

27/08/24

9:00 - 9:30

### Morning Activity

- \* How to approach aptitude topics?
- • 40 question 90 min (maybe 60)
- Each question (max 40 sec)
- Accuracy (learn tricks)
- Attempt all the questions. [3 way approach]
- [Go for familiar question to save time]

x — x

\* COS

9:30 to 1:30 pm

Malkeet Singh

1) OS?

- • It manages all the hardware resources of the computer system.
- To manage all the ~~tasks~~ <sup>task</sup> or processes in a computer system.
- It is basically an interface between the user and the computer system (hardware).

2) Diff between OS & Appl<sup>n</sup> software:-

- 1) OS manages hardware resources & provide a platform for application software to run. Application software is designed to specific tasks.

2) OS handles system level tasks like,

• process mgmt • memory mgmt • file mgmt

Application focuses on user centric tasks like creating documents, etc



Abstraction layer: ~~User is not~~  
User is interacting  
with appl<sup>n</sup> on  
this layer.

#### 4) Basic computer organization req<sup>y</sup> for OS.

##### → a) CPU (central processing unit):

- Executes instructions from programs.
- Manages the execution of processes.

##### b) Memory (RAM):

- Temporarily stores data & instructions that the CPU needs while performing tasks.

##### c) Storage (HDD, SSD):

- Permanent storage for
  - Data
  - files
  - OS itself.

##### d) I/O devices:

- peripherals (keyboards, mice, monitor, etc)

##### e) Bus:

- A communication system that transfers data between diff components of computer

#### 5) Example of well known OS:

##### → a) Desktop OS: (Ex: - Windows, macOS, Linux)

- (i) Designed for general purposes on P.C.
- (ii) Provides user friendly & rich interface,
- (iii) Supports wide range of applications.

##### → b) Mobile OS: (Ex: Android, iOS)

- (i) optimized for mobile devices.
- (ii) Focuses: power efficiency, touch interface & mobile connectivity.



Resistor  
Cache  
RAM  
HDD  
Ext

↑ Speed & Cost  
(Bitwise)

Resistor  
Cache  
RAM

↓ High disk space

Volatile memory

## \* Memory Hierarchy

- Two types of main memory

### A] Primary Memory

(i) ROM → PROM (Programmable ROM)  
→ EPROM (Erasable Programmable ROM)  
→ EEPROM (Electrically Erasable Programmable ROM)

(ii) RAM → SRAM (Static RAM)

used in CPU cache

→ DRAM (Dynamic RAM) stores data in form of bit in a capacitor

### B] Secondary Memory

(i) SSD (Solid State drive)

(ii) HDD

(iii) USB pendrive

(iv) Optical device



{ Whenever there is I/O call, shift is observed in user & kernel space }

### 8) User Mode & Kernel Mode

→ (i) User Mode:

- A restricted mode where user application operate
- CPU restricts the operations that can be performed to prevent accidental system damage

(ii) Kernel Mode:

- A privileged mode where the OS operates with unrestricted access to all sys resources.

### 9) Interrupts:

→ (i) Signals to the CPU indicating that an event needs immediate attention.

(ii) They can be:-

- (a) Hardware based (like pressing a key)
- (b) Software based (like a sys call)

• Interrupt Handling:

→ When interrupt occurs, CPU switches to kernel mode.

→ CPU handles

→ kernel mode handles the interrupt & then returns to the previous task.

### 10) System Call:

→ ① Controlled req/ are asked from the user appl<sup>n</sup> to the kernel, which requests it to perform tasks that require higher privileges

[reading a file / creating process]

Execution → ② The appl<sup>n</sup> makes a system call, the CPU switches to kernel mode, the kernel performs the task & then CPU switches back to user mode.



Jana Edition:-  
SE,

{ When  
User

6) Functions of OS:

→ a) Process Management

• (Creates, schedules & terminates processes)

b) Memory Management

• (Allocates & manages memory for processes)

c) File management

• (Manages files on storage devices)

d) Device management

• (Manages I/O devices & drivers)

e) Security & Access Control (Firewall)

• (Protects data & resources from unauthorized access)

f) Networking

• (Provides networking capabilities, allowing system to connect & communicate)

7) User & kernel space:

→ (i) User space: (PCB = Process control board)

• The memory area where user applications run. [Applications have limited access to system resources]

(ii) kernel space:

• The memory area where the OS kernel runs. [It has full access to all hardware & system resources]



BIOS:- Basic/Binary I/O System

BIOS has address of boot loader

→ c) Embedded System OS:

- (i) Used in devices with specific, limited function (ex:- microwaves, routers)
- (ii) Optimized for specific data.

→ d) RTOS (Real time OS): (Ex:- QNX, RTLinux)

- (i) provides precise timing & high reliability for time sensitive applications.  
[Ex: Industrial Automation] (strict time)
- (ii) Guarantees response time within constraints
- Type:- HRT (Hard real time)  
SRT (Soft real time)

→ e) Server OS: (Ex: Windows Server, Ubuntu Server)

- (i) Optimized to manage server resources & serve multiple clients.
- (ii) Focuses: stability, scalability & network services

\* BIOS (Basic/Binary I/O System) / Firmware

→ 1) has address of boot loader

- 2) Before BIOS was loaded on ~~EEPROM~~ ROM (Static)
- 3) Nowadays it is loaded on EEPROM (Dynamic)



3) OS: Directly interact with the hardware & OSs are often tailored to the specific arch. of hardware (Hardware dependent).

Application: relies on the OS to interact with the hardware (Hardware Independent).

### 3) Components of OS:

→ a) kernel → core part of OS, manages sys resources: CPU, memory, I/O devices.  
• → operates in kernel space & complete control over sys.

#### b) Process management:

- → Handles: creation, scheduling & termination of processes
- → Ensures: Multiple processes can run concurrently w/o conflict.

#### c) Memory management: (handles Virtual memory)

- → Manages: System memory, including allocation & deallocation of memory spaces for programs.

#### d) File System:

- → Manages: files on storage devices.
- → Services: file creation, deletion & access control.

#### e) User Interface:

- → GUI (Graphical user interface).
- → CLI (command line interface).