1. Slow learner Activity

1. Activity 1: Match the OS Types with Their Descriptions

Description Cards:

1. Used to run jobs in groups with little or no user interaction.
2. Allows multiple users to use the system at the same time by dividing CPU time.
3. Coordinates multiple computers to appear as a single system.
4. Provides quick and predictable response time, used in critical systems.

Answer Key:

- Batch OS → Description 1
- Time-Sharing OS → Description 2
- Distributed OS → Description 3
- Real-Time OS → Description 4

2. Activity 2: Role Play Cards – OS Components

Instructions: Assign each student a role and act out how a command flows from user to hardware and back.

Roles and Script:

- User: "I want to open a file."
- Shell: "I'll take the user's command and pass it to the system call."
- System Call: "I'm the gateway between user and kernel. I'll pass the request to the kernel."
- Kernel: "I'll manage hardware and send the command to the right device."
- Hardware: "I'll open the file from memory and send it back."

Each student reads their line and passes the command to the next.

3. Activity 3: Fill-in-the-Blanks Worksheet – OS Vocabulary

Instructions: Give students this worksheet and ask them to fill in the blanks using the provided word bank.

Word Bank: Kernel, Shell, User, Hardware, System Call

Worksheet:

1. The _____ is the person who uses the computer.
2. The _____ interprets user commands and communicates with the OS.
3. The _____ is the core of the OS and manages resources.
4. The _____ helps user programs talk to the OS.
5. The OS controls the _____ like CPU, memory, and I/O devices.

Answer Key:

1. User
2. Shell
3. Kernel
4. System Call
5. Hardware

Activity 4: Color and Label OS Architecture Layers

Instructions: Provide students the following labeled blocks and ask them to color each and write one function.

Blocks to Color:

- Application → Blue
- Shell → Green
- Kernel → Yellow
- Hardware → Red

Functions to Write Inside Each Block:

- Application: Sends commands to the system
- Shell: Converts user commands to system calls
- Kernel: Manages memory, CPU, devices
- Hardware: Executes operations requested by OS

You can also ask students to draw arrows showing the flow:
Application → Shell → Kernel → Hardware

2. Activity for moderate learner

1. OS Feature Comparison Table

Objective:

Deepen understanding of popular operating systems by comparing key features.

Activity:

Please fill the table with OS platforms

Feature / OS	Windows 10	Linux Ubuntu	macOS	Android
Open Source?	No	Yes	No	Yes
GUI Available?	Yes	Yes	Yes	Yes
Used In	PCs	Servers/PCs	Apple Devices	Mobiles/Tablets
Package Management	.exe/MSI	apt/dpkg	.dmg	APK
Customizable?	Limited	Highly	Limited	High

Activity 2. Kernel Types Sorting Game

Please place the operating system under the correct kernel architecture.

OS List:

- Windows
- Linux
- macOS
- QNX
- MINIX
- MS-DOS

Categories:

- Monolithic Kernel
- Microkernel
- Hybrid Kernel

Answer Key:

- Monolithic: Linux, MS-DOS
- Microkernel: MINIX, QNX
- Hybrid: Windows, macOS

3. Activity 3: System Call Tracing Task

You are given a user-level task :
"Open a file in a text editor."

please trace the system call path by writing or drawing: You can use the flowchart

1. User Command (e.g., open notepad file.txt)
2. Shell interprets the command
3. System call invoked (e.g., open())
4. Kernel accesses file system

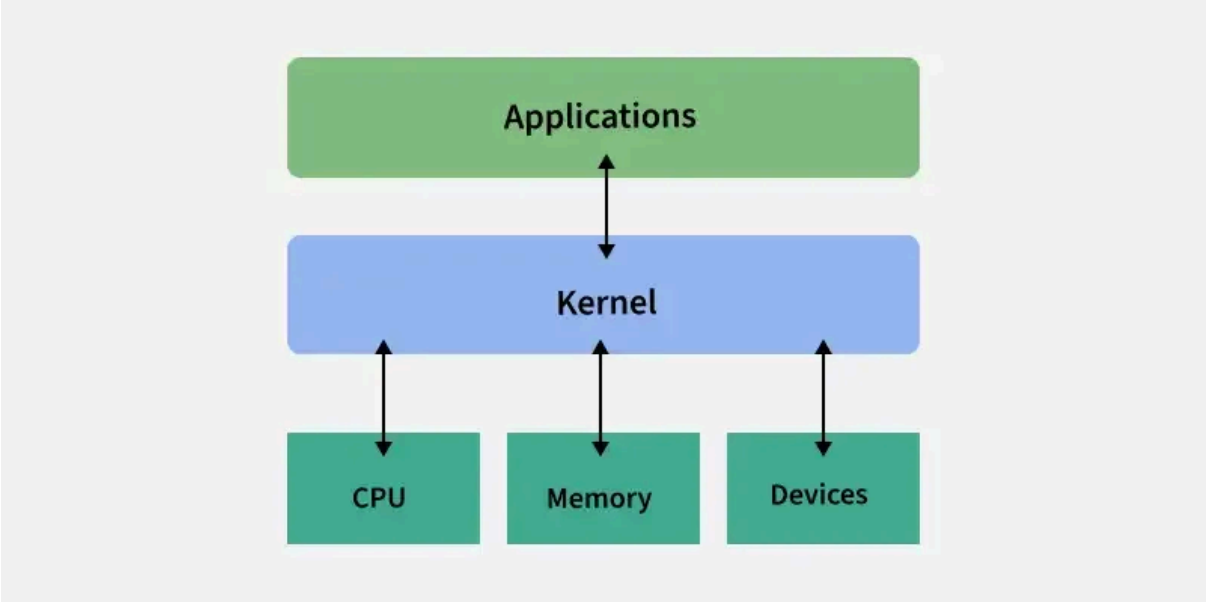
5. File read from disk and passed back

Activity 4. OS Architecture Diagram Completion

OS Architecture Diagram Completion

- Application Layer
- Shell
- System Call Interface
- Kernel
- Hardware

Create the block diagram and label them correctly



3. Fast Learner

1. Industry Use-Case Research & Presentation

Task: In pairs or small groups, choose 1 industry from the list and create a short presentation on how RTOS is used.

Example Industries:

- Automotive (e.g., Anti-lock Braking System)
- Aviation (e.g., Flight Control Systems)
- Medical (e.g., Patient Monitoring Devices)
- Robotics (e.g., Real-time Sensor Fusion)

- Industrial Automation (e.g., Conveyor belt controllers)

Presentation Guidelines:

- What is the real-time requirement?
- Why is RTOS essential?
- Name of the OS used (e.g., VxWorks, FreeRTOS)
- Consequences of OS failure

2. Embedded OS Case Study Writing

Task: Individually, choose a device with an embedded OS and write a short case study.

Sample Devices:

- Smartwatch (e.g., Apple Watch → watchOS)
- Smart TV (e.g., Tizen OS)
- Digital Camera
- Home Assistant Device (e.g., Alexa → Fire OS)
- Fitness Tracker

Content to Include:

- What OS is used?
- What are its limitations & benefits?
- How is it optimized for hardware?
- Why can't a general-purpose OS be used here?

Deliverable: 1-page case study or poster presentation

3. Debate: General-Purpose OS vs Real-Time OS

Task: Divide fast learners into two teams and host a structured debate.

Motion:

> "Real-Time Operating Systems should replace General-Purpose OS in all critical systems."

Team A – *Pro RTOS*: Emphasize reliability, low latency, deterministic behavior
Team B – *Pro GPOS*: Highlight flexibility, multi-user support, UI capabilities