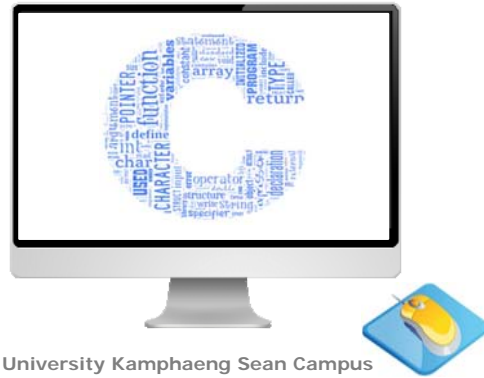


02204171 Structured Programming

Chapter 9 part2: C File Handling



Computer Engineering, Kasetsart University Kamphaeng Sean Campus

Outline

- What is a String?
- String Declaration and Initialization
- Accessing element of String
- Strings I/O in C Programming
- String handling functions
- File Input/Output

ALPPL

2



What is a File?

- แฟ้มข้อมูล (Files) กลุ่มข้อมูลที่ถูกบันทึกไว้ในที่เดียวกัน ในหน่วยความจำสำรองของคอมพิวเตอร์ ซึ่งจะไม่สูญหายเมื่อปิดเครื่อง
 - **Text files:** คือ ไฟล์ที่เก็บอักขระเรียงต่อกัน ซึ่งสามารถเปิดอ่านและ แก้ไขได้ด้วยโปรแกรมประเภทประมวลผลคำ เช่น notepad, Ms Word
 - **Binary files:** คือไฟล์ที่เก็บข้อมูลในรูปแบบเลขฐานสอง (Binary Code) ที่คอมพิวเตอร์สามารถอ่านเข้าใจแต่มนุษย์ไม่เข้าใจ ตัวอย่าง เช่น ไฟล์นามสกุล .exe

ALPPL

3



Basic File Operations

- การเปิดไฟล์ (Opening a file)
- การอ่านข้อมูลจากไฟล์ (Reading data from a file)
- การเขียนข้อมูลลงไฟล์ (Writing data to a file)
- การปิดไฟล์ (Closing a file)



4

1. การเปิดไฟล์ (Opening a file)

- การเปิดไฟล์เพื่อเลือกไฟล์ที่จะประมวลผล เป็นสิ่งแรกที่จะต้องทำก่อนการดำเนินการประมวลผลไฟล์อื่นๆ (อ่าน/เขียน)
การเปิดไฟล์จะต้องใช้ 2 คำสั่ง
 - คำสั่งแรก เป็นการสร้าง Pointer ชนิด FILE เพื่อใช้เป็นตัวชี้ตำแหน่งไฟล์
 - คำสั่งที่สอง คือ **คำสั่งเปิดไฟล์** และกำหนดตัวชี้ตำแหน่งไฟล์ให้กับตัวแปร

Pointer

รูปแบบ (Syntax):

```
FILE *filepointer;  
filepointer = fopen("filename", "mode");
```

- *filename* เป็นชื่อหรือตำแหน่งของไฟล์

- *mode* เป็นชนิดการเปิดไฟล์



5

1. การเปิดไฟล์ (Opening a file)

- การกำหนดชนิดการเปิดไฟล์ (Mode) ในฟังก์ชัน **fopen()** เพื่อให้ C compiler ภาษา C เข้าใจว่าเราต้องการเปิดไฟล์เพื่อทำงานใด โดยชนิดการเปิดไฟล์แสดงดังตารางต่อไปนี้

| Mode | ความหมาย |
|-----------|---|
| r | เปิดไฟล์เพื่ออ่านข้อมูลอย่างเดียว |
| w | เปิดไฟล์เพื่อเขียนข้อมูลทับลงไปทับข้อมูลเก่าในไฟล์ โดยข้อมูลเก่าจะถูกลบทิ้ง |
| a | เปิดไฟล์สำหรับเขียนข้อมูลต่อท้ายข้อมูลเดิมในไฟล์ |
| r+ | เปิดไฟล์สำหรับอ่านและเขียนข้อมูลลงไปในไฟล์ |
| w+ | เปิดไฟล์สำหรับอ่านและเขียนข้อมูลใหม่ทับข้อมูลเดิม |
| a+ | เปิดไฟล์สำหรับอ่านและเขียนข้อมูลใหม่ต่อท้ายข้อมูลเดิม |

การเปิดไฟล์เพื่ออ่าน ไฟล์ที่จะเปิดต้องมีอยู่จริงจึงจะเปิดได้ ส่วนโหมดอื่นๆ ถ้าไฟล์ที่ระบุชื่อยังไม่มีอยู่ ระบบจะสร้างไฟล์ใหม่ขึ้นมา



6

1. การเปิดไฟล์ (Opening a file)

- ตัวอย่าง: การเปิดไฟล์

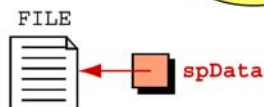
```
FILE *spData;  
spData = fopen("MYDATA.DAT", "w");
```

```
#include <stdio.h>  
...  
{  
  int main (void)  
  {  
    FILE* spData;  
    ...  
    spData = fopen("MYDATA.DAT", "w");  
    ...  
  }  
} // main
```

Internal File Variable

External File Name

ผลลัพธ์ของการเปิดไฟล์ มี 2 กรณี
ถ้าเปิดไฟล์สำเร็จ (ไม่มีข้อผิดพลาดใดๆ) ผลลัพธ์ คือ ตำแหน่งตัวชี้ไฟล์
แต่หากไม่สามารถเปิดไฟล์
ผลลัพธ์ คือ NULL



7

1. การเปิดไฟล์ (Opening a file)

- ตัวอย่าง 1: การเปิดไฟล์

```
FILE *fptr;  
char filename[] = "c:\\myfile.txt";  
fptr = fopen (filename, "r");  
  
if (fptr == NULL) {  
  printf ("Cannot open file to read!\n");  
  return (-1);  
}
```

หากไม่มี ไฟล์ชื่อ "myfile.txt" ที่ตำแหน่ง c:
ตัวชี้ไฟล์ชื่อ fptr จะมีค่าเป็น NULL



8

2. การปิดไฟล์ (Closing a file)

- ฟังก์ชัน `fclose()` ใช้สำหรับการปิดไฟล์ในภาษา C โดยระบุตัวชี้ไฟล์ที่ต้องการปิด หากสำเร็จฟังก์ชันจะส่งค่า 0 หากเกิดข้อผิดพลาดจะส่งค่า EOF (End Of File)

รูปแบบ (Syntax):

```
fclose(filepointer);
```

```
FILE *spData;  
spData= fopen("MYDATA.DAT", "w");  
  
//file operation  
fclose(spData)
```

หลังจากเสร็จสิ้นการประมวลผลไฟล์ เราจำเป็นต้องปิดไฟล์เพื่อให้ระบบนำไฟล์ไปใช้อย่างอื่นได้ต่อไป



9

3.การอ่านข้อมูลจากไฟล์ (Reading data from a file)

- ฟังก์ชันที่ใช้ในการอ่านข้อมูลจาก Text file

| | |
|------------------------|--------------------------------------|
| <code>—fgetc()</code> | อ่านข้อมูลที่ละตัวอักษร |
| <code>—fgets()</code> | อ่านข้อมูลมาเป็นข้อความทีละบรรทัด |
| <code>—fscanf()</code> | อ่านข้อมูลเป็นข้อความเหมือนกับ scanf |



10

3.การอ่านข้อมูลจากไฟล์ (Reading data from a file)

- `fgetc()` อ่านข้อมูลจากไฟล์ทีละตัวอักษร รวมถึงสัญลักษณ์พิเศษ เช่น `'\n'`, `'\t'`

- หากอ่านไฟล์ไปจนถึงจุดสิ้นสุดของไฟล์ ฟังก์ชันจะคืนค่าเป็น EOF (End of file)

รูปแบบ (Syntax):

```
fgetc(filepointer);
```

ตัวอย่าง 2: อ่านไฟล์ทีละตัวอักษร

```
FILE *fp;  
char c ;  
fp = fopen ("quote2.txt", "r") ;  
while ((c = fgetc (fp)) != EOF)  
    putchar (toupper (c));  
fclose (fp) ;
```

โปรแกรมจะอ่านทีละตัวอักษรในไฟล์ แล้วแปลงเป็นตัวพิมพ์ใหญ่

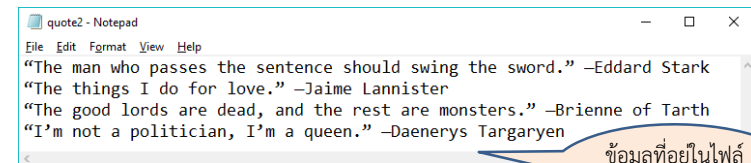
ทำซ้ำไปเรื่อยๆ จนกว่าจะเจอจุดสิ้นสุดของไฟล์ (EOF)



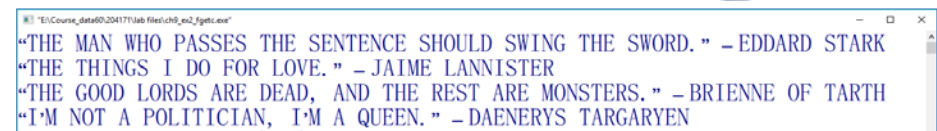
11

3.การอ่านข้อมูลจากไฟล์ (Reading data from a file)

- ผลการทำงานของโปรแกรมใน ตัวอย่างที่ 2



ข้อมูลที่อยู่ในไฟล์
quote2.txt



ผลลัพธ์ทาง
จอภาพ



12

3.การอ่านข้อมูลจากไฟล์ (Reading data from a file)

- **fgets()** อ่านข้อมูลจากไฟล์ทีละบรรทัด (อ่านจนกว่าจะเจอ ‘\n’)
ถ้าอ่านจนครบทุกบรรทัดในไฟล์ ฟังก์ชันจะคืนค่าเป็น NULL
รูปแบบ (Syntax):

```
fgets(str_variable, str_size, filepointer);
```

ตัวอย่าง 3: อ่านไฟล์ทีละบรรทัด

```
FILE *fp;  
char line[100];  
fp = fopen ("quote2.txt", "r") ;  
while (fgets (line, 100, fp) != NULL)  
    printf ("%s",line);  
fclose (fp) ;
```



13

3.การอ่านข้อมูลจากไฟล์ (Reading data from a file)

- **fscanf()** อ่านข้อมูลจากไฟล์เป็นข้อความเหมือนกับ scanf

```
fscanf(filepointer, "format code", variable);
```

ตัวอย่าง 4: อ่านไฟล์บรรทัดแรก

```
FILE *fp;  
char line[100];  
  
fp = fopen ("quote2.txt", "r") ;  
fscanf(fp, "%[^\\n]s", line);  
printf ("%s",line);  
  
fclose (fp) ;
```

หากไม่ใส่ผลลัพธ์
ที่ได้ คือ "The

```
"The man who passes the sentence should swing the sword." -Eddard Stark  
Process returned 0 (0x0)   execution time : 0.034 s  
Press any key to continue.
```

14

4.การเขียนข้อมูลลงไฟล์ (writing data to a file)

- ฟังก์ชันที่ใช้ในการเขียนข้อมูลลง Text file

—**fputc()** เขียนข้อมูลที่ละตัวอักษร

รูปแบบ
(Syntax): **fputc**(char, filepointer);

—**fputs()** เขียนข้อมูลแบบข้อความ

รูปแบบ
(Syntax): **fputs**(string, filepointer);

—**fprintf()** เขียนข้อมูลเป็นข้อความเหมือนกับ printf

รูปแบบ
(Syntax): **fprintf**(filepointer, "format code", string);



15

4.การเขียนข้อมูลลงไฟล์ (writing data to a file)

- ตัวอย่าง 5: การเขียนไฟล์

เขียนข้อมูลนี้ ลงบนไฟล์ชื่อ grade.txt

| | | |
|------|---|---|
| Math | 4 | B |
| Eng | 3 | B |

```
FILE *fp;  
char subject[]="Math";  
fp = fopen ("grade.txt", "w") ;  
  
fprintf(fp, "%s\t 4\t B\\n", subject);  
  
fputs("Eng\\t", fp);  
fputc('3', fp);  
fputc('\\t', fp);  
fputc('B', fp);  
  
fclose (fp) ;
```

ข้อมูลที่อยู่ในไฟล์
grade.txt

| | | | | |
|------|------|--------|------|------|
| File | Edit | Format | View | Help |
| Math | 4 | B | | |
| Eng | 3 | B | | |



16

4.การเขียนข้อมูลลงไฟล์ (writing data to a file)

■ ตัวอย่าง 6: การเขียนไฟล์ ต่อท้าย

เขียนข้อมูลชื่อวิชา เกรด ที่รับจากคีย์บอร์ด และหน่วยกิต เป็น 3 เสมอ
ลงบนไฟล์ชื่อ grade.txt

```
do{
    printf("Enter Subject:");
    gets(subject);

    printf("Enter grade:");
    grade=getchar();

    fprintf(fp, "\n%s\t %d\t %c", subject, credit, grade);

    printf("Subject info already save in file\n");
    printf("Do you wish to continue (Y/N):");
    next=getchar();
}while(next!='N');
fclose (fp);
```

```
FILE *fp;
char subject[10], grade;
char next='Y';
int credit=3;
fp = fopen ("grade.txt", "a");
```

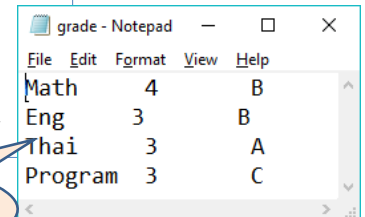
17

4.การเขียนข้อมูลลงไฟล์ (writing data to a file)

■ ผลการทำงานของโปรแกรมใน ตัวอย่างที่ 6

Enter Subject: **Thai**
Enter grade: **A**
Subject info already save in file
Do you wish to continue (Y/N): **Y**

Enter Subject: **Program**
Enter grade: **C**
Subject info already save in file
Do you wish to continue (Y/N): **N**



| | | |
|---------|---|---|
| Math | 4 | B |
| Eng | 3 | B |
| Thai | 3 | A |
| Program | 3 | C |

ข้อมูลที่อยู่ในไฟล์
grade.txt ที่เพิ่มขึ้นมา

18

Summary

❖ ชนิดของแฟ้มข้อมูล

- ❖ Text file เป็นการจัดเก็บข้อมูลในลักษณะข้อความ (Text)
- ❖ Binary file เป็นการจัดเก็บแฟ้มข้อมูลในระบบเลขฐานสอง

❖ หลักการเขียนโปรแกรมการจัดแฟ้มข้อมูล ประกอบด้วย

- ❖ การเปิดแฟ้มข้อมูล
- ❖ การอ่านข้อมูล / การบันทึกข้อมูล
- ❖ การปิดแฟ้มข้อมูล

❖ การเปิดแฟ้มข้อมูล (Open file) รูปแบบ

```
FILE *fp
fp = fopen(fileName, Mode)
```

❖ การปิดแฟ้มข้อมูล (Close file) รูปแบบ

```
fclose(fp)
```

19