

## FILE MANAGEMENT IN OS

### **फ़ाइल क्या होती है?**

- डेटा या जानकारी का संग्रह एक फ़ाइल कहलाता है, जैसे कि डॉक्यूमेंट, फोटो, वीडियो, प्रोग्राम आदि।
- हर फ़ाइल का एक नाम और एक्सटेंशन (जैसे .txt, .mp3, .exe) होता है जो उसके प्रकार को दर्शाता है।

### Type of FILES

#### **1. टेक्स्ट फ़ाइल (Text File)**

- यह सिंपल फ़ाइल होती है जिसमें केवल टेक्स्ट (लिखावट) होता है।
- इसमें कोई फॉर्मेटिंग नहीं होती (जैसे बोल्ड, इमेज या टेबल)।
- इसका एक्सटेंशन .txt होता है।
- इसे नोटपैड, वर्डपैड या किसी टेक्स्ट एडिटर में आसानी से खोला जा सकता है।
- उदाहरण: कोई दस्तावेज़ या लॉग फ़ाइल (notes.txt, data.txt)।

#### **2. सोर्स फ़ाइल (Source File)**

- यह प्रोग्रामिंग कोड वाली फ़ाइल होती है जिसे कंपाइल करके Executable File में बदला जाता है।
- इसे कंपाइलर या इंटरप्रेटर द्वारा पढ़ा जाता है।

- एक्सटेंशन:
  - C/C++ → .c, .cpp
  - Python → .py
  - Java → .java
  - JavaScript → .js

### **3. एकजीक्यूटेबल फ़ाइल (Executable File)**

- यह चलाई जाने वाली (Run करने वाली) फ़ाइल होती है।
- यह आमतौर पर कंपाइल्ड कोड से बनती है।
- ऑपरेटिंग सिस्टम इस फाइल को रन करता है।
- एक्सटेंशन:
  - Windows → .exe
  - Linux → .out, .sh
  - MacOS → .app
- उदाहरण: setup.exe, game.exe, program.out

## **FILES attributes**

### **1. नाम (Name)**

- यह फ़ाइल का विशिष्ट नाम होता है जिससे हम उसे पहचान सकते हैं।
- हर फ़ाइल का नाम यूनिक हो सकता है (उदाहरण: document.txt, video.mp4)।

- नाम का एक्सटेंशन उसकी प्रकार को दर्शाता है, जैसे .jpg, .exe, .py।

## 2. पहचानकर्ता (Identifier)

- हर फ़ाइल का एक विशिष्ट पहचान कोड (Unique Identifier) होता है।
- यह पहचानकर्ता (फ़ाइल आईडी) ऑपरेटिंग सिस्टम द्वारा दी जाती है।
- आमतौर पर इसे फ़ाइल इंडेक्स या इनोड नंबर कहा जाता है (Linux में inode होता है)।

## 3. प्रकार (Type)

- फ़ाइल का प्रकार यह दर्शाता है कि वह टेक्स्ट, इमेज, वीडियो, या प्रोग्रामिंग फ़ाइल है।
- उदाहरण:
- Text File** → .txt, .log
- Image File** → .jpg, .png
- Executable File** → .exe, .sh
- Source Code File** → .cpp, .py

## 4. आकार (Size)

- फ़ाइल कितनी बड़ी है, इसे आकार (Size) कहा जाता है।
- इसे किलोबाइट (KB), मेगाबाइट (MB), गीगाबाइट (GB) में मापा जाता है।

- उदाहरण: image.png = 1.5 MB।

## 5. समय, दिनांक और उपयोगकर्ता पहचान (Time, Date & User Identification)

- हर फ़ाइल का निर्माण (Creation Date), संशोधन (Modification Date) और अंतिम उपयोग (Last Access Date) होता है।
- यह बताता है कि फ़ाइल कब बनाई गई थी, कब बदली गई थी और आखिरी बार कब एक्सेस हुई।
- कुछ ऑपरेटिंग सिस्टम में यह दिखाता है कि कौन उपयोगकर्ता (User ID) उस फ़ाइल का मालिक है।

## FILE ACCESS METHOD IN OS :-

### 1. Sequential Access (अनुक्रमिक एक्सेस)

- डेटा को क्रमशः (एक के बाद एक) पढ़ा या लिखा जाता है।
- यदि किसी फ़ाइल में 100 लाइनें हैं और आपको 50वीं लाइन पढ़नी है, तो पहले 1 से 49 तक पढ़ना पड़ेगा।
- यह तरीका टेप ड्राइव, लॉग फाइल और टेक्स्ट प्रोसेसिंग में उपयोगी होता है।

विशेषताएँ:

- सरल और सीधा तरीका
- क्रमवार संगठित डेटा पढ़ने के लिए आदर्श
- स्टोरेज लागत कम

**कमियाँ:**

- धीमी गति, क्योंकि डेटा को शुरू से अंत तक पढ़ना पड़ता है
- रैम एक्सेस संभव नहीं

**उदाहरण:**

- टेप स्टोरेज (जैसे पुराने बैकअप सिस्टम)
- लॉग फाइलें (जैसे सिस्टम रिकॉर्डिंग)
- टेक्स्ट प्रोसेसिंग (जैसे स्कैनर द्वारा डेटा पढ़ना)

## 2. Direct Access (प्रत्यक्ष एक्सेस)

- इसमें डेटा को किसी भी स्थान से सीधे पढ़ा या लिखा जाता है।
- यदि आपको 50वीं लाइन चाहिए, तो आप सीधे 50वें स्थान पर जा सकते हैं, बिना 1 से 49 पढ़े।
- यह तरीका हार्ड डिस्क, SSD और डेटाबेस सिस्टम में उपयोगी होता है।

**विशेषताएँ:**

- तेजी से डेटा एक्सेस
- रैम एक्सेस संभव
- बड़ी मात्रा में डेटा प्रबंधन

## कमियाँ:

- ✖ संरचना जटिल, क्योंकि हर डेटा का इंडेक्स रखना पड़ता है
- ✖ स्टोरेज लागत अधिक

## उदाहरण:

- हार्ड ड्राइव और SSD (जैसे विंडोज़ और लिनक्स फाइल सिस्टम)
  - डायरेक्ट एक्सेस डेटाबेस (जैसे MySQL, Oracle)
  - मल्टीमीडिया स्टोरेज (जैसे वीडियो फाइलें)
- 
- **Indexed Sequential Access Method (ISAM)** एक फाइल एक्सेस तकनीक है, जिसमें सीक्वेंशियल (क्रमानुसार) एक्सेस और इंडेक्सिंग (सूचीबद्ध करना) का संयोजन होता है। इसका उपयोग डाटाबेस और बड़े डेटा स्टोरेज में किया जाता है।
  - कैसे काम करता है?
  - □इंडेक्स बनाया जाता है – फाइल के महत्वपूर्ण डेटा बिंदुओं का एक इंडेक्स टेबल बनती है।  
▪सीक्वेंशियल पढ़ने की सुविधा – डेटा को क्रमशः पढ़ने या लिखने की सुविधा होती है।  
▪तो जी से एक्सेस – इंडेक्स की मदद से किसी भी रिकॉर्ड को रैंडमली (Randomly) ढूँढ़ा जा सकता है।

**प्रसंशोधन योग्य** – नई एंट्री जोड़ने पर इंडेक्स अपडेट किया जाता है।

## डायरेक्ट्री संरचना (Directory Structure) ऑपरेटिंग सिस्टम में

डायरेक्ट्री संरचना ऑपरेटिंग सिस्टम में फ़ाइलों और फ़ोल्डरों को व्यवस्थित करने का तरीका है। यह डेटा को प्रबंधित (Manage), एक्सेस (Access) और संरक्षित (Secure) करने में मदद करता है।

### ट्री स्ट्रक्चर डायरेक्ट्री (Tree Structure Directory)

ट्री स्ट्रक्चर डायरेक्ट्री ऑपरेटिंग सिस्टम (OS) में फ़ाइलों और फ़ोल्डरों को व्यवस्थित करने का एक पदानुक्रमित (Hierarchical) तरीका है। इसमें मुख्य डायरेक्ट्री (Root Directory) होती है, जिसमें कई सब-डायरेक्ट्री (Subdirectories) होती हैं।

कैसे काम करता है?

#### 1. रूट डायरेक्ट्री (Root Directory):

- यह सबसे ऊपरी स्तर की डायरेक्ट्री होती है।
- इसमें अन्य सब-डायरेक्ट्री और फ़ाइलें मौजूद हो सकती हैं।
- उदाहरण: C:\ (Windows में), / (Linux में)।

#### 2. सब-डायरेक्ट्री (Subdirectory):

- रूट डायरेक्ट्री में कई सब-फोल्डर होते हैं।
- हर सब-डायरेक्ट्री में अपनी अपनी फ़ाइलें और अन्य सब-डायरेक्ट्री हो सकती हैं।

## फ़ाइल सिस्टम संरचना और इसके प्रकार

फ़ाइल सिस्टम (File System) ऑपरेटिंग सिस्टम (OS) द्वारा डेटा को संग्रहित, व्यवस्थित और एक्सेस करने का तरीका है। विभिन्न फ़ाइल सिस्टम संरचनाएँ कंप्यूटर के स्टोरेज को अलग-अलग तरीकों से संभालती हैं।

### 1. FAT (File Allocation Table)

- ✓ पुराने लेकिन लोकप्रिय फ़ाइल सिस्टम
- ✓ Windows और बाहरी स्टोरेज (USB, SD कार्ड) में आम
- ✓ सरल संरचना और कम स्टोरेज उपयोग
- ✗ फ्रैगमेंटेशन समस्या (डेटा बिखर सकता है)
- ✗ सुरक्षा कमज़ोर (कोई एन्क्रिप्शन नहीं)

#### प्रमुख प्रकार:

- **FAT16** – पुराना, केवल **2GB** तक का स्टोरेज सपोर्ट
- **FAT32** – व्यापक उपयोग, **4GB** तक की फ़ाइल सीमा
- **exFAT** – बड़ी फ़ाइलों के लिए अनुकूल, सीमाओं को हटाया गया

### 2. NTFS (New Technology File System)

-  **Windows** का प्राथमिक फ़ाइल सिस्टम
-  बेहतर सुरक्षा और डेटा संरचना
-  लार्ज फ़ाइल सपोर्ट (बिना आकार सीमा)
-  एन्क्रिप्शन, एक्सेस कंट्रोल, और लॉगिंग
-  कुछ बाहरी डिवाइस NTFS को नहीं पहचानते
-  कम्प्लेक्स संरचना, अधिक स्टोरेज उपयोग

## Device management in OS

"डिवाइस मैनेजमेंट ऑपरेटिंग सिस्टम की वह प्रक्रिया है, जो विभिन्न इनपुट, आउटपुट और स्टोरेज डिवाइस को नियंत्रित, मॉनिटर और समन्वित करके उनके प्रभावी उपयोग को सुनिश्चित करती है

यह हार्डवेयर और सॉफ्टवेयर के बीच संचार स्थापित करने, संसाधनों को प्रभावी ढंग से बाँटने, त्रुटियों को संभालने, और डेटा ट्रांसफर की प्रक्रिया को सुचारू बनाने में सहायता करता है।

## Features in Device management

### 1. डिवाइस ड्राइवर मैनेजमेंट

- ऑपरेटिंग सिस्टम हर हार्डवेयर डिवाइस को पहचानता है और उसके लिए उचित ड्राइवर लोड करता है।
- ड्राइवर सॉफ्टवेयर की मदद से हार्डवेयर और OS के बीच सही संचार होता है।

## 2. डिवाइस कम्युनिकेशन

- ऑपरेटिंग सिस्टम इनपुट और आउटपुट डिवाइस को आपस में सुचारू रूप से संचार करने की सुविधा प्रदान करता है।
- डेटा ट्रांसफर के लिए बफरिंग और कैशिंग जैसी तकनीकों का उपयोग किया जाता है।

## 3. डिवाइस शेड्यूलिंग

- जब एक ही समय में कई डिवाइस काम कर रहे होते हैं, तो OS संसाधनों को सही तरीके से आवंटित करता है।
- प्राथमिकता के आधार पर डिवाइस उपयोग को व्यवस्थित करता है।

## 4. एरर डिटेक्शन और हैंडलिंग

- यदि कोई डिवाइस ठीक से काम नहीं कर रहा है, तो OS उसे पहचानकर उपयोगकर्ता को सचेत करता है।
- हार्डवेयर फेल होने या सही से रिस्पॉन्स न देने पर एरर कोड या संदेश दिखाया जाता है।

## 5. डेटा ट्रांसफर और बफरिंग

- जब कोई डिवाइस बड़ी मात्रा में डेटा भेजता है, तो OS उसे बफर (**Buffer**) में स्टोर करके प्रोसेस करता है।
- यह तकनीक डेटा लॉस से बचाती है और ट्रांसफर को तेज बनाती है।

## 6. स्पूलिंग (Spooling)

- जब कई प्रिंट कमांड दिए जाते हैं, तो OS उन्हें कतार (Queue) में संग्रहीत करता है और क्रमवार प्रिंट करता है।
- इससे प्रिंटर को बार-बार लोड नहीं करना पड़ता और कार्य सुचारू रूप से चलता है।

## 7. डिवाइस सिक्योरिटी और एक्सेस कंट्रोल

- OS यह सुनिश्चित करता है कि केवल अधिकृत उपयोगकर्ता ही डिवाइस का उपयोग कर सकें।
- डिवाइस एक्सेस पर प्रतिबंध लगाने या परमिशन सेट करने की सुविधा देता है।

### Types of Devices :-

#### 1) Dedicated Device (समर्पित डिवाइस)

- ये वे डिवाइस होते हैं, जो एक समय में केवल एक ही उपयोगकर्ता या प्रक्रिया द्वारा उपयोग किए जाते हैं।
- ये विशेष रूप से एक कार्य के लिए समर्पित होते हैं और अन्य कार्यों में साझा नहीं किए जाते।
- उदाहरण: प्रिंटर, सीडी/डीवीडी ड्राइव, ग्राफिक्स कार्ड (जब किसी विशेष कार्य के लिए उपयोग किया जाता है)।

#### 2) Shared Device (साझा डिवाइस)

- ये वे डिवाइस होते हैं, जिन्हें कई उपयोगकर्ता या प्रक्रियाएँ एक साथ साझा कर सकते हैं।

- OS इन्हें कुशल तरीके से प्रबंधित करता है ताकि सभी उपयोगकर्ता इनका प्रभावी रूप से उपयोग कर सकें।
- उदाहरण: हार्ड डिस्क, नेटवर्क प्रिंटर, रैम (जब कई प्रोसेसेस इसे साझा करती हैं)।

### 3) Virtual Device (आभासी डिवाइस)

- ये हार्डवेयर डिवाइस नहीं होते बल्कि ऑपरेटिंग सिस्टम द्वारा बनाए गए वर्चुअल (आभासी) संसाधन होते हैं।
- इन्हें वास्तविक हार्डवेयर की तरह कार्य करने के लिए सॉफ्टवेयर द्वारा नियंत्रित किया जाता है।
- उदाहरण: वर्चुअल रैम (स्वैप मेमोरी), वर्चुअल प्रिंटर (PDF प्रिंटिंग), वर्चुअल नेटवर्क डिवाइस।

### 4) Device Drivers (डिवाइस ड्राइवर)

- ये सॉफ्टवेयर प्रोग्राम होते हैं, जो ऑपरेटिंग सिस्टम और हार्डवेयर डिवाइस के बीच संचार स्थापित करते हैं।
- हर हार्डवेयर डिवाइस को काम करने के लिए एक विशेष ड्राइवर की आवश्यकता होती है।
- उदाहरण: ग्राफिक्स ड्राइवर (**NVIDIA, Intel**), प्रिंटर ड्राइवर, साउंड कार्ड ड्राइवर।

## Device Controller

डिवाइस कंट्रोलर एक हार्डवेयर कंपोनेंट है, जो कंप्यूटर के इनपुट और आउटपुट डिवाइस को नियंत्रित करता है। यह डिवाइस को ऑपरेटिंग सिस्टम से जोड़ने और उनके बीच डेटा ट्रांसफर को प्रबंधित करने में मदद करता है।

## डिवाइस कंट्रोलर क्या होता है?

- यह एक इलेक्ट्रॉनिक उपकरण है जो किसी विशेष डिवाइस (जैसे हार्ड डिस्क, कीबोर्ड, प्रिंटर) को नियंत्रित करता है।
- ऑपरेटिंग सिस्टम और हार्डवेयर के बीच मध्यस्थ (interface) की भूमिका निभाता है।
- हर डिवाइस के लिए एक अलग कंट्रोलर होता है, जैसे डिस्क कंट्रोलर, वीडियो कंट्रोलर, प्रिंटर कंट्रोलर आदि।

## डिवाइस कंट्रोलर के प्रमुख कार्य:

1. डेटा ट्रांसफर – डिवाइस और मेमोरी के बीच डेटा को भेजता और प्राप्त करता है।
2. एरर डिटेक्शन – यदि किसी डिवाइस में कोई खराबी या त्रुटि होती है, तो उसे पहचानता और रिपोर्ट करता है।
3. डिवाइस कम्युनिकेशन – OS और हार्डवेयर के बीच सही संचार स्थापित करता है।
4. डिवाइस ड्राइवर लोड करना – ऑपरेटिंग सिस्टम को निर्देश देता है कि कौन सा ड्राइवर लोड करना है।
5. इंटरफेस सपोर्ट – यह कंप्यूटर के बस (Bus) सिस्टम से जुड़कर डाटा ट्रांसफर को नियंत्रित करता है।

## इंटरप्ट (Interrupt)

एक सिग्नल या प्रक्रिया है, जो ऑपरेटिंग सिस्टम को यह सूचित करती है कि उसे किसी महत्वपूर्ण कार्य पर ध्यान देने की आवश्यकता है। जब कोई

हार्डवेयर या सॉफ्टवेयर इवेंट होता है, तो OS को एक इंटरप्ट भेजा जाता है ताकि वह तुरंत कार्वाई कर सके।

## इंटरप्ट क्या होता है?

- इंटरप्ट का मुख्य कार्य CPU को उसकी वर्तमान प्रक्रिया रोकने और किसी अन्य महत्वपूर्ण कार्य को प्राथमिकता देने के लिए मजबूर करना है।
- यह हार्डवेयर और सॉफ्टवेयर दोनों से आ सकता है।
- ऑपरेटिंग सिस्टम इंटरप्ट को हैंडल करने के लिए इंटरप्ट हैंडलर (Interrupt Handler) नामक एक विशेष प्रोग्राम का उपयोग करता है।

डिवाइस एक्सेस मेमोरी (Device Access Memory) का अर्थ यह है कि कंप्यूटर या ऑपरेटिंग सिस्टम किस प्रकार बाहरी डिवाइस (जैसे कीबोर्ड, प्रिंटर, हार्ड ड्राइव आदि) से डेटा को पढ़ता और लिखता है। इसे आसान भाषा में समझें:

### 1. डिवाइस एक्सेस मेमोरी क्या है?

जब कोई डिवाइस कंप्यूटर से जुड़ता है, तो उसे ऑपरेटिंग सिस्टम द्वारा पहचाना जाता है। फिर, वह डिवाइस मेमोरी का उपयोग करके डेटा को स्टोर या प्रोसेस करता है।

### 2. यह कैसे काम करता है?

ऑपरेटिंग सिस्टम डिवाइस को तीन तरह से एक्सेस कर सकता है:

- **सीधे मेमोरी एक्सेस (Direct Memory Access - DMA):** इसमें डिवाइस CPU को बायपास कर सीधे मेमोरी में डेटा लिखता या पढ़ता है। यह तेज़ और प्रभावी होता है।
- **इनपुट/आउटपुट मैपिंग (I/O Mapping):** इसमें हर डिवाइस को एक विशेष एड्रेस दिया जाता है, जिससे CPU उस डिवाइस के डेटा को एक्सेस कर सकता है।
- **बफरिंग और स्पूलिंग (Buffering & Spooling):** अगर कोई डिवाइस धीरे-धीरे डेटा प्रोसेस कर रहा है (जैसे प्रिंटर), तो ऑपरेटिंग सिस्टम एक बफर या स्पूल में डेटा को स्टोर करता है ताकि बाद में भेजा जा सके।