

File Handling

BY NATSUDA KAOTHANTHONG

1

Today's Lecture

File handling

- ไฟล์ (File) ใช้นำเสนอชุดข้อมูลที่ถูกเก็บไว้บนพื้นที่ภายในคอมพิวเตอร์
- มีรูปแบบในการเก็บข้อมูล

Malee 20 30 50
Chalie 10 26 36
Ken 18 27 45

ตัวอย่างไฟล์ score.txt

Jackson 01/11/90 Florida
Nikky 12/02/91 London
Amy 09/09/90 NewYork

ตัวอย่างไฟล์ personallInfo.txt

2

ขั้นตอนสำคัญในการทำงานกับไฟล์

3 ขั้นตอนสำคัญ

1. เปิดไฟล์ : ต้องเปิดไฟล์ก่อนทุกครั้ง
2. ทำงานกับไฟล์
 - อ่านไฟล์
 - เขียนไฟล์
3. ปิดไฟล์ : ต้องปิดไฟล์ทุกครั้ง

3

การทำงานกับไฟล์

ภาษาซีใช้ตัวแปรพอยน์เตอร์ในการระบุตำแหน่งของไฟล์ในหน่วยเก็บในคอมพิวเตอร์

- การสร้างตัวแปร **ไฟล์พอยน์เตอร์ (file pointer)**
`FILE *fPtr;`

4

การเปิดไฟล์

- เมื่อประกาศตัวแปรไฟล์พอยท์เตอร์แล้ว
- กำหนดตำแหน่งของไฟล์ให้กับตัวแปรพอยท์เตอร์โดยการเปิดไฟล์
- ฟังก์ชันสำหรับเปิดไฟล์ `fopen("ที่อยู่ไฟล์", "การทำงาน");`
- ตัวอย่าง

```
FILE *fPtr;  
fPtr = fopen("ที่อยู่ไฟล์", "การทำงาน");
```

5

การทำงานกับไฟล์(Mode)

- ในการทำงานกับไฟล์ต้องเลือกว่าเปิดไฟล์เพื่ออะไร
- กำหนด **Mode** การทำงาน
 - **Mode : r** คือ โหมดสำหรับอ่านไฟล์เท่านั้น
 - **Mode : w** คือ โหมดสำหรับเขียน โดยเขียนทับกับเนื้อหาที่มีอยู่แล้ว
 - **Mode : a** คือ โหมดสำหรับเขียนต่อจากเนื้อหาที่มีอยู่แล้ว
- ตัวอย่าง

```
FILE *fPtr;  
fPtr = fopen("score.txt", "r");
```

6

ขั้นตอนสำคัญในการทำงานกับไฟล์

3 ขั้นตอนสำคัญ

1. เปิดไฟล์ : ต้องเปิดไฟล์ก่อนทุกครั้ง
2. ทำงานกับไฟล์
 - อ่านไฟล์
 - เขียนไฟล์
3. ปิดไฟล์ : ต้องปิดไฟล์ทุกครั้ง

7

คำสั่งในการอ่าน/เขียนไฟล์

หลังจากเปิดไฟล์สามารถทำงานกับไฟล์ได้โดยใช้คำสั่งดังนี้

- **อ่านข้อมูลในไฟล์**
 - **fgetc(ตัวแปรเก็บค่า, จำนวนตัวอักษร, ไฟล์พอยท์เตอร์);**
 - ใช้เพื่ออ่านข้อมูลทั้งบรรทัด
 - **fscanf(ไฟล์พอยท์เตอร์, "รูปแบบข้อความ", ที่อยู่ตัวแปรต่าง ๆ);**
 - อ่านข้อความและเก็บข้อมูลในตัวแปร
- **เขียนข้อมูลลงไฟล์**
 - **fprintf(ไฟล์พอยท์เตอร์, "รูปแบบข้อความ", ตัวแปรต่าง ๆ);**
 - เขียนข้อมูลตามรูปแบบที่กำหนด

8

fgets(ตัวแปรเก็บค่า, จำนวนตัวอักษร, ไฟล์พอยเตอร์);

ฟังก์ชัน fgets ใช้อ่านทั้งบรรทัดในไฟล์

fgets(stringVariable, size, file_ptr);

ไฟล์พอยเตอร์ และ เปิดไฟล์โดยใช้ mode r

```
FILE *fptr;  
fptr = fopen("score.txt", "r");
```

```
char line[1000];  
fgets(line, 1000, fptr);
```

อ่าน 1 บรรทัด และ เก็บค่าในตัวแปร line

```
printf("Line: %s\n", line);
```

```
Narong 15.0  
Joy 20.5  
Nam 22.0  
Fon 18.0  
Ploy 20.0  
Petch 25.0  
Malee 13.5  
Sing 24.5  
Ni 19.5  
Papon 19.0  
Nuta 23.0  
Mai 21.5
```

score.txt

9

อ่านทั้งไฟล์โดยใช้ฟังก์ชัน fgets()

- ใช้ while และ fgets เพื่ออ่านทั้งไฟล์

```
FILE *fptr;  
fptr = fopen("score.txt", "r");
```

```
char line[1000];
```

อ่านจำนวน 1000 ตัวอักษร

```
while(fgets(line, 1000, fptr) != NULL)  
{  
    printf("Line: %s\n", line);  
}
```

อ่านจนจบไฟล์ ใช้
เปรียบเทียบกับค่า
NULL

10

การอ่านไฟล์โดยใช้ fscanf()

- ข้อมูลถูกอ่านทั้งบรรทัด
- ตัวอย่าง fgets(line, 1000, fptr);
- ค่าภายในตัวแปร line คือ
 - “Narong 15.0”
 - เป็นชนิด string
- หากต้องการอ่านค่าแยกเป็น
 - ชื่อ Narong
 - คะแนน 15.0
 - ต้องใช้ฟังก์ชัน fscanf()

```
Narong 15.0  
Joy 20.5  
Nam 22.0  
Fon 18.0  
Ploy 20.0  
Petch 25.0  
Malee 13.5  
Sing 24.5  
Ni 19.5  
Papon 19.0  
Nuta 23.0  
Mai 21.5
```

score.txt

11

fscanf(ไฟล์พอยเตอร์, “รูปแบบข้อความ”, ที่อยู่ตัวแปรต่าง ๆ);

- อ่านข้อมูลในไฟล์ตามรูปแบบข้อความ
- แต่ละค่าที่กำหนดในรูปแบบสามารถเก็บในตัวแปรต่าง ๆ ได้

```
FILE *fptr;  
fptr = fopen("score.txt", "r");
```

```
char name1[100];  
float number;
```

ประกาศตัวแปรสำหรับเก็บค่า

```
fscanf(fptr, "%s %f", name1, &number);
```

อ่านค่า 1 บรรทัด และเก็บค่าลงใน ตัวแปร name1 และ number

```
Narong 15.0  
Joy 20.5  
Nam 22.0  
Fon 18.0  
Ploy 20.0  
Petch 25.0  
Malee 13.5  
Sing 24.5  
Ni 19.5  
Papon 19.0  
Nuta 23.0  
Mai 21.5
```

score.txt

12

fscanf(ไฟล์พอยท์เตอร์, "รูปแบบข้อความ", ที่อยู่ตัวแปรต่าง ๆ);

เมื่ออ่านค่า 1 บรรทัดจากไฟล์ และเก็บค่าลงใน ตัวแปร

name1 และ number โดยใช้ฟังก์ชัน fscanf()

```
fscanf(fp_ptr, "%s %f", name1, &number);  
printf("name1: %s\n", name1);  
printf("number: %.2f\n", number);
```

■ ผลการแสดงผลคือ

name1: Narong

number: 15.00

```
Narong 15.0  
Joy 20.5  
Nam 22.0  
Fon 18.0  
Ploy 20.0  
Petch 25.0  
Malee 13.5  
Sing 24.5  
Ni 19.5  
Papon 19.0  
Nuta 23.0  
Mai 21.5
```

score.txt

13

อ่านทั้งไฟล์โดยใช้ฟังก์ชัน fscanf

```
FILE *fp_ptr = fopen("score.txt", "r");  
char name[numLine][100];  
float score[numLine];  
int index = 0;  
while(fscanf(fp_ptr, "%s %f", name[index], &score[index]) != EOF)  
{  
    printf("Name : %s\tScore : %.2f\n", name[index], score[index]);  
    index++;  
}
```

อ่านจนจบไฟล์ ใช้
เปรียบเทียบกับค่า EOF

14

คำสั่งในการอ่าน/เขียนไฟล์

หลังจากเปิดไฟล์สามารถทำงานกับไฟล์ได้โดยใช้คำสั่งดังนี้

■ อ่านข้อมูลในไฟล์

■ fgets(ตัวแปรเก็บค่า, จำนวนตัวอักษร, ไฟล์พอยท์เตอร์);

■ ใช้เพื่ออ่านข้อมูลทั้งบรรทัด

■ fscanf(ไฟล์พอยท์เตอร์, "รูปแบบข้อความ", ที่อยู่ตัวแปรต่าง ๆ);

■ อ่านข้อความและเก็บข้อมูลในตัวแปร

■ เขียนข้อมูลลงไฟล์

■ fprintf(ไฟล์พอยท์เตอร์, "รูปแบบข้อความ", ตัวแปรต่าง ๆ);

■ เขียนข้อมูลตามรูปแบบที่กำหนด

15

การเขียนข้อมูลลงไฟล์

ฟังก์ชัน printf("stringFormat", variables); ใช้แสดงข้อความบน
หน้าจอตามรูปแบบและค่าตัวแปรที่กำหนด

ฟังก์ชัน fprintf ใช้เขียนข้อมูลลงไฟล์ตามรูปแบบและค่าตัวแปรที่กำหนด
fprintf(file_ptr, "stringFormat", variables);

16

การเปิดไฟล์เพื่อเขียนข้อมูลลงไฟล์

- การทำงาน (mode) ที่ใช้ในการเปิดไฟล์เพื่อเขียนข้อมูลคือ w
- w ย่อมาจาก write

ไฟล์พอยท์เตอร์ และ เปิดไฟล์โดยใช้ mode w

```
FILE *filePtr = NULL;  
filePtr = fopen("myfile.txt", "w");
```

17

การเขียนข้อมูลลงไฟล์โดยใช้ fprintf()
fprintf(file_ptr, "stringFormat", variables);

```
FILE *filePtr = NULL;  
filePtr = fopen("myfile.txt", "w");
```

- เขียนเฉพาะข้อความลงไฟล์
- fprintf(filePtr, "Hello World\n");
- สามารถใส่ส่วนของ variables ออกได้

18

การเขียนข้อมูลลงไฟล์โดยใช้ fprintf()
fprintf(file_ptr, "stringFormat", variables);

```
FILE *filePtr = NULL;  
filePtr = fopen("myfile.txt", "w");
```

- เขียนข้อความและค่าจากตัวแปรลงไฟล์
- int age = 18;
- char name[] = "John";
- fprintf(filePtr, "name: %s\tage: %d\n", name, age);
- ในไฟล์ myfile.txt :

name: John	age: 18
------------	---------

myfile.txt

19

การเขียนข้อมูลลงไฟล์โดยใช้โหมดการทำงาน "a"

```
FILE *filePtr = NULL;  
filePtr = fopen("myfile.txt", "a");
```

- เขียนข้อความและค่าจากตัวแปรเพิ่มลงไฟล์
- int age = 19;
- char name[] = "Ken";
- fprintf(filePtr, "name: %s\tage: %d\n", name, age);

name: John	age: 18
------------	---------

myfile.txt

(ข้อมูลที่มีอยู่ก่อนเปิดไฟล์)

name: John	age: 18
name: Ken	age: 19

myfile.txt

(ข้อมูลหลังเขียนข้อความเพิ่ม)

20

ขั้นตอนสำคัญในการทำงานกับไฟล์

3 ขั้นตอนสำคัญ

1. เปิดไฟล์ : ต้องเปิดไฟล์ก่อนทุกครั้ง
2. ทำงานกับไฟล์
 - อ่านไฟล์
 - เขียนไฟล์
3. ปิดไฟล์ : ต้องปิดไฟล์ทุกครั้ง

21

การปิดไฟล์

หลังจากใช้ไฟล์เสร็จแล้ว ต้องทำการปิดไฟล์ทุกครั้ง

- ปิดไฟล์ด้วยฟังก์ชัน **fclose**
 - **fclose(ไฟล์พอยน์เตอร์);**
- ในหนึ่งโปรแกรมสามารถเปิดปิดไฟล์ได้มากกว่าหนึ่งครั้ง
- ต้องปิดไฟล์ก่อน แล้วจึงเปิดอีกครั้ง

22

ตัวอย่างการเปิดปิดไฟล์ 2 ครั้งใน 1 โปรแกรม

```
int numLine = 0;
char line[1000];

FILE *fptr;
fptr = fopen("score.txt", "r");
while (fgets(line, 1000, fptr) != NULL)
    numLine++;
fclose(fptr);

fptr = fopen("score.txt", "r");
char name[numLine][100];
float score[numLine];
int index = 0;

float min = 10000;
float max = 0;
float average = 0;
while (fscanf(fptr, "%s %f", name[index], &score[index]) != EOF)
{
    }
```

ใช้นับจำนวนข้อมูลในไฟล์

เปิดไฟล์อีกรอบเพื่ออ่านข้อมูลในไฟล์

23

สรุป

- การทำงานร่วมกับไฟล์
- การเปิดไฟล์
 - โหมดที่สามารถใช้ได้ในการทำงานกับไฟล์
 - **r** : โหมดสำหรับอ่านไฟล์
 - **w** : โหมดสำหรับเขียนไฟล์ โดยจะเขียนทับข้อมูลที่มีอยู่แล้ว
 - **a** : โหมดสำหรับเขียนไฟล์ต่อจากข้อมูลที่มีอยู่แล้วในไฟล์

24

สรุป

- คำสั่งในการอ่าน และ เขียนไฟล์
 - การอ่านไฟล์
 - `fgets();`
 - `fscanf();`
 - การเขียนไฟล์
 - `fprintf();`
- การปิดไฟล์