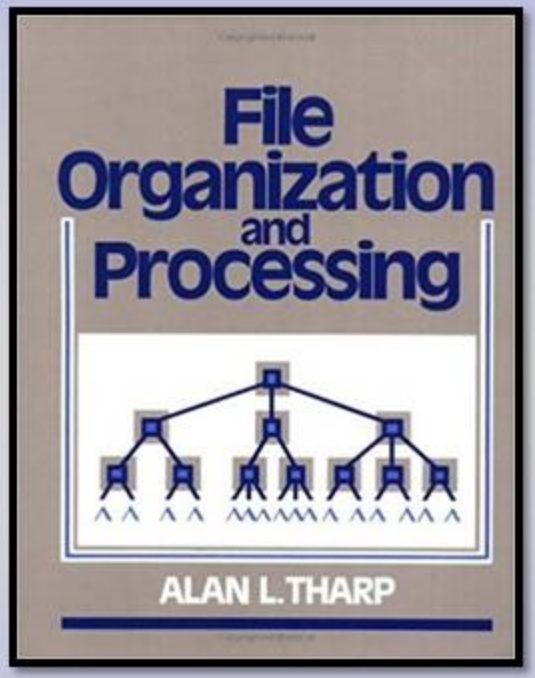
File Organization & Processing

CS2202



Instructor:

Dr. Mohamed Hassan

Textbook (s)

Main textbook,

File Organization and Processing,

Alan L. Tharp,

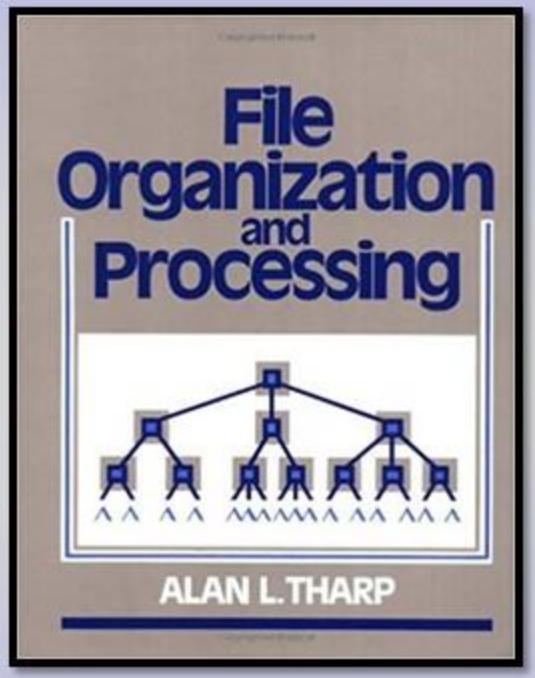
Wiley Edition

Course Structure

- **Work & Grading:**
  - ★ Assignments and quizzes
  - ★ Midterm
  - ★ Final Lab Exam
  - ★ Final

Chapter 1:

- Introduction to file organization and management



Motivation

- Most computers are used for data processing, as a big growth area in the "information age"
- Data processing from a computer science perspective:
  - Storage of data
  - Organization of data
  - Access to data
  - Processing of data

Data Structures vs File Structures

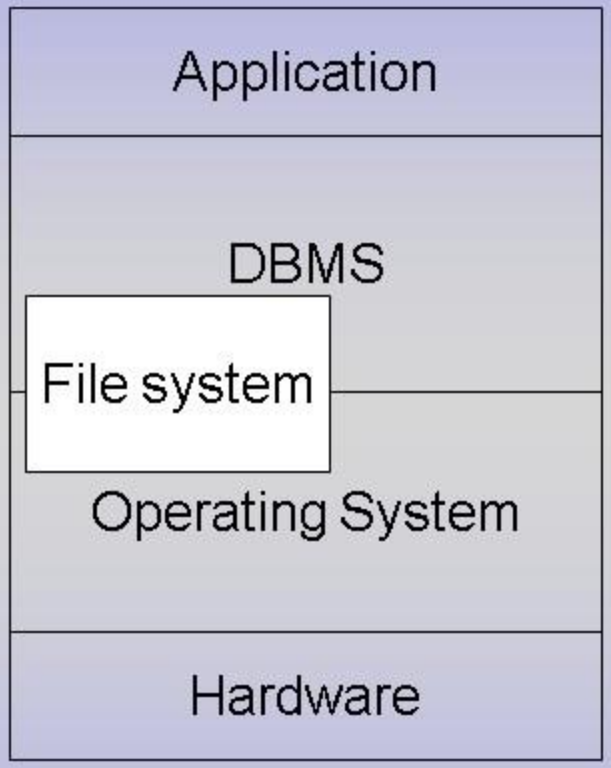
Both involve:

- Representation of Data + Operations for accessing data

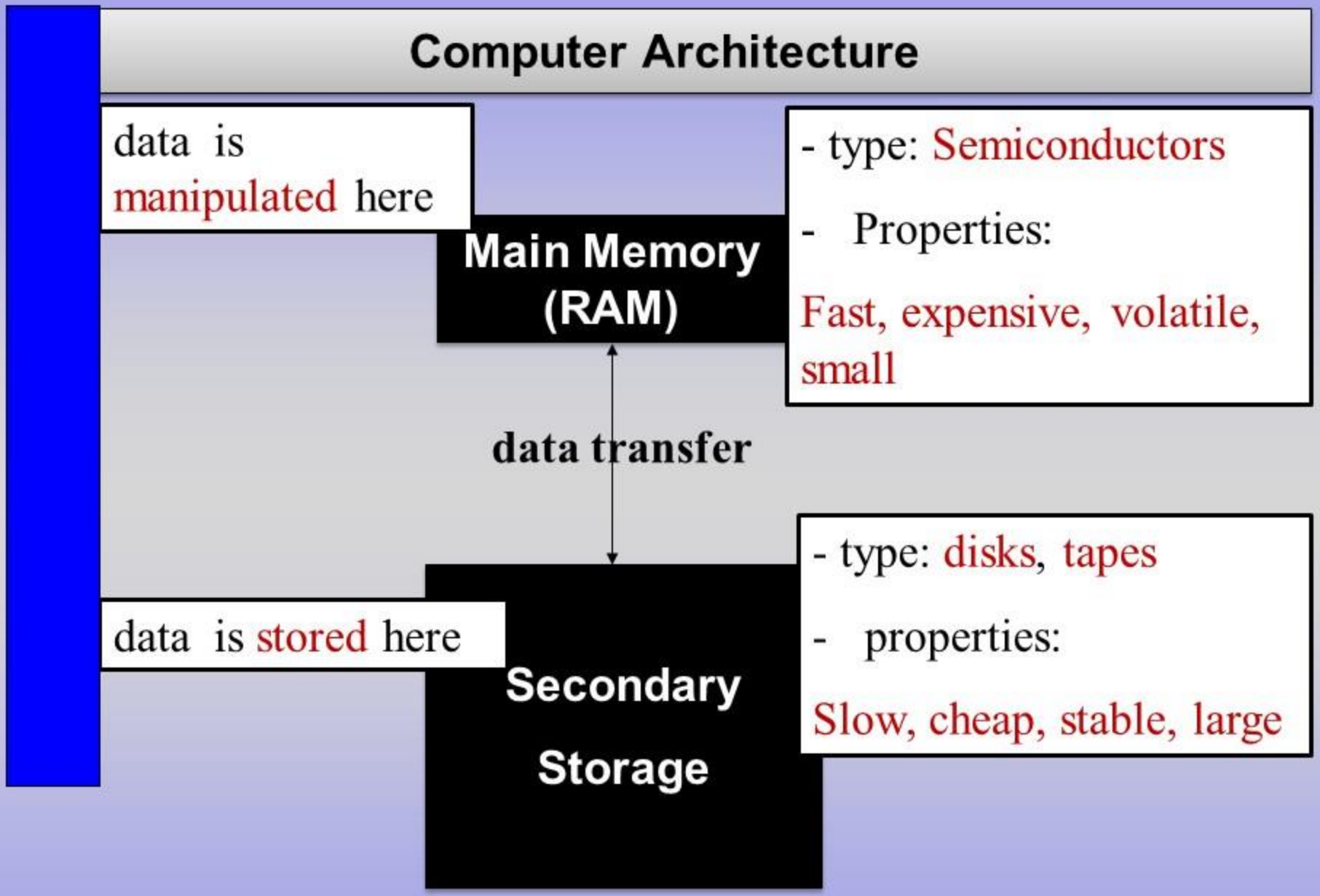
Difference:

- Data structures: deal with data in the main memory
- File structures: deal with the data in the secondary storage

Where do File Structures fit in Computer Science?



Computer Architecture



Main Memory-MM *VS.* Secondary Storage-SS

Main Memory-MM

- fast
- small
- volatile, i.e. data is lost during power failures.

Secondary Storage-SS

- big (because it is cheap)
- stable (non-volatile) i.e. data is not lost during power failures
- slow (10,000 times slower than MM)

Typical time for getting info from:

Main memory: ~10 nanosec = 10×10^{-9} sec

Hard disks: ~10 milisec = 10×10^{-3} sec

Goal of the file structures

- **What is performance**

  Time

  - Minimize the number of trips to the SS in order to get desired information
  - Group related information so that we are likely to get everything we need with fewer trip to the SS.

  Memory

  - Balance the memory size and the time

- **How to improve performance**
  - Use the right file structure
    - Understand the advantages disadvantages of alternative methods

Metrics used to measure efficiency and effectiveness of a File structure-1

- simplicity,
- reliability,
- time complexities,
- space complexities,
- scalability,
- programmability, and
- maintainability.

➢ *Note that the domains of the efficiency and effectiveness concerns rely on time and space complexity more than any other factor.*

Metrics used to measure efficiency and effectiveness of a File structure-2

- The file structures involve two domains:
  - Hardware

    primarily involves the physical characteristics of the storage medium.
  - Software

    Involves the data structures used to manipulate the files and methods or algorithms to deal with these structures.

File operations

- search for a particular data in a file,
- add a certain data item,
- remove a certain item,
- order the data items according to a certain criterion,
- merge of files,
- creation of new files from existing file(s).

finally create, open, and close operations which have implications in the operating system.

File structures versus DBMS

- According to Alan Tharp, "file structures is used to process data in physical level, DBMS is used to manage data in a logical level"
- According to Raghu Ramakrishnan, "DBMS is a piece of software designed to make data maintenance easier, safer, and more reliable".
  - **Thus, file processing is a pre-requisite to DBMSs.**

Physical Files and Logical Files

- **physical file**:

a collection of bytes stored on a disk or tape

- **logical file:**

an “interface" that allows the application programs to access the physical file on the SS

- **The operating system**

is responsible for associating a logical file in a program to a physical file in a SS. Writing to or reading from a file in a program is done through the operating system.

Files

- The physical file has a name, for instance `myfile.txt`
- The logical file has a logical name (a `variable`) inside the program.
- **In C :**

```
FILE * outfile;
```

- **In C++:**

```
fstream outfile;
```

Basic File Processing Operations

- Opening
- Closing
- Reading
- Writing
- Seeking

Opening Files

- **Opening Files:**
  - links a logical file to a physical file.
- **In C:**

```
FILE * outfile;

outfile = fopen("myfile.txt", "w");
```

- **In C++:**

```
fstream outfile;

outfile.open("myfile.txt", ios::out);
```


Closing Files

- Cuts the link between the physical and logical files.
- After closing a file, the logical name is free to be associated to another physical file.
- Closing a file used for output guarantees everything has been written to the physical file. (When the file is closed the leftover from the buffers in the MM is flushed to the file on the SS.)
- **In C :**

```
fclose(outfile);
```

- **In C++ :**

```
outfile.close();
```

Reading

- Read data from a file and place it in a variable inside the program.
- **In C:**

```
char c;
FILE * infile;
infile = fopen("myfile.txt","r");
fread(&c, 1, 1, infile);
```

- **In C++:**

```
char c;
fstream infile;
infile.open("myfile.txt",ios::in);
infile >> c;
```

Writing

- Write data from a variable inside the program into the file.
- **In C:**

```
char c;
FILE * outfile;
outfile = fopen("mynew.txt","w");
fwrite(&c, 1, 1, outfile);
```

- **In C++:**

```
char c;
fstream outfile;
outfile.open("mynew.txt",ios::out);
outfile << c;
```


Seeking

- Used for direct access; an item can be accessed by specifying its position in the file.
- **In C:**

```
fseek(infile,0, 0); // moves to the beginning
fseek(infile, 0, 2); // moves to the end
fseek(infile,-10, 1); //moves 10 bytes from
                      //current position
```

- **In C++:**

```
infile.seekg(0,ios::beg);
infile.seekg(0,ios::end);
infile.seekg(-10,ios::cur);
```


File Systems

- Data is not scattered on disk.
- Instead, it is organized into **files**.
- Files are organized into **records**.
- Records are organized into **fields**.

Example

- A student file may be a collection of student records, one record for each student
- Each student record may have several fields, such as
  - Name
  - Address
  - Student number
  - Gender
  - Age
  - GPA
- Typically, each record in a file has the same fields.

Properties of Files

- **Persistence**:

Data written into a file persists after the program stops, so the data can be used later.

- **Share ability**:

Data stored in files can be shared by many programs and users simultaneously.

- **Size**:

Data files can be very large. Typically, they cannot fit into MM.