PNS SCHOOL OF ENGINEERING & TECHNOLOGY

NishamaniVihar, Marshaghai, Kendrapara

LECTURE NOTES ON

Operating System

DEPARTMENT OF COMPUTER SCIENCE & ENGG.

4TH SEMESTER

PREPARED BY

Miss Madhusmita Ram

LECTURER IN COMPUTER SCIENCE & ENGG.

Course Content Outline: Operating System

Unit No.	Unit Title	Topics / Sub-Topics	Allotted Time (Hours)
I	**Operating System**	Overview, basic concepts, functions of OS,UNIX/LINUX Architecture, Kernel.	7
II	**Process Management**	Process concepts, operations on processes, IPC, Process Scheduling, Multi-threaded programming, Memory management: Memory allocation, Swapping, Paging, Segmentation, Virtual Memory, various faults, Deadlock:concept of deadlock.	12
III	**File management**	Concept of a file, access methods, directory structure, file system mounting, file sharing and protection, File system structure and implementation, directory implementation, Free space management, efficiency and performance. Different types of file systems.	10
IV	**I/O System**	Mass storage structure - overview, disk structure, disk attachment, disk scheduling algorithms, Swap space management, RAID types.	8
V	**OS Security**	Authentication, Access Control, Access Rights, System Logs	8

Total Duration: 45 Hours

Chapter 1

Operating System (OS)

1. Overview of Operating System

An **Operating System (OS)** is system software that acts as an interface between the **user** and the **computer hardware**. It manages all the hardware and software resources of the computer and provides common services for computer programs.

Examples of Operating Systems

- Windows
- Linux
- UNIX
- macOS
- Android

2. Basic Concepts of Operating System

2.1 User

A **user** is a person who uses the computer system to perform tasks like typing documents, browsing the internet, or running applications.

2.2 Hardware

Hardware includes physical components such as CPU, memory, hard disk, keyboard, mouse, and monitor.

2.3 Software

Software is a set of programs that tells the computer what to do.

System Software: Operating System

Application Software: MS Word, Browser, Media Player

2.4 Process

A **process** is a program that is currently being executed.

2.5 Memory

Memory is used to store data and programs temporarily or permanently.

Primary Memory: RAM

Secondary Memory: Hard Disk, SSD

3. Functions of Operating System

3.1 Process Management

- Creates and deletes processes
- Schedules processes
- Controls execution of programs

3.2 Memory Management

- Allocates memory to programs
- Frees memory after use
- Keeps track of memory usage

3.3 File Management

- Creates, deletes, and modifies files
- Manages directories
- Controls access to files

3.4 Device Management

- Manages input/output devices
- Uses device drivers
- Allocates devices to processes

3.5 Security and Protection

- Protects system data
- Provides password and permission features

3.6 User Interface

Provides interface for user interaction

Types:

Command Line Interface (CLI)

Graphical User Interface (GUI)

4. UNIX / LINUX Architecture

UNIX and Linux follow a layered architecture.

4.1 Architecture Diagram (Conceptual)

User → Shell → Kernel → Hardware

4.2 Components of UNIX / LINUX Architecture

4.2.1 User

End user who gives commands

4.2.2 Shell

Interface between user and kernel

Interprets user commands

Examples: Bash, C Shell

4.2.3 Kernel

Core part of the operating system

Manages system resources

4.2.4 Hardware

Physical components of the system

5. Kernel

5.1 Definition of Kernel

The **Kernel** is the heart of the operating system. It directly interacts with the hardware and manages system resources.

5.2 Functions of Kernel

- Process management
- Memory management
- Device management
- File system management
- System calls handling

5.3 Types of Kernel

5.3.1 Monolithic Kernel

- All services run in kernel space
- Example: UNIX, Linux

5.3.2 Microkernel

- Minimal services in kernel
- Other services run in user space

6. Advantages of Operating System

- Easy to use
- Efficient resource management
- Provides security
- Supports multitasking

Chapter 2

Process Management

2.1 Process Concepts

A **process** is a program that is currently under execution. When a program is loaded into memory and starts running, it becomes a process.

States of a Process

- **New** – Process is being created
- **Ready** – Process is waiting for CPU
- **Running** – Process is being executed
- **Waiting** – Process is waiting for I/O
- **Terminated** – Process has finished execution

2.2 Operations on Processes

The operating system performs the following operations on processes:

Create – New process is created

Execute – Process instructions are executed

Wait – Process waits for an event

Terminate – Process is ended

2.3 Inter-Process Communication (IPC)

IPC is a method that allows processes to communicate with each other.

Methods of IPC

- **Shared Memory** – Processes share a common memory area
- **Message Passing** – Processes exchange messages

IPC is useful in multitasking systems.

2.4 Process Scheduling

Process Scheduling decides which process gets the CPU for execution.

Scheduling Queue

- **Ready Queue** – Processes waiting for CPU
- **Waiting Queue** – Processes waiting for I/O

Scheduling Algorithms (Basic)

- **FCFS (First Come First Serve)** – First process arrives, first executed
- **SJF (Shortest Job First)** – Process with shortest time first
- **Round Robin** – Each process gets fixed time slice

2.5 Multi-Threaded Programming

A **thread** is a small unit of a process.

Advantages of Multithreading

- Better CPU utilization
- Faster execution
- Resource sharing

Example

- Web browser: One thread for loading, one for display

2.6 Memory Management

Memory management is the function of OS that manages primary memory.

Memory Allocation

Memory is allocated to processes when they start execution.

Types

Contiguous Allocation – Memory given in one block

Non-Contiguous Allocation – Memory given in parts

Swapping

Swapping moves processes between main memory and secondary memory.

- Helps to run multiple processes
- Uses swap space on disk

Paging

Paging divides memory into fixed-size blocks called **pages**.

- Logical memory → Pages
- Physical memory → Frames
- Avoids external fragmentation

Segmentation

Segmentation divides memory into variable-sized segments.

Examples:Code segment

Data segment

Stack segment

Virtual Memory

Virtual Memory allows execution of large programs using limited RAM.

- Uses secondary memory as extension of RAM
- Example: Demand paging

Various Faults

Page Fault

- ❖ Occurs when required page is not in main memory.

Segmentation Fault

- ❖ Occurs due to illegal memory access.

Deadlock

Concept of Deadlock

A **deadlock** is a situation where two or more processes wait forever for resources held by each other.

Example:Process P1 holds printer and waits for scanner

Process P2 holds scanner and waits for printer

Both processes are stuck → Deadlock

Necessary Conditions of Deadlock

- Mutual Exclusion
- Hold and Wait
- No Preemption
- Circular Wait

Chapter - 3

File management

1. Concept of a File

A **file** is a collection of related data stored on secondary storage (hard disk, SSD, etc.). It has a name and is used to store information permanently.

File Attributes

- **Name** – Human-readable file name
- **Type** – Text, binary, executable, etc.
- **Size** – Amount of data in the file
- **Location** – Address on storage device
- **Protection** – Read, write, execute permissions
- **Time/Date** – Creation, modification, access time

File Operations

- ✓ Create
- ✓ Open
- ✓ Read
- ✓ Write
- ✓ Close
- ✓ Delete
- ✓ Rename

2. Access Methods

Access method defines how data in a file is accessed.

1. Sequential Access

Data is accessed in order (from beginning to end)

Example: Tape storage, text files

Operations: read next, write next, rewind

2. Direct (Random) Access

Data can be accessed directly using file pointer

Faster than sequential access

Example: Hard disk files

3. Indexed Access

Uses an index to find data quickly

Combination of sequential and direct access

Example: Database systems

3. Directory Structure

A **directory** is a special file that stores information about files.

Types of Directory Structures

1. Single-Level Directory

One directory for all users

Simple but causes name conflicts

2. Two-Level Directory

Separate directory for each user

Avoids name conflict

3. Tree-Structured Directory

Directories can contain sub-directories

Most commonly used

4. Acyclic Graph Directory

Allows sharing of files and directories ,No cycles allowed

4. File System Mounting

Mounting is the process of attaching a file system to the directory tree.

Before mounting, file system is not accessible

After mounting, files can be accessed normally

Example: USB drive mounted to a folder

5. File Sharing and Protection

File Sharing

Allows multiple users to access same file

Controlled by permissions

Useful in networks and multi-user systems

File Protection

Protects files from unauthorized access.

Protection Methods

- **Password protection**
- **Access Control List (ACL)**
- **User permissions** (Read, Write, Execute)

6. File System Structure and Implementation

File System Structure

File system is divided into layers:

1.Application Programs

2.Logical File System

3.File-Organization Module

4.Basic File System

5.I/O Control

6.Storage Devices

File System Implementation

- ✓ Uses data structures like **File Control Block (FCB)**
- ✓ FCB stores file metadata

7. Directory Implementation

Directories are implemented using:

1. Linear List

Simple list of file names and pointers

Slow searching

2. Hash Table

Faster searching

Uses hash function

8. Free Space Management

Keeps track of unused disk space.

Methods of Free Space Management

1. Bit Map (Bit Vector)

Each bit represents a block

0 = free, 1 = allocated

2. Linked List

Free blocks linked together

3. Grouping

First free block stores addresses of other free blocks

4. Counting

Stores start block and number of free blocks

9. Efficiency and Performance

Factors Affecting Performance

- ✓ Disk access time
- ✓ File allocation method
- ✓ Caching and buffering

Techniques to Improve Performance

- ✓ Disk caching
- ✓ Read-ahead
- ✓ Write-behind

10. Different Types of File Systems

Common File Systems

- ✓ **FAT (File Allocation Table)** – Simple, used in USB drives
- ✓ **NTFS** – Windows file system, supports security
- ✓ **EXT4** – Linux file system
- ✓ **HFS+ / APFS** – macOS file systems
- ✓ **ISO 9660** – CD/DVD file system

Chapter-4

I/O System

Mass Storage Structure

Introduction

In an operating system, the Input/Output (I/O) system is responsible for managing communication between the computer and external devices. **Mass storage** is a key part of the I/O system because it provides large-capacity, non-volatile storage for data and programs. Examples include hard disks, solid-state drives (SSD), USB drives, and optical disks.

These lecture notes are designed for **diploma-level students**, with simple explanations and examples.

1. Overview of Mass Storage Structure

What is Mass Storage?

Mass storage devices store large amounts of data permanently (data is not lost when power is off).

Characteristics of Mass Storage

- Non-volatile storage
- Large storage capacity
- Slower than main memory (RAM)
- Low cost per bit

Examples

- Hard Disk Drive (HDD)
- Solid State Drive (SSD)
- Magnetic Tape
- Optical Disks (CD, DVD)

Role of Mass Storage in OS

- Stores operating system files
- Stores user data and applications
- Supports virtual memory (swap space)

2. Disk Structure

A **disk** is a circular storage device coated with magnetic material.

Physical Structure of Disk

- ✓ **Platter**: Circular disk surface where data is stored
- ✓ **Track**: Circular path on a platter
- ✓ **Sector**: Smallest unit of storage on a disk
- ✓ **Cylinder**: Set of tracks at the same position on all platters
- ✓ **Read/Write Head**: Reads or writes data on the disk

Logical Structure of Disk

- ✓ Disk is divided into blocks
- ✓ Each block has a fixed size (e.g., 512 bytes or 4 KB)

- ✓ OS reads and writes data in blocks

Disk Access Time

Disk access time consists of:

1. **Seek Time** – Time to move the head to the required track
2. **Rotational Latency** – Time for the sector to rotate under the head
3. **Transfer Time** – Time to transfer data

3. Disk Attachment

Disk attachment refers to how disks are connected to the computer system.

Types of Disk Attachment

1. Host-Attached Storage

- Disk is directly connected to the computer
- Example: SATA, IDE, USB
- Simple and low cost

2. Network-Attached Storage (NAS)

- Storage device connected to a network
- Accessed using network protocols
- Used for file sharing

3. Storage Area Network (SAN)

- High-speed network connecting storage devices
- Used in large organizations
- Provides block-level access

4. Disk Scheduling Algorithms

Disk scheduling algorithms decide the order in which disk I/O requests are serviced to reduce seek time.

Need for Disk Scheduling

- Multiple processes may request disk access
- Proper scheduling improves performance

Common Disk Scheduling Algorithms

1. First Come First Serve (FCFS)

- Requests are served in the order they arrive
- Simple but may cause long waiting time

Advantages: Easy to implement

Disadvantages: Poor performance

2. Shortest Seek Time First (SSTF)

Serves the request closest to current head position

Advantages: Reduces seek time

Disadvantages: May cause starvation

3. SCAN (Elevator Algorithm)

- Disk head moves in one direction servicing requests
- Reverses direction at the end

Advantages: Better performance than FCFS and SSTF

4. C-SCAN (Circular SCAN)

Head moves in one direction only

After reaching end, returns to start

Advantages: Provides uniform waiting time

5. Swap Space Management

What is Swap Space?

Swap space is a portion of disk used as an extension of main memory (RAM).

Purpose of Swap Space

- Supports virtual memory
- Allows more processes to run than available RAM

Swapping

- Process is moved from RAM to disk (swap out)
- Process is moved back to RAM (swap in)

Swap Space Management Techniques

- **Dedicated swap partition** (faster)
- **Swap file** in the file system (flexible)

6. RAID (Redundant Array of Independent Disks)

RAID combines multiple disks to improve performance and reliability.

Advantages of RAID

- Increased data reliability
- Improved performance
- Fault tolerance

RAID Types

RAID 0 (Striping)

- Data is split across multiple disks
- No redundancy

Advantages: High speed

Disadvantages: No fault tolerance

RAID 1 (Mirroring)

- Same data stored on two disks

Advantages: High reliability

Disadvantages: High cost

RAID 2

- Uses error correction codes
- Rarely used

RAID 3

- Uses parity disk
- Good for large data transfers

RAID 4

- Block-level striping with parity disk

RAID 5

- Block-level striping with distributed parity

Advantages: Good balance of performance and reliability

RAID 6

- ✧ Similar to RAID 5 but with two parity blocks

Advantages: Can tolerate two disk failures

Chapter-5

OS Security

Introduction to OS Security

Operating System (OS) Security is the protection of the operating system from unauthorized access, misuse, damage, or attacks.
It ensures that **data, programs, and system resources** are safe.

Main components of OS Security:

- ✧ Authentication
- ✧ Access Control
- ✧ Access Rights
- ✧ System Logs

1. Authentication

Meaning

Authentication is the process of **checking the identity of a user** before allowing access to the system.

Types of Authentication

✧ **Password-based authentication**

- Username and password
- Most common method

✧ **Token-based authentication**

- Smart cards, USB tokens, OTP (One-Time Password)

✧ **Biometric authentication**

- Fingerprint
- Face recognition
- Iris scan

Examples

- Logging into Windows using a password
- Unlocking a phone using fingerprint
- ATM card with PIN

Importance

- Prevents unauthorized access
- Protects personal and system data

2. Access Control

Meaning

Access Control decides **what an authenticated user can access** in the system.

Types of Access Control

1.Discretionary Access Control (DAC)

- File owner decides who can access the file
- Example: Linux file permissions

2.Mandatory Access Control (MAC)

- OS decides access based on rules
- User cannot change permissions
- Example: SELinux

3.Role-Based Access Control (RBAC)

- Permissions given based on role
- Example:

➢ Admin → full access
➢ Student → limited access

Examples

❖ Only teachers can edit exam files
❖ Students can only read notes

3.Access Rights

Meaning:**Access Rights** define **what actions** a user can perform on a file or resource.

Common Access Rights

- **Read (R)** – View file content
- **Write (W)** – Modify file content
- **Execute (X)** – Run a program
- **Delete** – Remove files

UNIX/Linux Permission Example

rwxr-xr--

User Type	Permission
Owner	Read, Write, Execute
Group	Read, Execute
Others	Read only

Principle of Least Privilege

- Users should get **only required permissions**
- Improves security
- Reduces risk of misuse

4. System Logs

Meaning:System Logs are records of system activities and events.

What is Logged?

- Login and logout details
- Wrong password attempts
- File access

User Type	Permission

- System errors
- Security warnings

Examples

- Linux: `/var/log/auth.log`
- Windows: Event Viewer
- macOS: System log files

Importance of System Logs

Helps detect security attacks

Useful for troubleshooting

Supports investigation after problems

Log Management

- Logs should be protected
- Old logs should be deleted or stored safely
- Important logs should be reviewed regularly

Advantages of OS Security

- Protects data and files
- Prevents unauthorized access
- Improves system reliability
- Helps in monitoring user activity

***