INFORMATIKA UMN

UMN UNIVERSITAS MULTIMEDIA NUSANTARA

IF232

ALGORITHMS & DATA STRUCTURES

03

FILE PROCESSING

DENNIS GUNAWAN

REVIEW

Structures, Unions, & Enumerations:

Structures

Unions

Enumerations

OUTLINE

Data Hierarchy

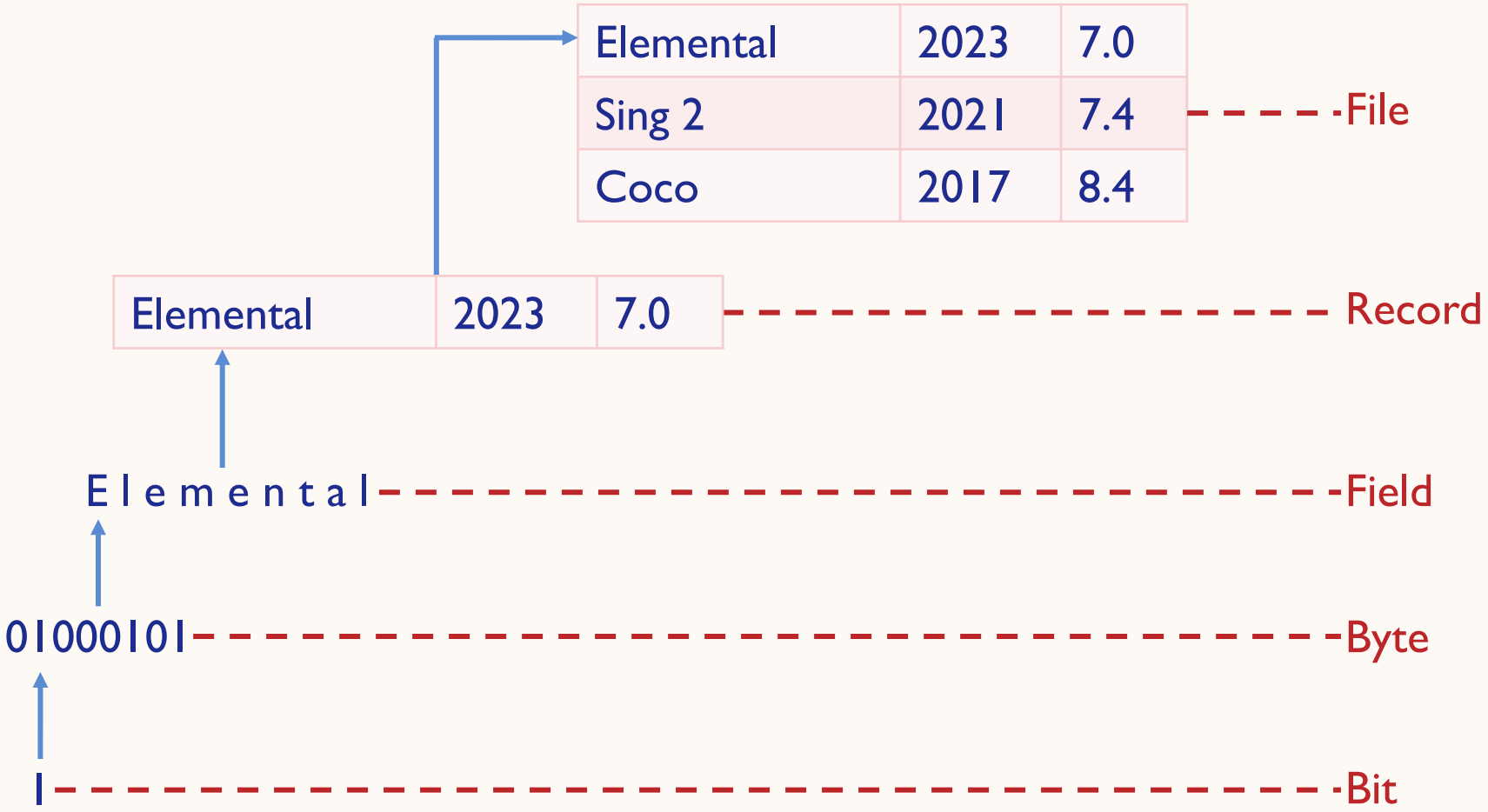
Files and Streams

Sequential-Access File Processing

INTRODUCTION

- Storage of data in variables and arrays is temporary
  - Such data is lost when a program terminates
- Files are used for permanent retention of data
- Computers store files on secondary storage devices, especially disk storage devices

DATA HIERARCHY

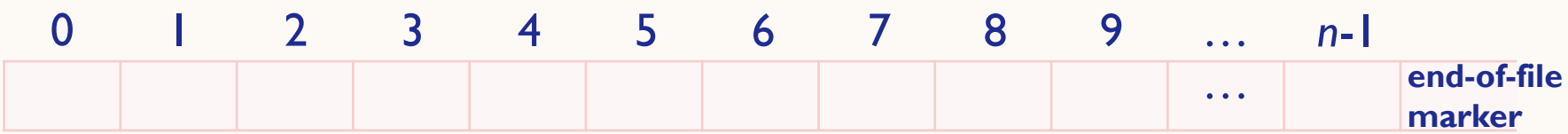


FILES AND STREAMS

- C views each file simply as a sequential stream of bytes
- Each file ends with an end-of-file marker
- When a file is opened, a stream is associated with the file
- Streams provide communication channels between files and programs

FILES AND STREAMS

- C's view of a file of n bytes



OPENING A FILE

- Syntax

```
FILE *fopen(const char *name, const char *mode);
```

  - The fopen() function opens the file with the specified name
  - The second argument is a character string that specifies the requested access mode
  - fopen() returns the FILE pointer for you to use in subsequent input or output operations on the file, or a null pointer if the function fails to open the file with the requested access mode

- Example

```
FILE *fp = fopen("data.txt", "r");
```

OPENING A FILE

- File Access Modes

Mode String	Access Mode	Notes
r	Read	The file must already exist
r+	Read and write	
w	Write	If the file does not exist, fopen() creates it If it does exist, fopen() erases its contents on opening
w+	Write and read	
a	Append	If the file does not exist, fopen() creates it Writing is done at the end of the file
a+	Append and read	

READING DATA FROM A FILE

- Syntax

```
char *fgets(char *buffer, int n, FILE *fp);
```

  - The fgets() function reads a sequence of up to n-1 characters from the file referenced by the FILE pointer argument, and writes it to the buffer indicated by the char pointer argument, appending the string terminator character '\0'

- Example

```
char data[70];
FILE *fp = fopen("data.txt", "r");
fgets(data, 70, fp);
```

data.txt

```
Sing 2#2021#7.4
```

  - If a newline character ('\n') is read, reading stops and the string written to the buffer is terminated after the newline character

READING DATA FROM A FILE

- Syntax

```
int fscanf(FILE *fp, const char *format, ...);
```

  - The fscanf() function is like scanf(), except that it reads input from the file referenced by first argument, fp, rather than from stdin

- Example

```
char title[50];
int year;
float rating;
FILE *fp = fopen("data.txt", "r");
fscanf(fp, "%[^#]#%d#%f", title, &year, &rating);
```

data.txt

```
Sing 2#2021#7.4
```

WRITING DATA TO A FILE

- Syntax

```
int fputs(const char *string, FILE *fp);
```

  - The fputs() function writes a string to the file specified by the FILE pointer argument

- Example

```
char data[70] = “Elemental#2023#7.0”;
FILE *fp = fopen(“data.txt”, “w”);
fputs(data, fp);
```

data.txt

```
Elemental#2023#7.0
```

WRITING DATA TO A FILE

- Syntax

```
int fprintf(FILE *fp, const char *format, ...);
```

  - The fprintf() function is similar to printf(), but writes its output to the stream specified by fp rather than to stdout

- Example

```
char title[50] = “Elemental”;
int year = 2023;
float rating = 7.0;
FILE *fp = fopen(“data.txt”, “w”);
fprintf(fp, “%s#%d#%.1f”, title, year, rating);
```

data.txt

```
Elemental#2023#7.0
```

END-OF-FILE MARKER

- Syntax

```
int feof(FILE *fp);
```

  - The feof() function tests whether the file position indicator of a given file is at the end of the file

END-OF-FILE MARKER

- Example

```
struct movie
{
    char title[50];
    int year;
    float rating;
};
```

data.txt

```
Elemental#2023#7.0
Sing 2#2021#7.4
Coco#2017#8.4
```

```
int i = 0;
struct movie m[3];
FILE *fp = fopen("data.txt", "r");
while(!feof(fp))
{
    fscanf(fp, "%[^#]#%[^#]#%f\n", m[i].title, &m[i].year, &m[i].rating);
    i++;
}
```

CLOSING A FILE

- Syntax

```
int fclose(FILE *fp);
```

  - The fclose() function closes the file associated with a given FILE pointer, and releases the memory occupied by its I/O buffer

- Example

```
fclose(fp);
```

EXAMPLES

```c
#include <stdio.h>

struct movie
{
    char title[50];
    int year;
    float rating;
};

int main()
{
    struct movie m;

    FILE *fin = fopen("data.txt","r");
    fscanf(fin,"%[^#]#%d#%f\n",m.title,&m.year,&m.rating);
    fclose(fin);

    printf("%s (%d): %.1f/10\n",m.title,m.year,m.rating);

    FILE *fout = fopen("output.txt","w");
    fprintf(fout,"%s (%d)\n",m.title,m.year);
    fprintf(fout,"Rating: %.1f/10\n\n",m.rating);
    fclose(fout);

    printf("\n[DG]");
    return 0;
}
```

```
Sing 2 (2021): 7.4/10

[DG]
Process returned 0 (0x0)   execution time : 1.432 s
Press any key to continue.
```

data.txt

File Edit View

```
Sing 2#2021#7.4
```

output.txt

File Edit View

```
Sing 2 (2021)
Rating: 7.4/10
```

EXAMPLES

```c
#include <stdio.h>

struct movie
{
    char title[50];
    int year;
    float rating;
};

int main()
{
    struct movie m[3];
    int totalData = 0, i;

    FILE *fin = fopen("data.txt","r");
    while(!feof(fin)){
        fscanf(fin,"%[^#]#%d#%f\n",m[totalData].title,&m[totalData].year,&m[totalData].rating);
        totalData++;
    }
    fclose(fin);

    for(i = 0;i < totalData;i++){
        printf("%s (%d): %.1f/10\n",m[i].title,m[i].year,m[i].rating);
    }

    FILE *fout = fopen("output.txt","w");
    for(i = 0;i < totalData;i++){
        fprintf(fout,"%s (%d)\n",m[i].title,m[i].year);
        fprintf(fout,"Rating: %.1f/10\n\n",m[i].rating);
    }
    fclose(fout);

    printf("\n[DG]");
    return 0;
}
```

```
Elemental (2023): 7.0/10
Sing 2 (2021): 7.4/10
Coco (2017): 8.4/10

[DG]
Process returned 0 (0x0)   execution time : 1.294 s
Press any key to continue.
```

data.txt

File Edit View

```
Elemental#2023#7.0
Sing 2#2021#7.4
Coco#2017#8.4
```

output.txt

File Edit View

```
Elemental (2023)
Rating: 7.0/10

Sing 2 (2021)
Rating: 7.4/10

Coco (2017)
Rating: 8.4/10
```

PRACTICE

EXERCISES

1. Find the error in each of the following and explain how to correct it.

   a. The file "tools.txt" should be opened to add data to the file without discarding the current data.

   ```
   if((tfPtr = fopen("tools.txt","w")) != NULL)
   ```

   b. The file "courses.txt" should be opened for appending without modifying the current contents of the file.

   ```
   if((cfPtr = fopen("courses.txt","w+")) != NULL)
   ```

EXERCISES

2. Write a single statement to accomplish each of the following.

   a. Open the file `"olddata.txt"` for reading and assign the returned file pointer to `ofPtr`.

   b. Open the file `"transaction.txt"` for reading and assign the returned file pointer to `tfPtr`.

   c. Open the file `"newdata.txt"` for writing (and creation) and assign the returned file pointer to `nfPtr`.

EXERCISES

2. Write a single statement to accomplish each of the following.

   d. Read a record from the file `"olddata.txt"`. The record consists of integer `account`, string `name`, and floating-point `currentBalance`.

   e. Read a record from the file `"transaction.txt"`. The record consists of the integer `account` and floating-point `dollarAmount`.

   f. Write a record to the file "newdata.txt". The record consists of the integer `account`, string `name`, and floating-point `currentBalance`.

EXERCISES

3. Write a program that reads **reviews.txt** and writes **ratings.txt**. Use a structure to create the program.

ratings.txt

```
The Courage to Be Disliked
4.5 out of 5
14651 global ratings

Atomic Habits
4.7 out of 5
122291 global ratings

The Psychology of Money
4.6 out of 5
46523 global ratings
```

reviews.txt

Format: book_title#5star#4star#3star#2star#1star

```
The Courage to Be Disliked#9859#2913#1198#348#333
Atomic Habits#99485#16521#4410#948#927
The Psychology of Money#34232#9069#2305#413#504
```

LAB

EXERCISES

1. Write a program that enables you to list all menus, add menus, order foods, and print bills.

menu.txt

```
Carne Asada Burrito#4.49
French Dip Sandwich#10.49
Monte Cristo Sandwich#8.99
Californian Clam Chowder#12.69
Tuna Tartare#16.00
Avocado Toast#12.00
Huckleberry Pie#7.25
Marionberry Pie#7.50
Hot Fudge Sundae#8.95
```

EXERCISES

1. Write a program that enables you to list all menus, add menus, order foods, and print bills.

```
1. Show menu
2. Add menu
3. Order
4. Show order
5. Print bill

0. Exit

Menu: 1
```

```
No. Menu                          Price

1.  Carne Asada Burrito            4.49
2.  French Dip Sandwich           10.49
3.  Monte Cristo Sandwich          8.99
4.  Californian Clam Chowder      12.69
5.  Tuna Tartare                  16.00
6.  Avocado Toast                 12.00
7.  Huckleberry Pie                7.25
8.  Marionberry Pie                7.50
9.  Hot Fudge Sundae               8.95
```

EXERCISES

1. Write a program that enables you to list all menus, add menus, order foods, and print bills.

```
1. Show menu
2. Add menu
3. Order
4. Show order
5. Print bill

0. Exit

Menu: 2
```

```
Menu : Shrimp Cocktail
Price: 9.00
```

EXERCISES

1. Write a program that enables you to list all menus, add menus, order foods, and print bills.

```
1. Show menu
2. Add menu
3. Order
4. Show order
5. Print bill

0. Exit

Menu: 1
```

```
No. Menu                          Price

1.  Carne Asada Burrito            4.49
2.  French Dip Sandwich           10.49
3.  Monte Cristo Sandwich          8.99
4.  Californian Clam Chowder      12.69
5.  Tuna Tartare                  16.00
6.  Avocado Toast                 12.00
7.  Huckleberry Pie                7.25
8.  Marionberry Pie                7.50
9.  Hot Fudge Sundae               8.95
10. Shrimp Cocktail                9.00
```

EXERCISES

1. Write a program that enables you to list all menus, add menus, order foods, and print bills.

```
1. Show menu
2. Add menu
3. Order
4. Show order
5. Print bill

0. Exit

Menu: 3
```

```
No. Menu                          Price

1.  Carne Asada Burrito            4.49
2.  French Dip Sandwich           10.49
3.  Monte Cristo Sandwich          8.99
4.  Californian Clam Chowder      12.69
5.  Tuna Tartare                  16.00
6.  Avocado Toast                 12.00
7.  Huckleberry Pie                7.25
8.  Marionberry Pie                7.50
9.  Hot Fudge Sundae               8.95
10. Shrimp Cocktail                9.00

Menu No.: 2
Qty     : 1
```

EXERCISES

1. Write a program that enables you to list all menus, add menus, order foods, and print bills.

```
1. Show menu
2. Add menu
3. Order
4. Show order
5. Print bill

0. Exit

Menu: 3
```

```
No. Menu                          Price

1.  Carne Asada Burrito            4.49
2.  French Dip Sandwich           10.49
3.  Monte Cristo Sandwich          8.99
4.  Californian Clam Chowder      12.69
5.  Tuna Tartare                  16.00
6.  Avocado Toast                 12.00
7.  Huckleberry Pie                7.25
8.  Marionberry Pie                7.50
9.  Hot Fudge Sundae               8.95
10. Shrimp Cocktail                9.00

Menu No.: 5
Qty     : 1
```

EXERCISES

1. Write a program that enables you to list all menus, add menus, order foods, and print bills.

```
1. Show menu
2. Add menu
3. Order
4. Show order
5. Print bill

0. Exit

Menu: 3
```

```
No. Menu                          Price

1.  Carne Asada Burrito            4.49
2.  French Dip Sandwich           10.49
3.  Monte Cristo Sandwich          8.99
4.  Californian Clam Chowder      12.69
5.  Tuna Tartare                  16.00
6.  Avocado Toast                 12.00
7.  Huckleberry Pie                7.25
8.  Marionberry Pie                7.50
9.  Hot Fudge Sundae               8.95
10. Shrimp Cocktail                9.00

Menu No.: 9
Qty     : 2
```

EXERCISES

1. Write a program that enables you to list all menus, add menus, order foods, and print bills.

```
1. Show menu
2. Add menu
3. Order
4. Show order
5. Print bill

0. Exit

Menu: 4
```

```
Qty  Menu

1    French Dip Sandwich
1    Tuna Tartare
2    Hot Fudge Sundae
```

EXERCISES

1. Write a program that enables you to list all menus, add menus, order foods, and print bills.

```
1. Show menu
2. Add menu
3. Order
4. Show order
5. Print bill

0. Exit

Menu: 5
```

bill.txt

```
French Dip Sandwich
1 x 10.49          10.49

Tuna Tartare
1 x 16.00          16.00

Hot Fudge Sundae
2 x 8.95           17.90

Total              44.39
```

REFERENCES

- Deitel, P. and Harvey Deitel (2022), C How to Program (9th Edition), Pearson Education.
- Thareja, R. (2014), Data Structures Using C (2nd Edition), India: Oxford University Press.

NEXT

Linked Lists:

Dynamic Memory Allocation

Single Linked Lists

VISION

To become an **outstanding** undergraduate Computer Science program that produces **international-minded** graduates who are **competent** in software engineering and have **entrepreneurial spirit** and **noble character**.

MISSION

1. To conduct studies with the best technology and curriculum, supported by professional lecturer
2. To conduct research in Informatics to promote science and technology
3. To deliver science-and-technology-based society services to implement science and technology

Without hard work,

nothing grows but weeds.

UMN UNIVERSITAS MULTIMEDIA NUSANTARA | INFORMATIKA UMN

Have patience.

All things are difficult before they become easy.