

can be accessed.

x An unmounted file s/m is mounted at mount point.

x mount point \Rightarrow where files are added

x Eg: C: | Turbo C

D: | Mount ^{path} C: | Turbo C

File sharing

x Sharing of files on multi user s/ms is desirable.

x Sharing may be done through a protection scheme.

x On distributed systems, files may be shared across a network.

x NFS (Network File s/m) is a common distributed file-sharing method

17/9/21 File Sharing - Multiple Users.

x User IDs - identify users, allowing permissions and protections to be per-user.

x Group IDs - allow users to be in groups, permitting group access rights

File Sharing - Remote File Systems

- Uses networking to allow file s/m access b/w systems.
 - Manually through pgms like FTP
- Client - Server model allow clients to mount remote file systems from servers.
 - Server can serve multiple clients.
 - Standard operating system file calls are translated onto remote calls

File Sharing - Failure Modes.

- Remote file systems add new failure modes, due to network failure, server failure.
- Recovery from failure can involve state info about status of each remote request

Protection

- * File owner / creator should be able to control:
 - what can be done
 - by whom
- * Types of Access
 - Read → Write → Execute
 - Append → Delete → List.
 - ↳ add to last

Access Lists and Groups.

* Mode of access - read, write, execute

* Three classes of users:

			R	W	X
a) Owner Access	7	\Rightarrow	1	1	1
b) Group Access	6	\Rightarrow	1	1	0
c) Public Access	4	\Rightarrow	1	0	0

File System Structure

* File Structure

→ Logical storage unit

→ Collection of related info

* File system resides on secondary storage (disks)

→ Provided user-interface to storage, mapping logical to physical

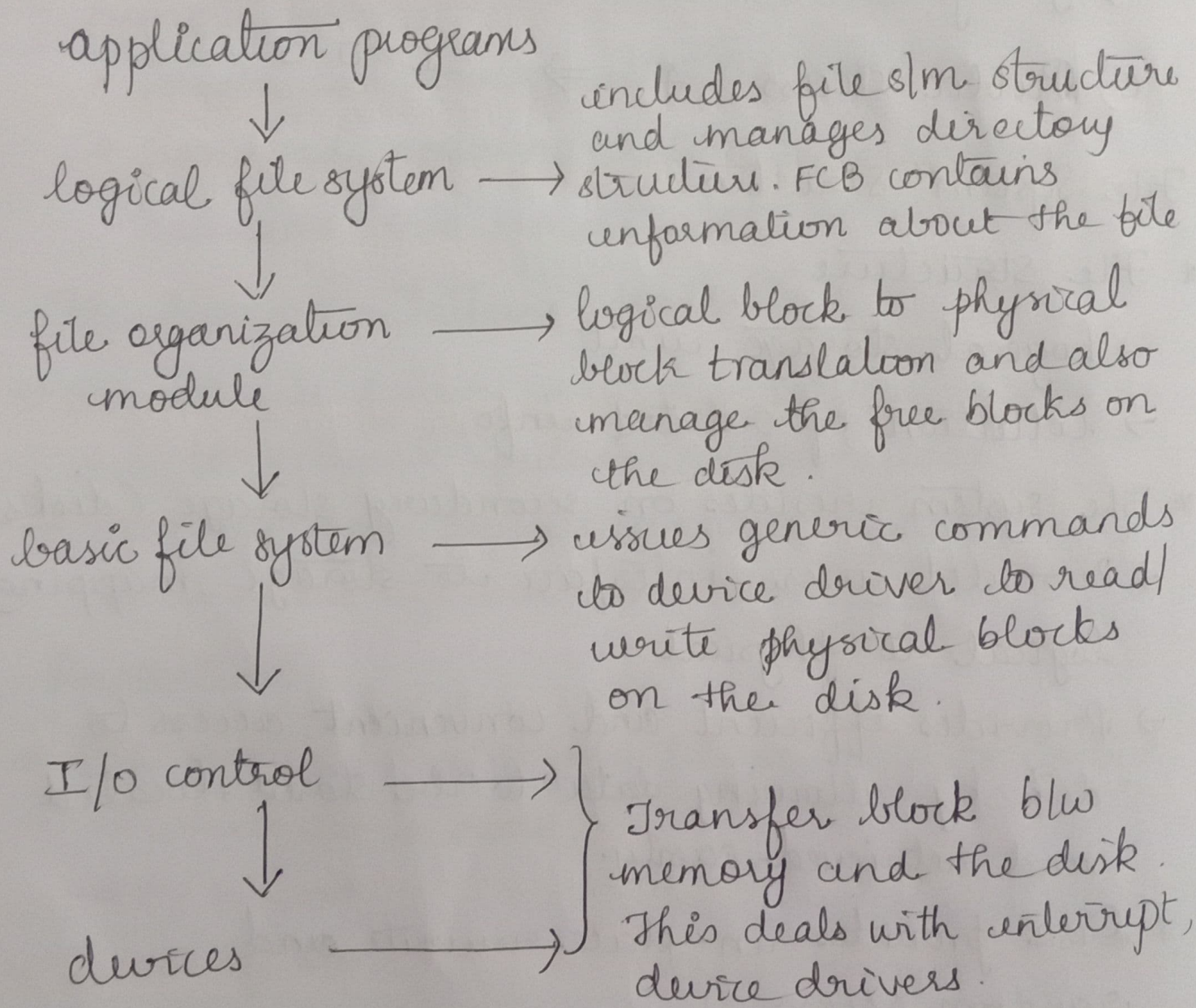
→ Provides efficient and convenient access to disk by allowing data to be stored, located and retrieved easily.

* Disk provides in-place rewrite and random access

→ I/O transfers performed in blocks of sectors (usually 512 bytes)

- * File Control Block - storage structure consisting of information about a file.
- * Device Driver - controls the physical device
- * File system organized into layers

Layered File System



File System Layers

- * Device drivers manage I/O devices at the I/O

control layer.

Given commands like "read drive 1, cylinder 42, track 2, sector 10, unto memory location 1060" outputs low-level hardware specific commands to hardware controller.

Basic File System - given commands translates to device driver.

Also manages memory buffers and caches (allocation, freeing, replacement)

- Buffers hold data in transit.

- caches hold frequently used data.

File Organization Module understands file, logical address and physical blocks.

- Translates logical block number to physical block number.

- Manages free space, disk allocation.

Logical File System manages metadata information

- Translates file name into file number, location by maintaining file control blocks

- Directory Management.

- Protection

Layering useful for reducing complexity & redundancy, but adds overhead and ~~can decrease~~

→ Logical layers can be implemented by any coding method according to os designer

File System Implementation

- 1) On disk
- 2) In memory

I On-disk structure

* Boot-control block contains info needed by system to boot os from that volume.