

# File Systems, I/O Management & Security

**Prof. Khushal Bhoyar**

Department of CSE (Cyber Security)

PIET, Parul University

# File System Architecture and Management

## ➤ File Systems

- Provide structured methods for storing, organizing, and accessing data on storage devices.
- Use hierarchical architectures for organization.

## ➤ Logical File System

- Manages **metadata**:
  - File names
  - File attributes
  - Directory structures

➤ **File Organization Module**

- Translates logical file operations to physical block operations.
- **Allocation Methods:**
  1. **Contiguous Allocation**

Stores files in consecutive blocks.

**Advantages:** Fast access.

**Disadvantages:** External fragmentation.
  2. **Linked Allocation**

Uses pointers between blocks.

**Advantages:** No external fragmentation.

**Disadvantages:** Poor direct access.
  3. **Indexed Allocation**

Centralizes pointers in index blocks.

**Advantages:** Efficient direct access.

**Disadvantages:** Extra overhead for index blocks.

## ➤ **Directory Structures**

- Organize files hierarchically.

### ➤ **Types:**

#### **1. Single-Level Directory**

All files stored in one directory.

#### **2. Two-Level Directory**

User directories under a master directory.

#### **3. Tree-Structured Directory**

Flexible hierarchy.

Uses pathnames for file access.

## ➤ **Free Space Management**

- **Purpose:** Track free blocks on storage devices.
- **Techniques:**
  1. **Bit Vector:** 1 bit per block (0 = free, 1 = used).
  2. **Linked List:** Free blocks linked with pointers.
  3. **Grouping:** One block stores addresses of multiple free blocks.
  4. **Counting:** Store start block + number of consecutive free blocks.
- **Use:** Efficient allocation, prevents data overwrite, improves performance.

# Disk Scheduling Algorithms

## ➤ Disk Scheduling Algorithms

- **Purpose:**

- Optimize disk head movement.

- Minimize seek time.

- Maximize throughput.

### 1. FCFS (First-Come-First-Served)

- Serves requests in order of arrival.

- Simple implementation.

- **Problems:**

- Poor performance with random requests.

- Long wait times if requests are scattered.

## 2. SSTF (Shortest Seek Time First)

- Picks request closest to current head position.
- **Pros:**
  - Reduces average seek time.
- **Cons:**
  - Can cause starvation for far-away requests.

## 3. SCAN (Elevator Algorithm)

- Head moves in one direction, serving requests along the way.
- Reverses at disk end.
- **Pros:**
  - Fair service to all requests.
  - Reduces variance in wait times.

#### **4. C-SCAN (Circular SCAN)**

- Moves head in one direction only.
- Jumps back to start after reaching end.
- **Pros:**
  - Uniform wait times.
  - Prevents long waits for edge requests.

#### **5. LOOK & C-LOOK**

- Variants of SCAN and C-SCAN.
- Head stops at last request in current direction, not disk end.
- **Pros:**
  - Reduces unnecessary movement.
  - Faster than SCAN/C-SCAN for sparse requests.



## 6. Anticipatory Scheduling (Modern Systems)

- Pauses briefly to collect nearby requests before moving.
- **Advantages:**
  - Efficient for sequential workloads.
  - Reduces frequent back-and-forth movement.

# File System Security Mechanisms

Control access, protect data confidentiality, and ensure file integrity.

## 1. File Permissions (DAC)

- Control access based on user identity and group membership.
- Use Discretionary Access Control (DAC), allowing owners to set permissions

## 2. Access Control Lists (ACLs)

- Extend basic file permissions.
- Allow finer-grained control for specific users or groups.
- Define multiple permission types for different users/groups.

## 3. Mandatory Access Control (MAC)

- Enforces system-wide security policies.
- Users cannot override these rules.
- Implemented in security frameworks like SELinux and AppArmor.

#### 4. File Encryption

- Protects data confidentiality.
- Types:
  - Full Disk Encryption (FDE)** – encrypts the entire storage volume.
  - File-Based Encryption (FBE)** – encrypts individual files or directories.

#### 5. Secure Deletion

- Ensures erased files cannot be recovered.
- Methods include:
  - Multiple overwrites
  - Cryptographic shredding
  - Physical destruction for highly sensitive data.

## **6. Journaling File Systems**

- Maintain transaction logs of file system operations.
- Ensure metadata consistency after crashes.
- Prevent security issues caused by corrupted file structures.

# I/O System Security

## ➤ **Device Access Control**

- Only allowed programs can use hardware devices.
- Stops unknown or unsafe programs from using I/O devices.

## ➤ **Trusted I/O Paths**

- Data moves safely between devices and applications.
- Prevents others from reading or changing the data in between.

## ➤ **Secure Device Drivers**

- Device drivers are carefully checked and tested.
- Prevents attackers from getting higher system privileges.

## ➤ **DMA Protection**

- Controls how devices access system memory.
- Stops devices from reading or writing unauthorized memory.

➤ **Encrypted I/O Channels**

- Data is encrypted while being transferred.
- Keeps information private and secure.

➤ **Input Validation**

- Checks input coming from devices before using it.
- Prevents malicious commands or attacks from peripherals.

➤ **Overall Protection**

- All these methods protect the system from attacks through hardware devices.

1. <https://www.druva.com/glossary/what-is-a-disaster-recovery-plan-definition-and-related-faqs>
2. <https://www.konverge.co.in/virtualization-in-cloud-computing-need-types-and-importance/>
3. <https://www.crowdstrike.com/cybersecurity-101/cloud-security/cloud-application-security/>

**Parul<sup>®</sup>**  
University

**NAAC**  
GRADE **A++**



<https://paruluniversity.ac.in/>

