ภาษาซีกับแฟ้มข้อมูล
(File)

Principle of Programming

บุญชู จิตนุพงศ์
วิทยาการคอมพิวเตอร์และเทคโนโลยีสารสนเทศ
คณะทรัพยากรและสิ่งแวดล้อม ม.เกษตรศาสตร์ วิทยาเขตศรีราชา

Agenda

- แฟ้มข้อมูล (File)?
- ขั้นตอนการทำงานร่วมกับแฟ้มข้อมูล
- การเปิดแฟ้มข้อมูล
- การปิดแฟ้มข้อมูล
- การอ่านข้อมูลจากแฟ้มข้อมูล
- การเขียนข้อมูลลงแฟ้มข้อมูล

File?

- แฟ้มข้อมูล (File) คือ ที่เก็บข้อมูลถาวร
- ก่อนหน้านี้ ข้อมูลที่เรารับค่าและแสดงผลจะถูกบันทึกอยู่ในที่เก็บข้อมูลชั่วขณะเท่านั้น เมื่อโปรแกรมทำงานจบ ข้อมูลเหล่านั้นจะหายไป
- แต่ไฟล์ จะทำให้เราสามารถเก็บข้อมูลสำหรับงานของเราไว้ได้อย่างถาวร เมื่อใดที่เราต้องการเข้าถึงข้อมูลที่ถูกบันทึกไว้ เราจึงสั่งให้โปรแกรมเข้าไปนำค่านั้นๆ ออกมาใช้งาน

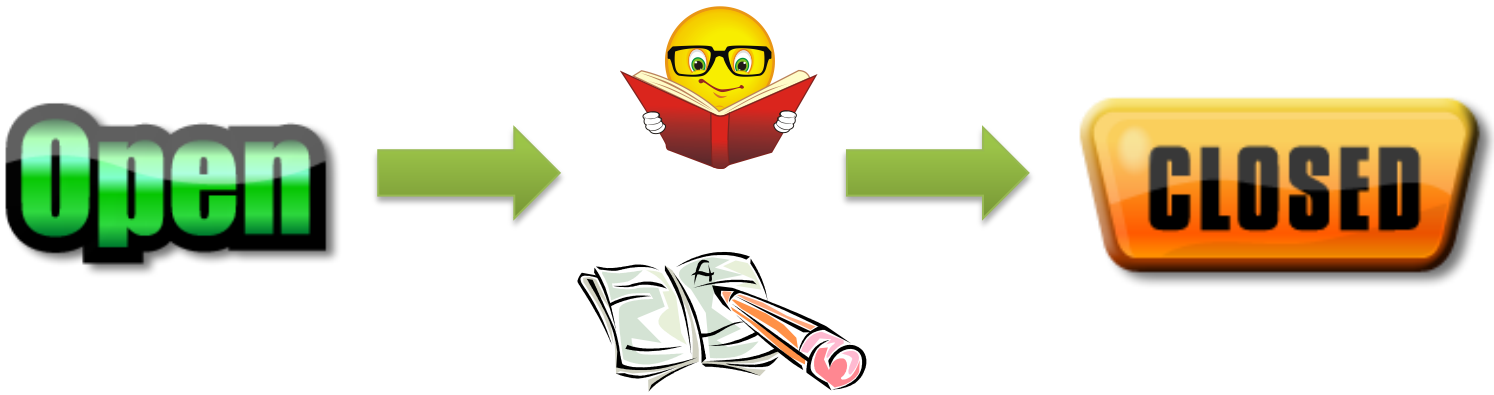
- แบ่งเป็น 2 ประเภท
  - Text Files แฟ้มข้อมูลที่เก็บข้อมูลในรูปแบบตัวอักษร
  - Binary Files แฟ้มข้อมูลสำหรับงานของคอมพิวเตอร์

File? (cont.)

- เราทำงานกับไฟล์ได้ทั้งการอ่าน หรือ เขียนไฟล์
- หากเราสั่งให้โปรแกรมที่เราสร้างขึ้นมาอ่านไฟล์ ข้อมูลจะถูกอ่านขึ้นมาเก็บไว้ในหน่วยความจำก่อนนำไปใช้
- หากเราสั่งให้โปรแกรมที่เราสร้างขึ้นมาเขียนไฟล์ ข้อมูลที่เราจะเขียนจะถูกเก็บไว้ในหน่วยความจำก่อนจะถูกเขียนลงไฟล์
- เราเรียกหน่วยความจำที่พักนี้ว่า บัฟเฟอร์ **(Buffer)**

ขั้นตอนทำงานร่วมกับแฟ้มข้อมูล

- เปิดแฟ้มข้อมูล (Open File)
- เขียน (Write) หรือ อ่าน (Read) ข้อมูล
- ปิดแฟ้มข้อมูล (Close File)



การเปิดแฟ้ม

- ต้องประกาศพอยท์เตอร์สำหรับชี้ไปยังแฟ้มที่เราต้องการอ่าน
- ระบุชื่อแฟ้ม
- บอกจุดประสงค์ของการเปิดแฟ้ม

```
FILE *fp;
fp = fopen("ชื่อแฟ้ม", "จุดประสงค์การเปิด");
```

- `fp` คือชื่อตัวแปรพอยท์เตอร์
- ชื่อแฟ้ม คือ ที่อยู่และชื่อไฟล์ที่เราต้องการเปิด
- จุดประสงค์การเปิด คือ โหมดหรือวิธีการเปิด

การเปิดแฟ้ม (ต่อ)

- จุดประสงค์การเปิดแฟ้มข้อมูล
  - "r" เปิดแฟ้มข้อมูลเพื่อ อ่าน
  - "w" เปิดแฟ้มข้อมูลเพื่อ เขียนทับไฟล์เก่า
  - "a" เปิดแฟ้มข้อมูลเพื่อ เขียนข้อมูลต่อ
  - "r+" เปิดแฟ้มข้อมูลเพื่อ อ่านหรือเขียนข้อมูลทับไฟล์เก่า
  - "w+" เปิดแฟ้มข้อมูลเพื่อ อ่านหรือเขียนข้อมูลทับไฟล์เก่า
  - "a+" เปิดแฟ้มข้อมูลเพื่อ เขียนข้อมูลต่อ
- หากต้องการเขียนแต่ไฟล์นั้นยังไม่มีอยู่ คอมไพเลอร์จะสร้างไฟล์นั้นขึ้นให้เรา

ตัวอย่าง

- เปิดไฟล์ที่เก็บใน ไดร์ฟ C ชื่อ student.txt ขึ้นมาเพื่ออ่านข้อมูล

```
FILE *fp;
fp = fopen("C:\\student.txt", "r");
if(fp==NULL)
    printf("File not exist");
```

- เปิดไฟล์ที่เก็บใน ไดร์ฟ D ในโฟลเดอร์ myfile และมีชื่อไฟล์ว่า employee.txt ขึ้นมาเพื่อเขียนข้อมูลต่อท้าย

```
FILE *fp;
fp = fopen("D:\\myfile\\employee.txt", "a");
```

การปิดแฟ้ม

- เมื่อทำงานกับแฟ้มข้อมูลใดๆ เสร็จเรียบร้อยแล้ว เราควรทำการปิดแฟ้มข้อมูล (close) เพื่อทำการบันทึกข้อมูลสุดท้ายลงในแฟ้มข้อมูล และทำการคืนทรัพยากรให้กับเครื่อง

```
fclose(fp);
```

- `fp` คือชื่อของตัวแปรพอยท์เตอร์ที่เราได้ชี้ไปยังแฟ้มข้อมูล

ตัวอย่าง

```
.
.
.
FILE *fp;
fp = fopen("D:\\myfile\\product.txt", "a");
.
.
.
fclose(fp);
.
.
.
```

การอ่านแฟ้มข้อมูล

- มีฟังก์ชั่น 3 ตัวสำหรับการอ่านข้อมูลที่ถูกบันทึกไว้ในแฟ้ม ได้แก่
- `getc()`
  - อ่านข้อมูลทีละอักขระ
- `fscanf()`
  - อ่านข้อมูล ตัวเลขจำนวนเต็ม, จุดทศนิยม หรือข้อความ
- `fread()`
  - อ่านข้อมูลชนิดข้อมูลแบบโครงสร้าง (structure) หรือ อาร์เรย์ (array) ได้

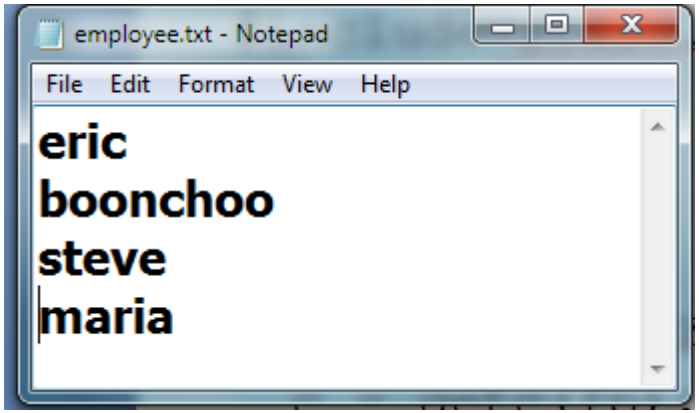
การอ่านแฟ้มข้อมูล (ต่อ)

- การใช้ฟังก์ชั่น `getc()`
- ใช้อ่านอักขระจากแฟ้มข้อมูลได้ทีละ 1 ตัวอักษร

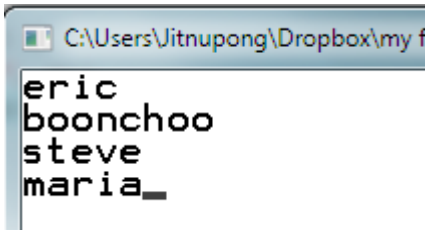
```
ch = getc(input);
```

- โดย
  - `ch` คือตัวแปรประเภทอักขระ (`char`)
  - `input` คือตัวแปรประเภทพอยท์เตอร์ของแฟ้มข้อมูล

ตัวอย่าง




```
#include<stdio.h>
char c;
main(){
   FILE *fp =
   fopen("D:\\myfile\\employee.txt", "r");
   c = getc(fp);
   while(c != EOF) {
      printf("%c", c);
      c = getc(fp);
   }
}
```



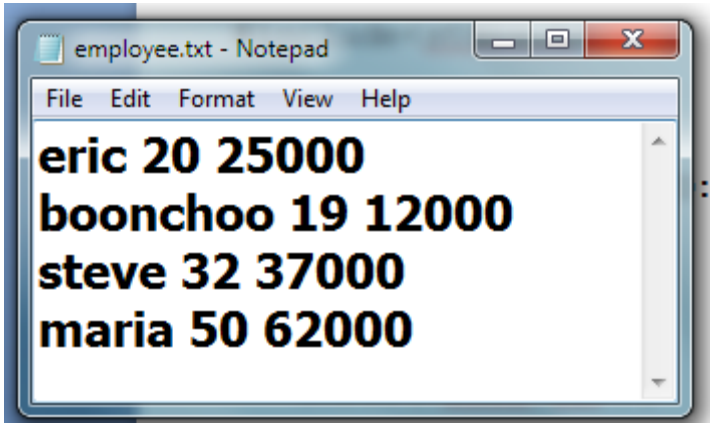
การอ่านแฟ้มข้อมูล (ต่อ)

- การใช้ฟังก์ชั่น `fscanf()`
- ใช้อ่านข้อมูลจากแฟ้มข้อมูลในรูปแบบตัวเลขจำนวนเต็ม, ตัวเลขจุดทศนิยมและข้อความได้

```
fscanf( fp, control string, variable list);
```

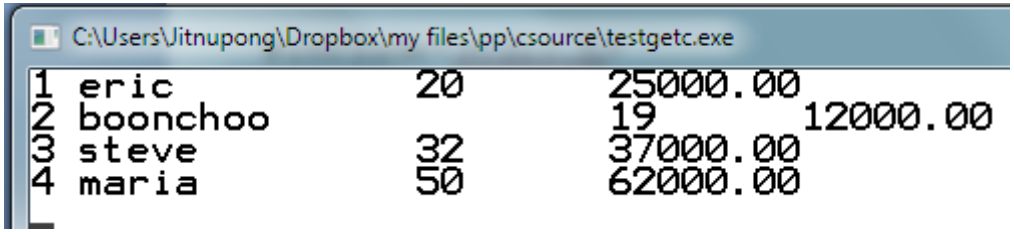
- โดย
  - `fp` คือชื่อตัวแปรพอยท์เตอร์
  - `control string` คือรูปแบบของการอ่านค่าจากแฟ้ม
  - `variable list` คือรายชื่อตัวแปรสำหรับเก็บค่า

ตัวอย่าง




```
#include<stdio.h>

main(){
   FILE *fp = fopen("D:\\myfile\\employee.txt", "r");
   char name[10];
   int age, i = 1;
   float salary;
   while(!feof(fp)){
        fscanf(fp, "%s %d %f\n", name, &age, &salary);
        printf("%d %s\t\t%d\t%.2f\n", i, name, age, salary);
        i++;
   }
   fclose(fp);
   getch();
}
```



การอ่านแฟ้มข้อมูล (ต่อ)

- การใช้ฟังก์ชั่น `fread()`
- ใช้อ่านข้อมูลจากแฟ้มข้อมูลที่อยู่ในรูปแบบข้อมูลชนิดโครงสร้างหรืออาร์เรย์ได้

```
fread(&var, size, n, fp);
```

- โดย
  - `var` คือตัวแปรที่เราต้องการเก็บสิ่งที่อ่านขึ้นมาจากแฟ้มข้อมูล
  - `size` ขนาดของตัวแปร
  - `n` จำนวนรอบที่ต้องการอ่าน
  - `fp` คือชื่อตัวแปรพอยท์เตอร์ที่เราชี้ไปยังแฟ้มข้อมูล

การเขียนแฟ้มข้อมูล

- มีฟังก์ชั่น 3 ตัวสำหรับการเขียนข้อมูลเพื่อบันทึกไว้ในแฟ้มข้อมูล ได้แก่
- `putc()`
  - เขียนข้อมูลทีละอักขระ
- `fprintf()`
  - เขียนข้อมูล ตัวเลขจำนวนเต็ม, จุดทศนิยม หรือข้อความ
- `fwrite()`
  - เขียนข้อมูลชนิดข้อมูลแบบโครงสร้างหรืออาร์เรย์ได้

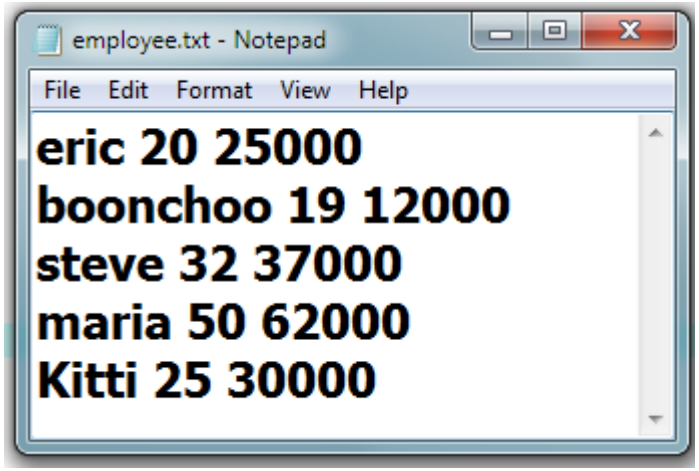
การเขียนแฟ้มข้อมูล (ต่อ)

- การใช้ฟังก์ชั่น `putc()`
- ใช้เขียนอักขระลงแฟ้มข้อมูลได้ทีละ 1 ตัวอักษร

```
putc(single_char,fp);
```

- โดย
  - `single_char` คือตัวแปรหรือค่าอักขระ
  - `fp` คือตัวแปรพอยท์เตอร์ที่ชี้ไปยังแฟ้มข้อมูล

ตัวอย่าง




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

main(){
        FILE *fp = fopen("D:\\myfile\\employee.txt", "a");
        char newEmployee[] = "Kitti 25 30000";
        int point = 0;
        putc('\n', fp);
        do {
            putc(newEmployee[point], fp);
            point++;
        }while(point<strlen(newEmployee));
        fclose(fp);
        getch();
}
```

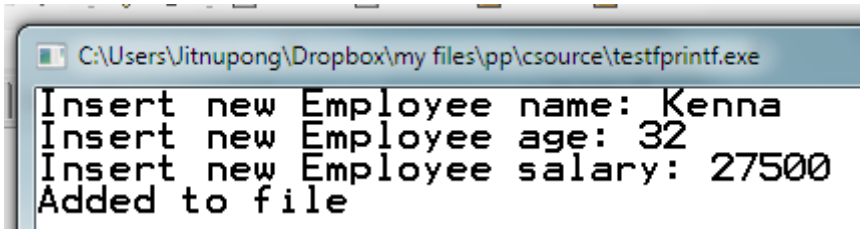
การเขียนแฟ้มข้อมูล (ต่อ)

- การใช้ฟังก์ชั่น `fprintf()`
- ใช้เขียนข้อมูลในรูปแบบตัวเลขจำนวนเต็ม, ตัวเลขจุดทศนิยมและข้อความลงแฟ้มข้อมูลได้

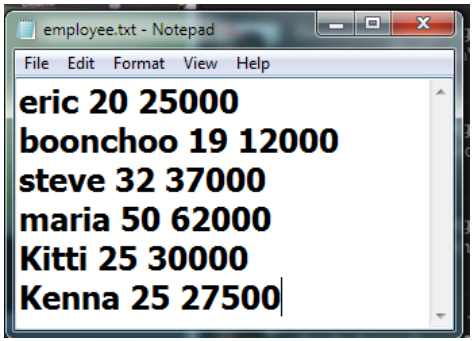
```
fprintf (fp,control string,variable list);
```

- โดย
  - `fp` คือชื่อตัวแปรพอยท์เตอร์
  - `control string` คือรูปแบบของการอ่านค่าจากแฟ้ม
  - `variable list` คือรายชื่อตัวแปรสำหรับเก็บค่า

ตัวอย่าง




```
#include<stdio.h>
main() {
      FILE *fp = fopen("D:\\myfile\\employee.txt", "a");
      char name[20];
      int age;
      float salary;
      printf("Insert new Employee name: ");
      scanf("%s", name);
      printf("Insert new Employee age: ");
      scanf("%d", &age);
      printf("Insert new Employee salary: ");
      scanf("%f", &salary);
      fprintf(fp, "\n%s %d %.0f", name, age, salary);
      printf("Added to file");
      getch();
}
```



การเขียนแฟ้มข้อมูล (ต่อ)

- การใช้ฟังก์ชั่น `fwrite()`
- ใช้เขียนข้อมูลที่อยู่ในรูปแบบข้อมูลชนิดโครงสร้างหรืออาร์เรย์ลงแฟ้มข้อมูลได้

```
fwrite(&var, size, n, fp);
```

- โดย
  - `var` คือตัวแปรที่เราต้องการบันทึกลงแฟ้มข้อมูล
  - `size` ขนาดของตัวแปร
  - `n` จำนวนรอบที่ต้องการเขียน
  - `fp` คือชื่อตัวแปรพอยท์เตอร์ที่เราชี้ไปยังแฟ้มข้อมูล

```
#include<stdio.h>
typedef struct{
	char name[10];
	int age;
	float salary;
}EMPLOYEE;
main(){
	FILE *fp = fopen("D:\\myfile\\employee.txt", "w");
	EMPLOYEE writeEmp, readEmp;
	printf("Insert new Employee name: ");
	scanf("%s", writeEmp.name);
	printf("Insert new Employee age: ");
	scanf("%d", &writeEmp.age);
	printf("Insert new Employee salary: ");
	scanf("%f", &writeEmp.salary);
	fwrite(&writeEmp, sizeof(EMPLOYEE), 1, fp);
	fclose(fp);

	fp = fopen("D:\\myfile\\employee.txt", "r");
	fread(&readEmp, sizeof(EMPLOYEE), 1, fp);
	printf("%s", readEmp.name);
	printf("%d", readEmp.age);
	printf("%.2f", readEmp.salary);
	fclose(fp);
	getch();
}
```