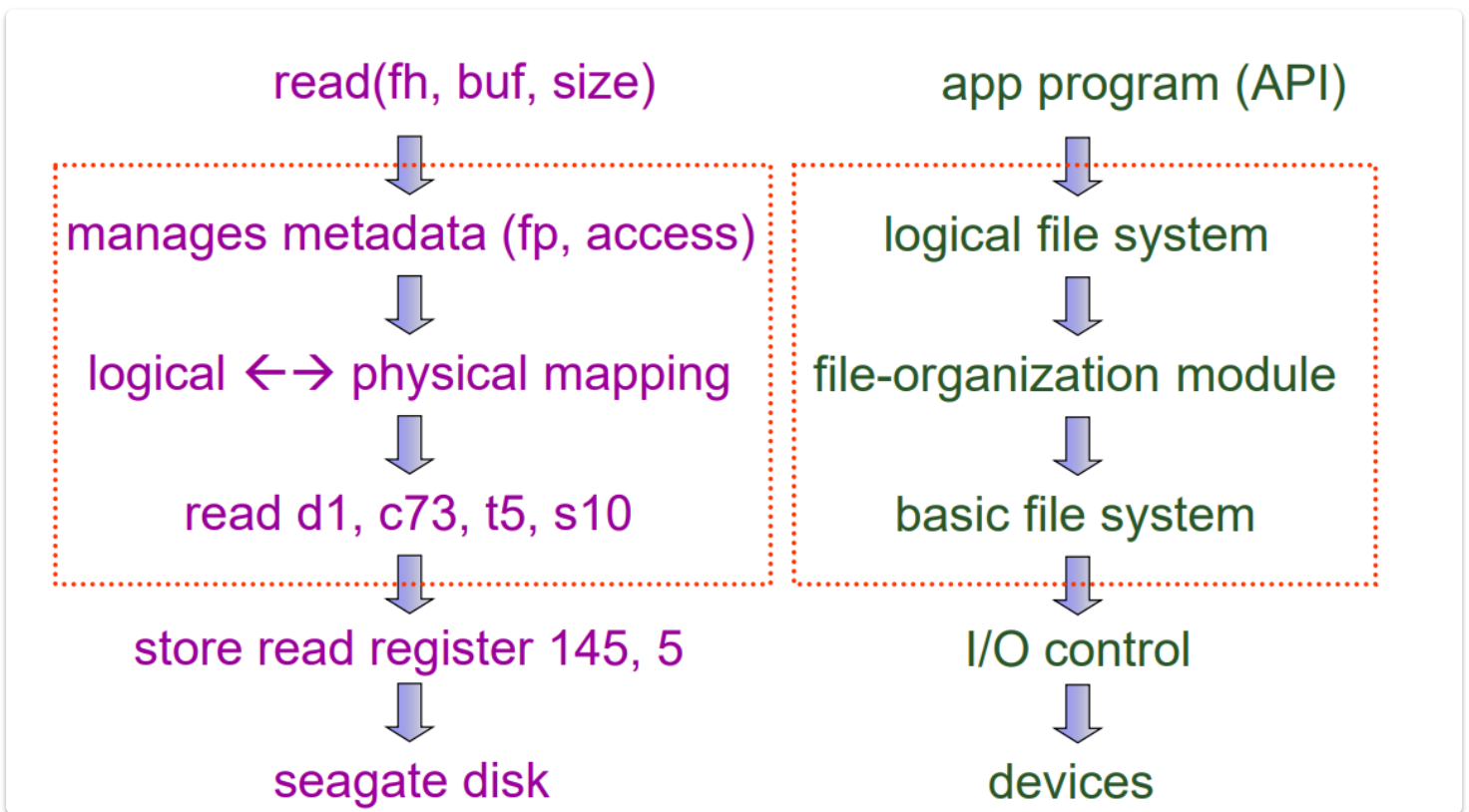


OS_110_CH11

File-System Structure

- 最小的單位就做 **block**
- one block is one or more **sectors**
- format : NTFS, FAT32
- 兩種會遇到的設計問題
 - interface to **user programs**
 - interface to **physical storage (disk)**

Layered File System



有可能在一個 FS manager 需要同時支援多種 FS，像是 NTFS、UFS 之類的，UFS (Unit File Sys.)

File System Implementation

On-Disk Structure

- 必須要存在 Disk 上面，已經在硬碟上面了
- Boot control block
 - 通常會是第一個 partition
 - UFS : boot block, NTFS : partition boot sector
- Partition control block
 - partition 的細節

- UFS : superblock, NTFS : Master File Table
- **File control block (FCB)**
 - 檔案資訊
 - UFS : inode, NTFS : stored in MFT (relational database)
- Directory structure
 - 檔案組織

On-Disk Structure

Partition

Boot Control Block (Optional)
Partition Control Block
List of Directory Control Blocks
List of File Control Blocks
Data Blocks

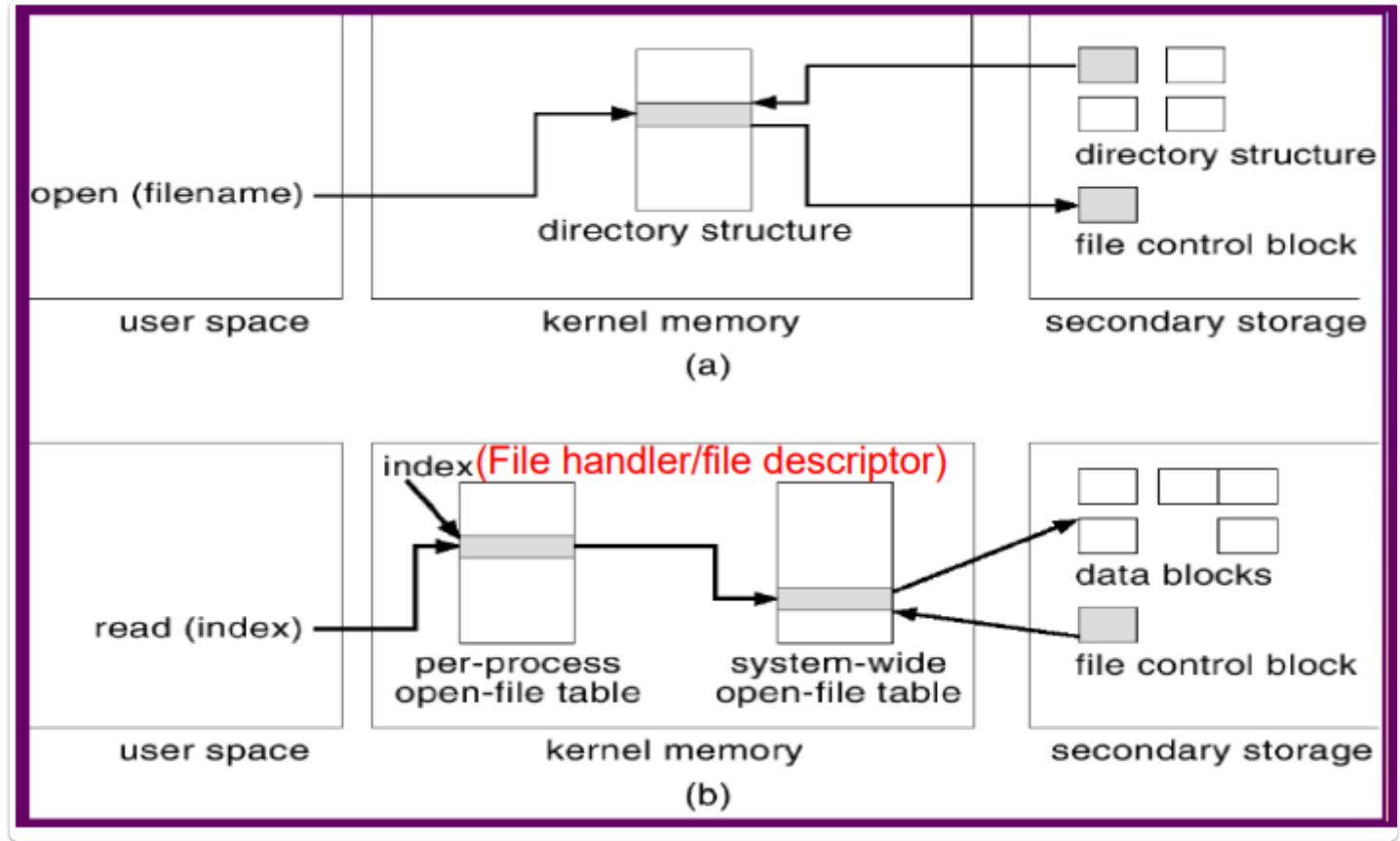
File Control Block (FCB)

file permissions
file dates (create, access, write)
file owner, group, ACL
file size
file data blocks

In-Memory Structure

- in-memory partition table
 - mounted partition 的資訊
- in-memory directory structure
 - recently accessed directories 的資訊
- system-wide open-file table
 - opened file's FCB
- per-process open-file table
 - pointer

File-Open & File-Read



File Creation Procedure

- OS allocates a new **FCB**
- Update **directory structure**
 - 將對應的路徑結構到記憶體
 - 更新結構
 - 寫回硬碟
- The file appears in user's `dir` command

Virtual File System (VFS)

- 他就是一個 interface
- 不同的 File system 可以用同一個 api 進行操作
- Inode - individual file
- File object - in memory
- Superblock - entry file syste
- Dentry object - directory

Directory Implementation

- Linear lists
- Hash table

Allocation Methods

How blocks are allocated for files

- Contiguous allocation
- Linked allocation
- Index allocation

Contiguous

- 無論從哪裡開始，就會從那裡開始連續
- start, length

優

- seek time is the smallest，因為用可以用算地將位置找出，單一一個 I O
- sequential & random access can be implemented efficiently

缺

- External fragmentation 會有空的空間
- File cannot grow

Extent-Based File System

- 一次給他多個 blocks，用完可以再增加 block，且將 block 用鏈結串起來
- Random access become more costly
- Both **internal & external fragmentation** art possible

Linked Allocation

- 將資料的 block 用鏈結串起來

優

- good at sequential access

缺

- random access
- 每一次讀取都是 disk I/O，因為鏈結指標被存在每一個資料塊李勉
- 解決
- cluster of blocks
- 就像是 extent 一樣，還是會有 internal fragmentation
- 只要有一節點壞掉，整個資料就會損毀

FAT (File Allocation Table) file system

是一個 Linked Allocation 的範例

- 將所有 pointer 存在一個 table 裡面直接載入 memory

- 這樣 seek time 也只有 1，就像有一個對應表，是用一個陣列去存的，也解決 random access 的問題
- FAT is often **cache in memory**

Indexed Allocation Example

- 與 FAT 不同，這個是由 Index 直接做 access
- 會有一個 table 可以存