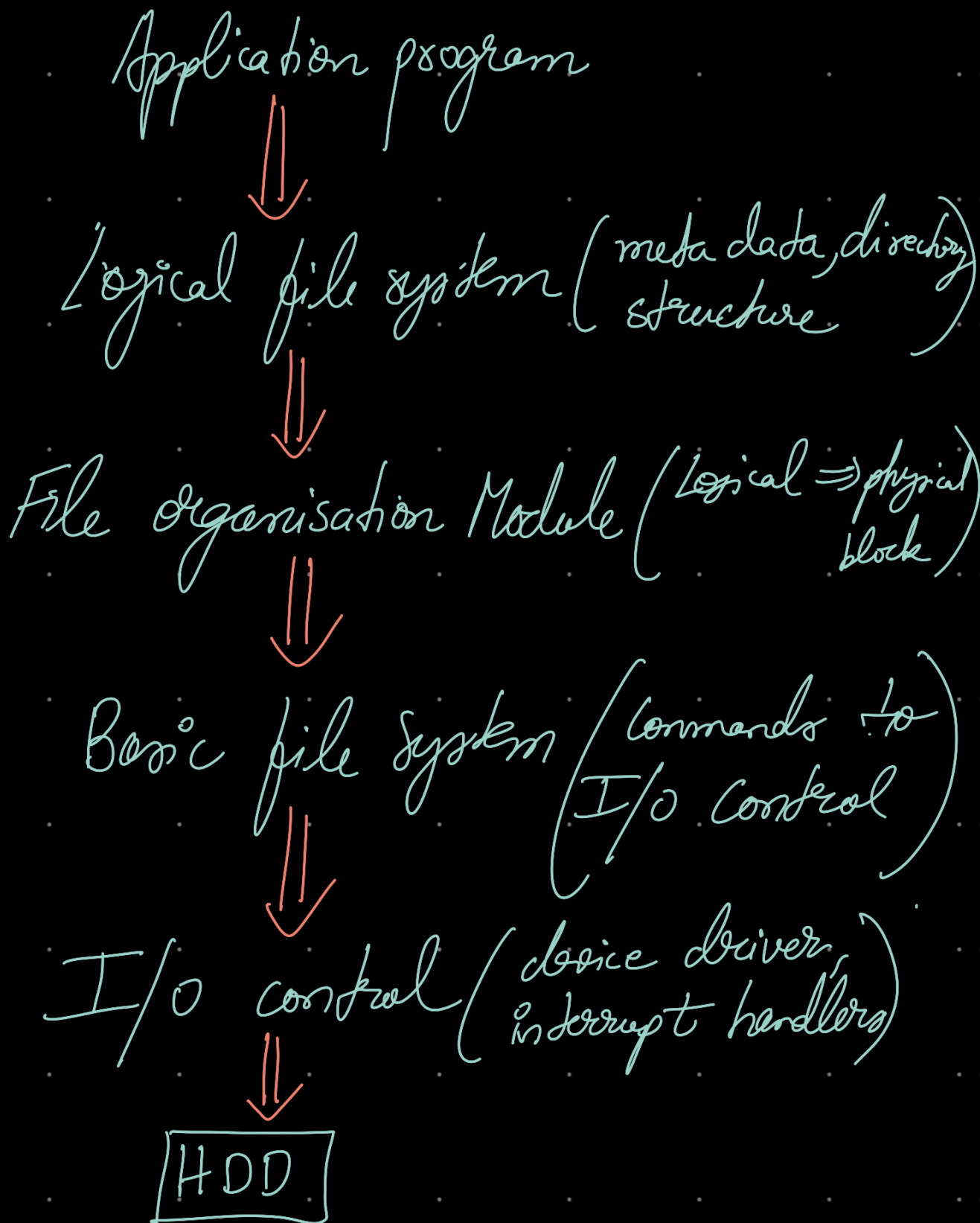
File System Mounting

* File system must be mounted before it can be available to processes.
* Once file system is mounted, system programs can use system calls to do Read/Write on HDD which is abstracted by O.S.
* File System
  - Data structure on Disk
  - Code present inside O.S Kernel
* A disk can be divided into blocks → 1 block = 1 or more sectors

* File system Maps Logical file system to physical storage device.

[Abstraction to access storage device]



Some major file system

Window → FAT32, NTFS

Linux → ext 4, ext 3

* To implement file system :

- Disk Data structure
- RAM data structure

Disk Data structures

- Boot control block — per partition
- Volume control block — partition
- Directory structure — per file system
- File control block — per file

RAM Data structures

→ Buffers

- Mount table
- In-memory directory structure Cache
- System-wide open file table
- Open file table per process

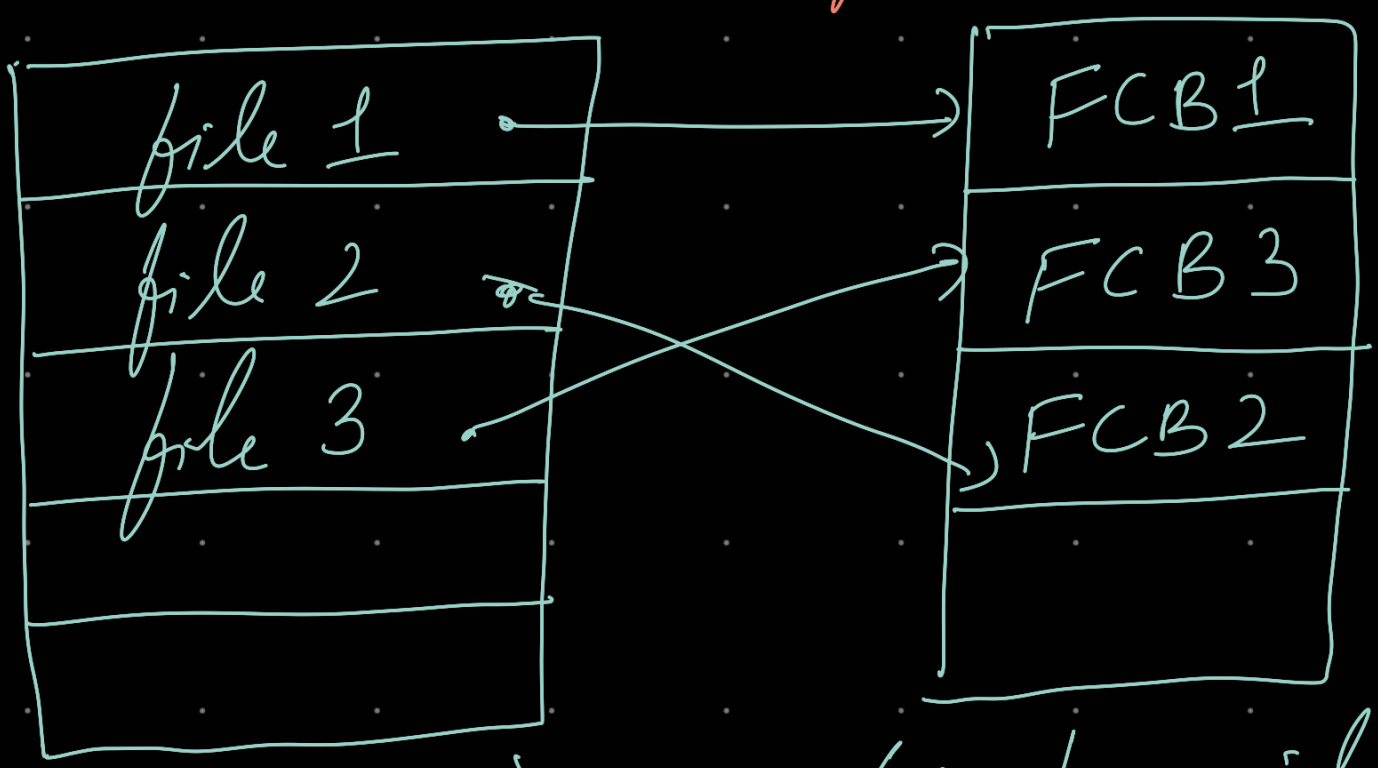
* Mount table → information of each Mounted volume.
  HDD, SSD, USB etc

* In-memory directory structure → Directory info of recently accessed directory.

* System wide open file table → Copy of File control block of every

open file

* Open file table per process → pointer to entry in system wide table

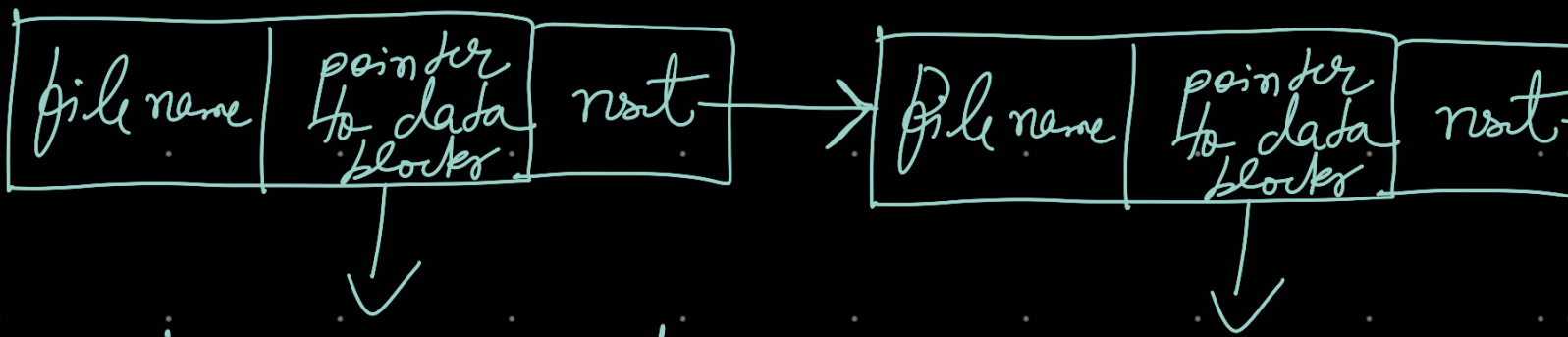


(open file table) (system wide open file table)

* Buffers → Holds file blocks when they are read/write to Disk

What data structure is used to implement directory

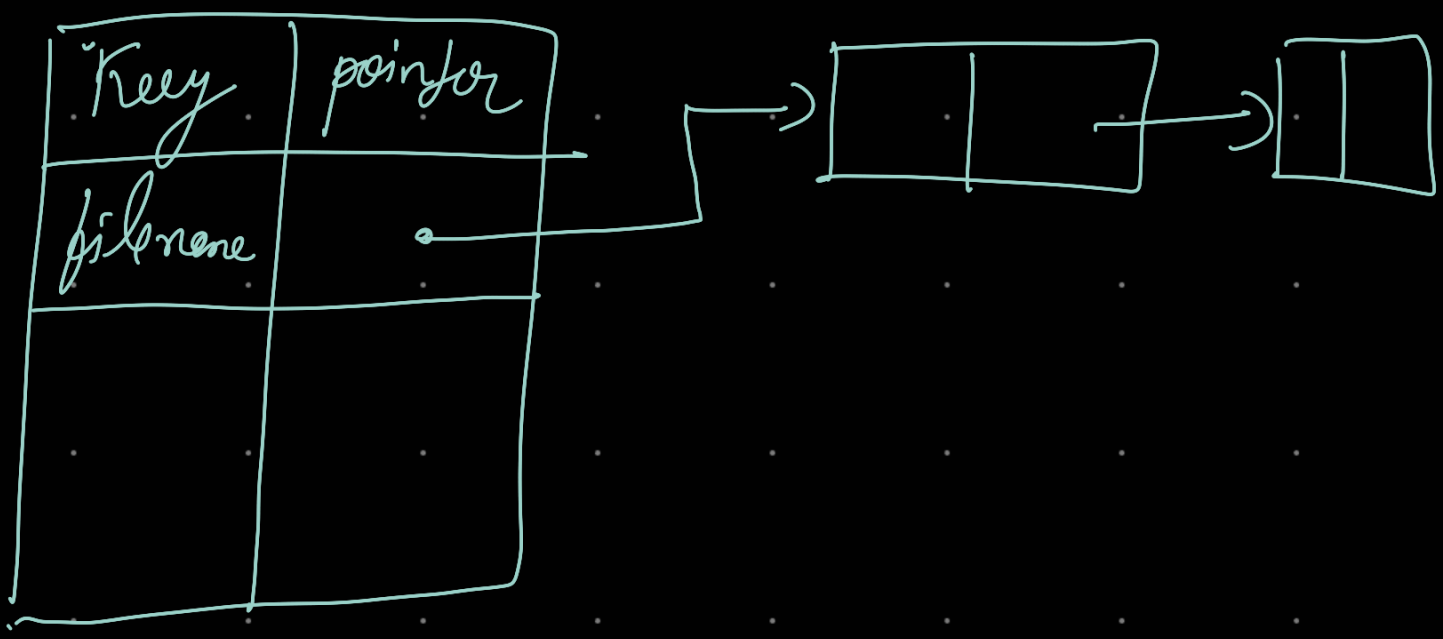
1. Linear List



* Linear Search

Request → search in directory list to find Node → inside node get pointers to data blocks in HDD

2. Hash table



* Search time less
* Chances of collision