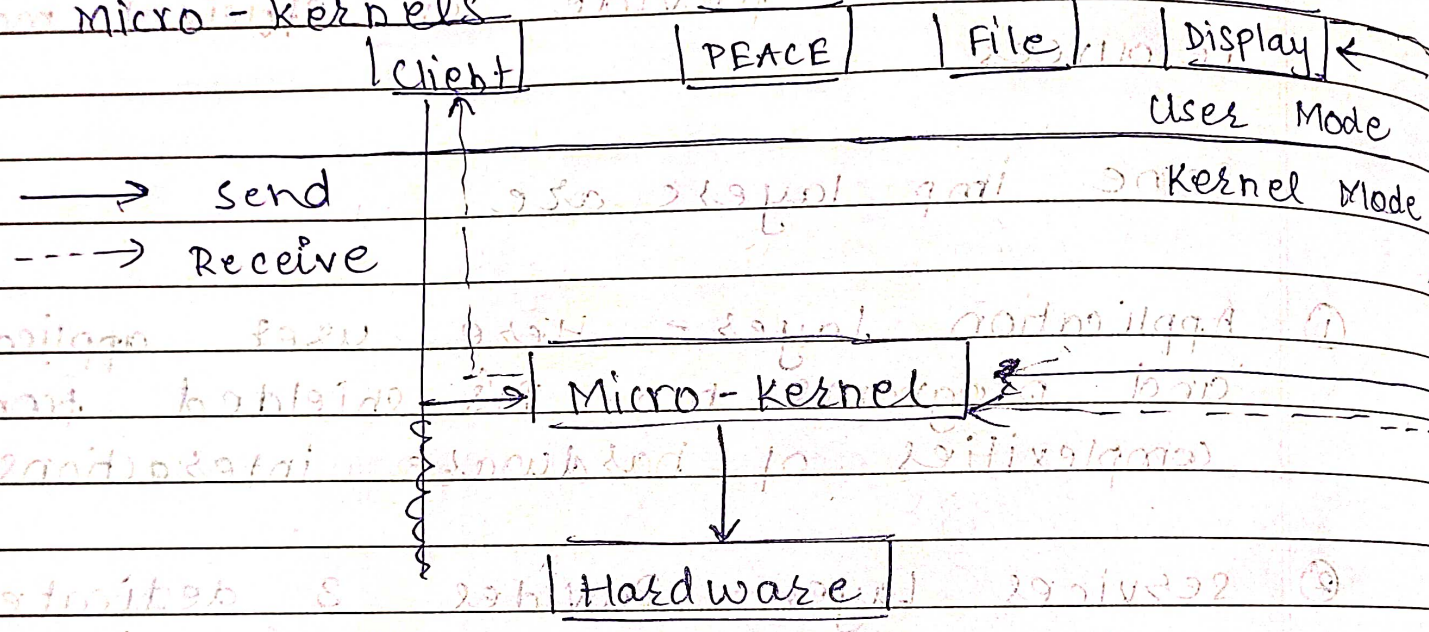
ⓐ ⓑ File System Layer :- Manages data storage and organization. Also provides functions ~~for~~ like read/write/open/close etc. for manipulating files.

ⓑ Memory and I/O layer. - Manages system memory, coordinates i/o operations and provides interface for apps to communicate with hardware.

(c) Process scheduler - Manages creation, execution and termination of all process in the O.S.

---

* Micro-kernels



→ Emphasis on keeping the kernel as small as possible. Only the most fundamental and essential functions are residing in kernel.

→ Most of the functionalities which were in the kernel are moved into the user spaces, processes and modules.

→ Inter-Process communication (IPC) become more crucial in Micro-kernels.

DATE / /

→ Isolates components into seperate address spaces, a failure in one module is less likely to effect others.

* Virtualization and Virtual Machines.

→ Virtualization involves creating a software based environment that mimics a physical machine. Providing exact duplicate of its underlying hardware.

→ Typically operates on user-mode, creating virtual user-mode, creating virtual user mode and virtual kernel mode.

→ Introduces overhead due to abstraction of data b/w virtual machine and physical hardware.

→ Hardware support from the physical processor is crucial. Modern processors provide virtualization extensions to provide assistance.

→ Processors with hardware support can switch between "host" and "guest" mode more efficiently.

* Virtual Machine

→ Embraces a layered approach where it treats both the hardware and the O.S kernel as if they were hardware components.

→ Presents interface to the user which has indistinguishable difference from the underlying physical hardware. So the software running within the VM interacts with the virtualized components.

→ O.S creates an illusion that V.M has its own dedicated resources. From the perspective of software, the V.M is a seperate machine.

Part 1.4.

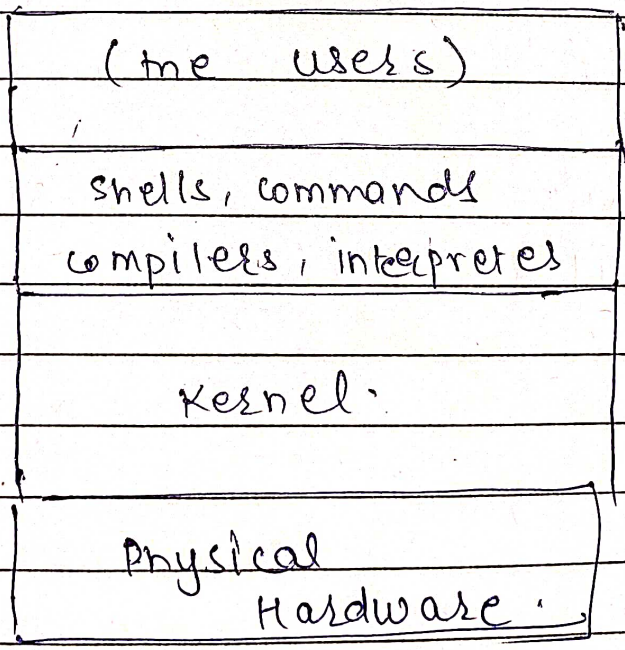
* Modern UNIX systems

UNIX has two seperable parts

① System Programs - Uses-level utilities and applications that provide various functionalities. Such as file manipulation, text editing, compilation, and communication

DATE / /

② The Kernel :- Forms the heart of UNIX O.S. Represents everything below the system-call interface and above the physical hardware.

Kernel functions

① File system
② CPU scheduling
③ Process Management
④ Device Management
⑤ Memory Management.