

บทที่ 8 การจัดการไฟล์ข้อมูล

วิชาการเขียนโปรแกรมภาษาคอมพิวเตอร์ (Computer Programming)
โดย อ.ชรินทร์ญา กล้าแข็ง
(www.charinya-lru.com)

1

ทำไมต้องมีไฟล์ข้อมูล

- * ก่อนหน้านี้เราจะเขียนโปรแกรมและแสดงผลออกทางจอภาพเท่านั้น ถ้าเราปิดโปรแกรมผลการรันโปรแกรมก็จะหายไป เราต้องรันโปรแกรมใหม่ เพื่อให้เห็นผลลัพธ์
- * ซึ่งงานหลายอย่างจำเป็นต้องมีการเก็บข้อมูลอย่างถาวรลงบนไฟล์ในเครื่องคอมพิวเตอร์ โดยสามารถเปิดอ่านและเขียนข้อมูลจากโปรแกรมอื่นๆได้ด้วย
- * ในภาษา C เราสามารถเขียนโปรแกรมในการเก็บข้อมูลลงบนไฟล์ข้อมูลหรือแฟ้มข้อมูลไว้บนหน่วยความจำของเครื่องคอมพิวเตอร์ได้

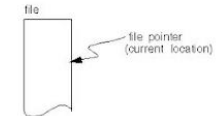
2

ไฟล์ข้อมูลในภาษา C

- * การจัดการไฟล์ข้อมูลในภาษา C เราจำเป็นต้องเรียนรู้คำสั่งต่างๆ ในการเปิดไฟล์ ปิดไฟล์ อ่านไฟล์ ฯลฯ
- * ซึ่งเราจะใช้ไลบรารี “stdio.h” ในการทำงานในเรื่องของไฟล์ได้
- * ซึ่งไฟล์ที่ได้จะถูกเก็บลงบนหน่วยความจำของเครื่องคอมพิวเตอร์
- * โดยการอ่านและเขียนไฟล์นี้ เราจะต้องใช้ตัวแปรตัวหนึ่งที่เรียกว่า “ไฟล์พอยเตอร์” (File Pointer)
- * ในบทนี้เราจะเน้นเฉพาะคำสั่งพื้นฐานในการอ่านและเขียนไฟล์

3

ไฟล์พอยเตอร์



- * ไฟล์พอยเตอร์นี้ (File Pointer) นี้จะเป็นตัวชี้ว่าเรากำลังจะอ่าน เขียนที่ตำแหน่งใดของไฟล์ เช่น ต้นไฟล์ กลางไฟล์ หรือต่อท้ายไฟล์ ถ้าไม่มีไฟล์พอยเตอร์ จะไม่สามารถจัดการใดๆกับไฟล์ได้เลย
- * คำสั่งในการจัดการไฟล์ แบ่งออกเป็น
- * 1. การเปิดไฟล์: จะต้องบอกให้โปรแกรมรู้ว่า จะจัดการกับไฟล์ชื่ออะไร มีอยู่แล้วหรือไม่
- * 2. การปิดไฟล์: เมื่ออ่านและเขียนข้อมูลลงไฟล์แล้วจะต้องปิดทุกครั้ง
- * 3. การดำเนินการไฟล์: ระบุให้โปรแกรมรู้ว่า เราต้องการจะเขียนหรืออ่านไฟล์

4

คำสั่งเปิดไฟล์

รูปแบบ:

```
FILE *fp; ← -----
fp = fopen("filename", "mode")
```

ตัวแปรชื่อ fp เป็น
ไฟล์พอยเตอร์

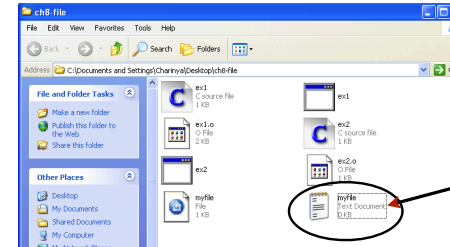
Mode	การทำงาน
w	สร้างไฟล์ใหม่ เพื่อเขียนข้อมูล
r	เปิดไฟล์ เพื่ออ่านข้อมูล
a	เปิดไฟล์ เพื่อเขียนข้อมูลต่อท้าย
wb	สร้างไฟล์ไบนารีใหม่ เพื่อเขียนข้อมูล
rb	เปิดไฟล์ไบนารี เพื่ออ่านข้อมูล
ab	เปิดไฟล์ไบนารี เพื่อเขียนข้อมูลต่อท้าย
w+	เปิดไฟล์ เพื่ออ่านหรือเขียนทับไฟล์เดิม หรือไฟล์ใหม่
r+	เปิดไฟล์ เพื่ออ่านหรือเขียนทับไฟล์เดิม
a+	เปิดไฟล์ เพื่อเขียนต่อท้ายไฟล์เก่า หรือเขียนไฟล์ใหม่
w+b	เปิดไฟล์ไบนารี เพื่ออ่านหรือเขียนทับไฟล์เดิม หรือไฟล์ใหม่
r+b	เปิดไฟล์ไบนารี เพื่ออ่านหรือเขียนทับไฟล์เดิม
a+b	เปิดไฟล์ไบนารี เพื่อเขียนต่อท้ายไฟล์เก่า หรือเขียนไฟล์ใหม่

Ex. 1 ตัวอย่างการเปิดไฟล์ เพื่อสร้างไฟล์ใหม่

```
#include <stdio.h>

void main() {
    FILE *fp;
    fp = fopen("myfile.txt", "w")
}
```

Note: ตัวอย่างนี้จะได้ไฟล์ใหม่ชื่อ "myfile.txt" โดยยังไม่เขียนข้อมูลลงไฟล์นี้ (ยังไม่ใส่คำสั่งในการเขียน)



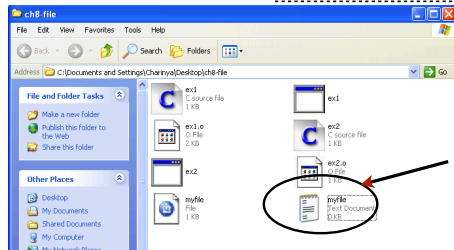
ได้ไฟล์ใหม่เกิดขึ้นมา

Ex2. ตัวอย่างการเปิดไฟล์ เพื่อสร้างไฟล์ใหม่

```
#include <stdio.h>

void main() {
    FILE *fp;
    if((fp = fopen("myfile.txt", "w")) == NULL) {
        printf("Error: Can not open file");
        exit(1);
    }
}
```

Note: ใช้คำสั่ง if เข้ามาช่วยในการตรวจสอบว่าสามารถเปิดไฟล์ได้หรือไม่ ถ้าไม่ได้ให้ออกจากโปรแกรมทันที ไม่ต้องทำคำสั่งอื่นๆอีก



ได้ไฟล์ใหม่เกิดขึ้นมา

คำสั่งปิดไฟล์

```
รูปแบบ:
int fclose(FILE *fp)
```

จะใช้ฟังก์ชันชื่อ fclose() ถ้ามีการเปิดไฟล์แล้ว จะต้องมีการปิดไฟล์ทุกครั้งเมื่อใช้งานเสร็จ

```
#include <stdio.h>

void main() {
    FILE *fp;
    fp = fopen("myfile.txt", "w")

    fclose(fp); //คำสั่งปิดไฟล์
}
```

คำสั่งเขียนข้อมูลลงไฟล์

- คำสั่งนี้จะเขียนข้อมูลลงไฟล์ที่เปิดไว้
- ดังนั้น เราต้องใช้คำสั่งเปิดไฟล์ก่อนเสมอ จึงจะใช้คำสั่งเขียนข้อมูลได้
- การเขียนข้อมูลลงไฟล์ ในที่นี้จะแยกออกเป็น 2 คำสั่งคือ
 - (1). เขียนข้อมูลลงไฟล์ ทีละไบต์ (ทีละ 1 ตัวอักษร)
 - (2). เขียนข้อมูลลงไฟล์ทั้งสตริง (ข้อความยาวๆ)

รูปแบบ: เขียนข้อมูลลงไฟล์ ทีละไบต์ (ทีละ 1 ตัวอักษร)

```
int fputc(char ch, FILE *fp);
```

รูปแบบ: เขียนข้อมูลลงไฟล์ทั้งสตริง (ข้อความยาวๆ)

```
int fputs("ข้อความสตริง", FILE *fp)
```

9

Ex. 3 ตัวอย่างการเขียนข้อมูลลงไฟล์ ทีละ 1 ตัวอักษร



ใช้คำสั่ง fputc()

```
#include<stdio.h>
```

```
main() {
```

```
FILE *fp;
```

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

```
fputc('H', fp);
```

```
fputc('E', fp);
```

```
fputc('L', fp);
```

```
fputc('L', fp);
```

```
fputc('O', fp);
```

```
fclose(fp);
```

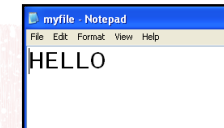
```
}
```

คำสั่งเปิดไฟล์

Mode เป็น "w"
เป็นการสร้างไฟล์ใหม่

คำสั่งเขียนข้อมูลลงไฟล์ ทีละตัวอักษร

คำสั่งปิดไฟล์



10

Ex. 4 ตัวอย่างการเขียนข้อมูลลงไฟล์ ทีละ 1 ตัวอักษร (ข้อความในอาเรย์)



ใช้คำสั่ง fputc()

```
#include<stdio.h>
```

```
main() {
```

```
char mystring[30] = "Good Morning!! How are you?";
```

```
FILE *fp;
```

```
fp = fopen("myfile4.txt", "w");
```

```
int i;
```

```
for(i=0; i<30; i++){
```

```
    fputc(mystring[i], fp);
```

```
}
```

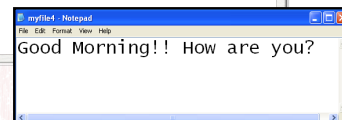
```
fclose(fp);
```

```
}
```

คำสั่งเปิดไฟล์

คำสั่งเขียนข้อมูลลงไฟล์ทีละตัวอักษร
วนลูปเขียนข้อมูลในสตริง

คำสั่งปิดไฟล์



11

Ex. 5 ตัวอย่างการเขียนข้อมูลลงไฟล์ ทีละ 1 ตัวอักษร (ข้อความในอาเรย์) เขียนข้อมูลต่อจากไฟล์เดิม



ใช้คำสั่ง fputc()

```
#include<stdio.h>
```

```
main() {
```

```
char mystring[30] = "\n I'm fine. Thank you &you?";
```

```
FILE *fp;
```

```
fp = fopen("myfile4.txt", "a");
```

```
int i;
```

```
for(i=0; i<30; i++){
```

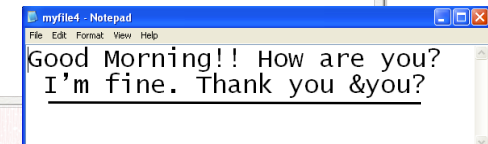
```
    fputc(mystring[i], fp);
```

```
}
```

```
fclose(fp);
```

```
}
```

เปลี่ยน Mode ให้เป็น "a"
เป็นการเขียนต่อท้ายไฟล์เดิมที่มีอยู่



12

Ex.6 ตัวอย่างการเขียนข้อมูลลงไฟล์ ทั้งข้อความสตริง

```
#include<stdio.h>


main(){

    FILE *fp;
    fp = fopen("myfile6.txt","w");

    fputs("I am student. \n",fp);
    fputs("My name is Charinya. \n",fp);
    fputs("I Love Business Computer major.\n",fp);

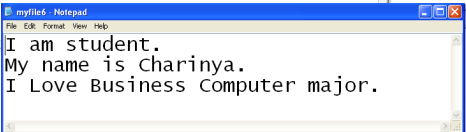
    fclose(fp);
}

```



ใช้คำสั่ง fputs()

คำสั่งเขียนข้อมูลลงไฟล์ทั้งข้อความสตริง



13

คำสั่งอ่านข้อมูลในไฟล์

รูปแบบ: อ่านข้อมูลไฟล์ ทีละไบต์ (ทีละ 1 ตัวอักษร)

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

รูปแบบ: อ่านข้อมูลทั้งสตริง (ข้อความยาวๆ)

```
char *fgets(ตัวแปรสตริง,num,FILE *fp);
```

14

Ex.7 ตัวอย่างการอ่านข้อมูลจากไฟล์ ทีละตัวอักษร

```
#include<stdio.h>

main(){


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

    char c;

    c = fgetc(fp);
    printf("data = %c \n",c);
    c = fgetc(fp);
    printf("data = %c \n",c);
    c = fgetc(fp);
    printf("data = %c \n",c);
    c = fgetc(fp);
    printf("data = %c \n",c);
    c = fgetc(fp);
    printf("data = %c \n",c);

    fclose(fp);
}

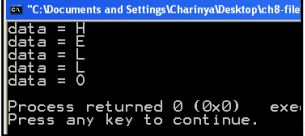
```



ใช้คำสั่ง fgetc()

Mode เป็น "r"
เป็นการเปิดไฟล์เพื่ออ่าน

อ่านข้อมูลทีละตัวอักษร คำสั่ง fgetc()
จะเลื่อนพอยเตอร์เองอัตโนมัติ



15

Ex.8 ตัวอย่างการอ่านข้อมูลจากไฟล์ ทีละตัวอักษร (ใช้ลูปมาช่วย)

```
#include<stdio.h>

main(){


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

    char c;

    while(!feof(fp)){
        c = fgetc(fp);
        printf("data = %c \n", c);
    }

    fclose(fp);
}

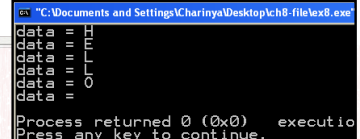
```



ใช้คำสั่ง fgetc()

Mode เป็น "r"
เป็นการเปิดไฟล์เพื่ออ่าน

ให้อ่านจนกว่าจะหมดไฟล์ คำสั่ง feof()
อ่านข้อมูลทีละตัวอักษร คำสั่ง fgetc() จะ
เลื่อนพอยเตอร์เองอัตโนมัติ



16

Ex.9 ตัวอย่างการอ่านข้อมูลจากไฟล์ อ่านข้อมูลทั้งหมด โดยใช้คำสั่ง fgets()

```
#include<stdio.h>

main() {
    FILE *fp;
    fp = fopen("myfile9.txt", "r");

    char str[500];

    while (!feof(fp)) {
        if (fgets(str, 500, fp)) {
            printf("%s", str);
        }
    }
    fclose(fp);
}
```

ใช้คำสั่ง fgets()

อ่านข้อมูลมาเก็บในตัวแปรสตริง แล้วแสดงผล

```
Happy new year
Happy new year
May we all have a vision now and then
Of a world where every neighbour is a friend
Happy new year
Happy new year
May we all have our hopes, our will to try
If we don't we might as well lay down and die
You and I
Process returned 0 (0x0)   execution time : 0.000 s
Press any key to continue.
```

Ex.10 คำสั่ง fprintf()

```
#include<stdio.h>

main() {
    FILE *fp;
    fp = fopen("myfile10.txt", "w");

    int i, j, y;

    i = 10, j = 20;
    y = i*j;

    /* printf("Result of %d * %d = %d \n\n", i, j, y); */
    fprintf(fp, "Result of %d * %d = %d", i, j, y);

    fclose(fp);
}
```

Mode เป็น "w" เขียนลงไฟล์

```
myFile10 - Notepad
File Edit Format View Help
Result of 10 * 20 = 200
```

Ex.11 รับข้อมูลจากแป้นพิมพ์ และเขียนข้อมูลลงไฟล์

```
#include<stdio.h>

main() {
    FILE *fp;
    fp = fopen("myfile11.txt", "w");

    char fname[50], lname[50];
    int age;

    printf("Hello. Enter ctrl+c to end input \n");

    while (!feof(stdin)) {
        printf("What's your firstname? \n");
        printf("> ");
        scanf("%s", &fname);

        printf("What's your lastname? \n");
        printf("> ");
        scanf("%s", &lname);

        printf("How old are you? \n");
        printf("> ");
        scanf("%d", &age);

        fprintf(fp, " %s %s : %d \n", fname, lname, age);
    }

    fclose(fp);
}
```

วนลูปรับข้อมูลจนกว่าจะกด Ctrl+c

```
Hello. Enter ctrl+c to end input
What's your firstname?
> charinya
What's your lastname?
> klakhang
How old are you?
> 27
What's your firstname?
> angry
What's your lastname?
> bird
How old are you?
> 19
What's your firstname?
> Donal
What's your lastname?
> Duck
How old are you?
> 30
What's your firstname?
>
```

fprintf() เขียนลงไฟล์

```
myFile11 - Notepad
File Edit Format View Help
charinya klakhang : 27
angry bird : 19
Donal Duck : 30
```

สรุป

- * ในภาษา C เราสามารถใช้คำสั่งในการและเขียนข้อมูลเกี่ยวกับไฟล์ได้ ซึ่งมีคำสั่งจำนวนมากที่ใช้ภายใต้ไลบรารี <stdio.h>
- * ไฟล์ที่ได้จากการเขียนส่วนใหญ่ จะสามารถเปิดได้ด้วยโปรแกรม Notepad
- * คำสั่งที่ใช้ในการบอกว่าสิ้นสุดไฟล์คือ feof()

แบบฝึกหัดในชั้นเรียน

- * จงเขียนโปรแกรมรับข้อมูลจาก user ดังนี้
แล้วเก็บข้อมูลลงไฟล์ชื่อ lab08.txt เก็บข้อมูลจนกว่าผู้ใช้จะ
กด Ctrl+c (ในที่นี้ให้เก็บข้อมูลอย่างน้อย 5 คน)
- * คำแนะนำ: คล้ายกับการทำงานใน Ex.11

ชื่อ (firstname):
วันเดือนปีเกิด(birthday):
อีเมลล์(email):
น้ำหนัก(weight):
ส่วนสูง(height):

21

ผลลัพธ์ที่ควรจะได้

```
cx "C:\Documents and Settings\Charinya\Desktop\ch8-file\test.exe"
Hello. Enter ctrl+c to end input
Enter your firstname
> Charinya
Enter your birthday (11/07/2530)
> 11/07/2527
Enter your email
> charin@mail.com
Enter your height
> 160
Enter your weight
> 47
Enter your firstname
> Somchai
Enter your birthday (11/07/2530)
> 01/01/2530
Enter your email
> somchai@mail.com
Enter your height
> 170
Enter your weight
> 60
Enter your firstname
> Kae
Enter your birthday (11/07/2530)
> 8/08/2532
```

```
lab08 - Notepad
File Edit Format View Help
Name: Charinya
Birthday: 11/07/2527
Email: charin@mail.com
Weight: 47.000000
Height: 160.000000
=====
Name: Somchai
Birthday: 01/01/2530
Email: somchai@mail.com
Weight: 60.000000
Height: 170.000000
=====
Name: Kae
Birthday: 8/08/2532
Email: kae@mail.com
Weight: 40.000000
Height: 140.000000
=====
```

22