**UNIT-2**

**File**

A file is a named collection of related information that is recorded on secondary storage.

**File concept**

1. **file attributes** - Name, Identification, Type, Location, Size, Access, Protection, Time, Date, and User Identification

2. **file operations** - Create, Write, Read, Remove, Delete, Truncate

3. **file types** - Executable (exe, bin), Object (obj, o), Source Code (c, java), Batch (bat, sh), Text (txt, doc), Library (lib, dll), Font (.ttf, .jpg), Archive (zip, .rar), Multimedia (.mp3, .avi)

**Access methods**

1. **Sequential access** - Information in the file is processed in order, one record after the other.

2. **Direct access** - A file is made up of fixed length logical records that allow programs to read and write records rapidly in no particular order.

3. **Other methods** - Indexing

**Free-space management**

1. **Bit-vector** - If the block is free, the bit is 1, if the block is allocated, the bit is 0. In given example (2, 3, 4, 5, 8, 9, 10, 14, 15 are free and the rest of blocks are allocated) then bit vector is 0011110011100011.

2. **Linked list** - links all the free blocks, disk blocks, keeping a pointer to the first free block in a special location on the disk and caching it in memory.

3. **Grouping** - store addresses of n blocks in 1st free block then first n-1 block are free and nth block contains another n free blocks and so on.

4. **Counting** - keep the address of first free block and the number n of free contiguous blocks that follow the first block.
4. **Allocation Methods** -
   - **Contiguous Allocation** - Each file occupies a set of contiguous blocks on the disk. Both sequential and direct access can be supported. **Disadvantage**: finding free space, fragmentation.
   - **Linked Allocation** - Each file is a linked list of disk blocks. **Disadvantage**: don't support direct access, space required for pointers.
   - **Indexed Allocation** - It brings all the pointers together into one list. **Disadvantage**: it is more complex.

5. **Directory Systems** -
   - The directory can be viewed as a symbol table that translates file names into their directory entries.
   - **Single-level Directory** - All files contain one directory in the same directory.
   - **Two-level Directory** - Create separate directory for each user.
   - **Tree-Hierarchical Directory** -
   - **Acyclic-Graph Directory** - A graph with no cycle. **Disadvantage**: error no cycle.
General graph terminology - have cycle

Protection - To save file from improper access

- Types of Access: Read, Write, Execute, Append, Delete, list, renaming, copying, editing (controlled access)

- Access Control: Access control list → Owner, group & Universe

- Other protection approaches - associate password with each file

File sharing:
- Multiple users - User ID, Group ID

- Remote file systems - Uses networking to allow file system access between systems → manually via programs like FTP, automatically, seamlessly using distributed file systems, semi automatically via the world wide web

File system implementation:
- Hybrid file system

  - Application programs
  - Logical file system
  - File organization module
  - Variable file system
  - I/O control
  - Device

  - File structure
    - File data (write access unit)
    - File owner, group, ACL
    - File size
    - File data, blocks, a pointer to file data block
Dividing Implementation -
- header list of file names with pointers to the data blocks
  - simple to program, time consuming to execute
- Hash table - linear list with hash data structure
  - decreases directory search time, avoids richte, collisions

Disk and drum scheduling -
Improving access time and bandwidth, calculates total head move

E.C.E.S scheduling - first come first serve algorithm
S.S.T.F scheduling - shortest seek time first
S.C.A.N scheduling - start at one end and move towards the other hand
C.S.C.A.N scheduling - circular-SCAN, when the head reaches the other end, however, if immediately returns to the beginning of the disk
L.O.O.K scheduling - same as SCAN but do not get to the end
C.S.L.O.O.K scheduling - same as C.S.C.A.N but do not get to the end

I/O device organization -
- I/O buffering - I/O handshake
- I/O port - typically consists of four registers -

I/O organization -
- Polling - Busy - Waiting
- Interto - CPU interrupt - request triggered by I/O devices
  - Interrupt handler receives interrupts
  - maskable to ignore or delay some interrupts

Direct Memory Access - Used to avoid programmed I/O for large data movement - requires DMA controller, bypasses CPU to transport data directly between I/O device and memory
- Bus - Group of wires and a rigidly defined protocols that specifies a set of messages that can be sent on the wires
- Controller - It is a collection of electronics that can operate a part of a bus or a device
12. Kernel I/O subsystem - (software)
   - Services: scheduling, buffering, caching, interrupting, device
   - Management of name space for files and devices
   - Access control to file and devices
   - File-system space allocation
   - Device driver configuration and initialization
   - Device allocation

13. Transforming I/O request to hardware operations
   - Consider reading a file from disk for a process
   - Determine device holding file
   - Translate name to device representation
   - Physically read data from disk into buffer
   - Make data available to requesting process
   - Return control to process

- Device Driver

14. I/O buffering
   - Buffering is a technique by which the device manager can keep
     I/O devices busy during times when a process is not acquiring I/O
     operations.

   - Types of I/O buffering schemes:
     1. Single buffering - only one buffer
     2. Double buffering - One buffer for device or controller to store
        data, other buffer is to store data from the lower level module
     3. Circular buffering - more than two buffers are used.
     4. No buffering

- Device Driver