

## Preview

- File System
  - File Name
  - File Structure
  - File Types
  - File Access
  - File Attributes
  - File Operation
- Directories
- Directory Operations
- File System Layout
- Implementing File –How to save file in the disk
  - Contiguous Allocation
  - Linked List Allocation
  - Linked List Allocation with File Allocation Table
  - Index-Node
- Implementing Directories

## File System

Three essential requirements for long term information storage

1. To store a very large amount of information
2. To store information permanently
3. To share the information with multiple processes

## File Name

<file\_name>.<extension>

length: 255

UNIX: case sensitive

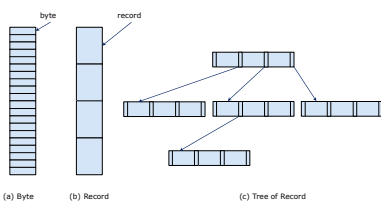
Dos, Window: not case sensitive

## File Structure

Files can be structured

- File is an un-structured sequence of bytes – OS does not know what is in the file (UNIX, Linux, Window)
- File is sequence of records – used until 2<sup>nd</sup> Generation main frame computer – 80 column punched card – files consist of 80 character record.
- File consists of a tree of records

## File Structure



## File Types

- **Regular Files** – for user's information
  - ASCII file – line of text. a line is terminated by carriage return or special character
  - Binary file
    - executable file - OS can execute a file if it has the proper format
    - archive file – consist of collection of library functions compiled but not linked,
    - Data,....
- **Directories** – System files for maintaining the structure of the file system

## Binary Files

---

### Binary File

- ❑ **Executable File**
  - Header – all information regarding execution
    - ❑ Magic number – identifying the file as an executable
    - ❑ Size of files, the address where execution start
  - Text-code itself
  - Data-constant and global variables
- ❑ **Archive File** – a collection of library modules compiled but not linked, or data file

COS450 Operating System, Fall 2024  
Dr. Sang-Eon Park 7

## File Type (Binary Files)

---

COS450 Operating System, Fall 2024  
Dr. Sang-Eon Park 8

## File Access

---

- ❑ **Sequential Access** –
  - a process can read all the bytes in a file in order, starting from the beginning.
  - Read operation change pointer to the next location for read. (magnetic tape)
- ❑ **Random Access** –
  - a process can read all the bytes in a file in any order.
  - Seek operation can move the pointer point to the location (HDD, SSD)

COS450 Operating System, Fall 2024  
Dr. Sang-Eon Park 9

## File Attributes

---

Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write, 1 for read only
Hidden flag	0 for normal, 1 for do not display in listings
System flag	0 for normal files, 1 for system file
Archive flag	0 for has been backed up, 1 for needs to be backed up
ASCII/binary flag	0 for ASCII files, 1 for binary file
Random access flag	0 for sequential access only, 1 for random access
Temporary flag	0 for normal, 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file has last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to

COS450 Operating System, Fall 2024  
Dr. Sang-Eon Park 10

## File Operations

---

- ❑ Create
- ❑ Delete
- ❑ Open
- ❑ Close
- ❑ Read
- ❑ Write
- ❑ Append
- ❑ Seek
- ❑ Get attributes
- ❑ Set attributes
- ❑ Rename

COS450 Operating System, Fall 2024  
Dr. Sang-Eon Park 11

## Directories

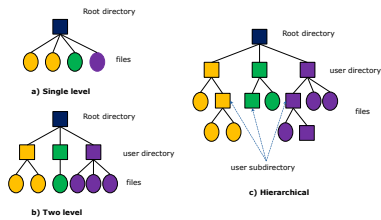
---

To keep track of files, file systems have directories which themselves are files.

- ❑ **Single-level directory System** – one directory containing all files
- ❑ **Two-level directory System** – Giving each user a private directory. User create a file under user's directory. Each file name must be different.
- ❑ **Hierarchical directory System** – User can create a directory to group their files in logical way. Same file name can be used under different directory!

COS450 Operating System, Fall 2024  
Dr. Sang-Eon Park 12

## Directories



CS509 Operating System, Fall 2024  
Dr. Sang-Eon Park

13

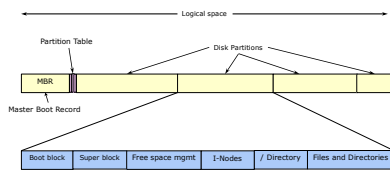
## Directory Operations

- create
- delete
- open directory
- close directory
- read directory
- rename directory
- link directory

CS509 Operating System, Fall 2024  
Dr. Sang-Eon Park

14

## File System Layout (secondary memory)



CS509 Operating System, Fall 2024  
Dr. Sang-Eon Park

15

## File System Layout

- When the Computer is Booted
  - The BIOS reads in and execute MBR.
    - MBR (Master Boot Record) locate the active partition
      - Read the first block (boot block) in a partition and execute it
      - The program in the boot block loads the OS contain in that partition (active partition)

CS509 Operating System, Fall 2024  
Dr. Sang-Eon Park

16

## File System Layout

- The System Layout of a disk partition varies strongly from file system to file system.
  - Super block – key parameters about the file system:
    1. magic number to identify the file system type,
    2. The size of blocks, the number of blocks,...
  - Free space management
  - I –Node
  - Root Directory (/)
  - Directories and Files

CS509 Operating System, Fall 2024  
Dr. Sang-Eon Park

17

## Implementing File

- Disk spaces are divided into blocks and one or more block is used to save a file.
- Implementing a File – How a file is saved in the disk.
  - **Contiguous Allocation**- a file is saved in contiguous blocks.
  - **Linked-List Allocation**- a block is used for saving data and next block information (block number)
  - **Linked-List Allocation with File Allocation Table** – block information (which blocks are used for each files) is saved in FAT.
  - **Index-Node Allocation** – block information for each file (which blocks are used for a file) is saved in a I-node

CS509 Operating System, Fall 2024  
Dr. Sang-Eon Park

18

### Implementing File

- Depends on the way to implement, different information need be saved in the directory for each file.
  - Contiguous Allocation – file name, first block number size (# of block used)
  - Linked Allocation – file name, first block number
  - Linked-List Allocation with File Allocation Table –file name first lock number
  - Index Node Allocation – file name, i-Node Number

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park

### Implementing File (Contiguous Allocation)

Contiguous Allocation- a file is saved in contiguous blocks.

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park

### Implementing File (Contiguous Allocation)

File name	First Block #	# of block used
File A	0	4
File B	4	3
File C	7	8
File E	18	12
File G	36	3
File H	41	4

Directory for files

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park

### Implementing File (Contiguous Allocation)

Two significant advantages with contiguous file allocation

- Simple implementation** – system need keep two numbers for each file in directory: disk address of the first block and the number of blocks used for the file.
- The read performance is excellent** – there is only one seek is needed to read entire file since each block is located contiguously.
  - Seek time – time taken to find the first block location

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park

### Implementing File (Contiguous Allocation)

Seek time: read/write head move to the right cylinder  
Rotation time: read/write head move to the right sector

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park

### Implementing File (Contiguous Allocation: Drawback)

Drawback with the contiguous allocation

- Fragmentation
  - If a newly created file size is fixed, system just finds out a big hole fit to the file. But if a file size is not fixed, there will be problem
  - Ex) A file A is reopened and modified version need more than three block to save!

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park

### Implementing File (Contiguous Allocation: drawback)

- Fragmentation: new file need 7 contiguous block to save, but does not have hole with 7 contiguous block.
  - Solution: Compaction
- File A reopened and size is increased, 4 contiguous blocks are not enough anymore.
  - Solution: find enough holes to move

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park 25

### Implementing File (Linked List Allocation)

Linked-List Allocation- a block is used for saving data and next block information (block number)

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park 26

### Implementing File (Linked List Allocation)

File name	First Block #
Yellow	8
Red	2
Green	3

Directory for files

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park 27

### Implementing File (Linked List Allocation)

**Disadvantage**

- Reading a file sequentially is straightforward, but random access is extremely slow!!
  - To read  $n^{th}$  block, always have to start from the first block.
- Since storage space per a block is not power of 2 anymore, need effort for reading and write in the block!!

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park 28

### Implementing File (Linked List Allocation with File Allocation Table)

- All link information for each of file is written into the File Allocation Table (FAT) which is located in main memory.
- Primary disadvantage
  - Since entire table must be in the memory all the time
    - Ex) 20 GB Hard Drive with 1KB block size, need 20 million entries for the FAT if one entry take 3 bytes, we need 60MB for the FAT.

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park 29

### Implementing File (Linked List Allocation with File Allocation Table)

File name	First Block #
Yellow	8, 5, 1, 0
Red	2, 6, 10
Green	3, 13
...	

File Allocation Table

COSC450 Operating System, Fall 2024  
Dr. Sang-Eon Park 30

### Implementing File (Index-Node)

- I-Node lists the attributes and disk addresses (blocks used to save a file) of the file's block
- To open a file, a system only need load corresponding i-node into the memory!!
- One **problem with i-nodes** is that i-node has room for a fixed number of disk addresses, **what if when a file grows beyond this limit?** - Solution: Reserve the last disk address not for block but for the address of block containing more block addresses.

CS5C450 Operating System, Fall 2024  
Dr. Sang-Eon Park 31

### Implementing File (Index-Node)

i-nodes for each file

CS5C450 Operating System, Fall 2024  
Dr. Sang-Eon Park 32

### Implementing File (Index-Node)

CS5C450 Operating System, Fall 2024  
Dr. Sang-Eon Park 33

### Implementing Directories

Where attributes for each file should be saved?

- MS-DOS or window – stored in a directory entry
- UNIX – stored in the i-node entry

CS5C450 Operating System, Fall 2024  
Dr. Sang-Eon Park 34

### Implementing Directories

How to handle different size of a file name?

- Set a **limit on file name** and stored in a directory entry or in the i-node entry – simple, but wasting space
- All directory entries are start with length of the entry followed by data with fixed format** – when a file is deleted, and created, there is **fragmentation!!!**
- Make the directory entries all fixed length and keep the file names together in a heap at the end of the directory

CS5C450 Operating System, Fall 2024  
Dr. Sang-Eon Park 35

### Implementing Directories

CS5C450 Operating System, Fall 2024  
Dr. Sang-Eon Park 36

## Implementing Directories

---

### Searching Directories

- If a directory is designed linear way, for extremely long directory, linear searching can be slow.
- To speed up the search
  - Use hash table in each directory – more complex administration
  - Use cache