

# การจัดการเพิ่มข้อมูล

โดย อาจารย์ภาณุวัฒน์ เมฆะ

สาขาวิชาวิทยาการคอมพิวเตอร์  
คณะวิทยาศาสตร์ มหาวิทยาลัยแม่โจ้


## การจัดเพิ่มข้อมูล

- ❖ เป็นการนำข้อมูลที่มีความเกี่ยวข้องกัน นำมาจัดเป็นแฟ้มข้อมูล
- ❖ โดยจะมีการอ่านข้อมูล บันทึกข้อมูล และการเพิ่มข้อมูล ลงในแฟ้มข้อมูล
- ❖ การจัดเก็บแฟ้มข้อมูล สามารถแบ่งออกได้เป็น 2 ประเภท ได้แก่
  - ❖ Sequential File
    - ❖ เป็นการจัดเก็บข้อมูลแบบเรียงลำดับ เริ่มตั้งแต่เรคคอร์ด (Record) แรกไปจนถึงเรคคอร์ดสุดท้าย
    - ❖ การเรียกใช้ข้อมูลจะต้องเริ่มตั้งแต่เรคคอร์ดแรก ไปจนถึงเรคคอร์ดสุดท้าย
  - ❖ Random File
    - ❖ เป็นการจัดเก็บข้อมูลแบบสุ่มข้อมูล โดยสามารถกำหนดเรคคอร์ดใดเป็นคีย์ (Key) สำหรับเรียกใช้ข้อมูล

# ชนิดของแฟ้มข้อมูล

## 1. Text file

- ❖ เป็นการจัดเก็บแฟ้มข้อมูลในลักษณะข้อความ (Text)
- ❖ ซึ่งจะต้องแปลงข้อความไปตามรหัสแอสกี (ASCII CODE) มีความยาวบรรทัดละ 256 ตัวอักษร ท้ายบรรทัดจะมีรหัส \n (new line) หมายถึง การ carriage return หรือขึ้นบรรทัดใหม่

ไฟล์แบบตัวอักษรหรือ Text File เป็นไฟล์ที่มีแต่ตัวอักษร เปิดอ่านด้วย notepad ได้รู้เรื่อง โดยการดูนามสกุลของไฟล์ หรืออาจดูจากไอคอน คือ 

# ชนิดของแฟ้มข้อมูล

- ตัวอย่างนามสกุลของ ชนิดของแฟ้มข้อมูล Text File มีดังนี้
  - .txt ไฟล์ข้อความ
  - .bak ไฟล์สำรอง
  - .log ไฟล์ Log
  - .ini ไฟล์เก็บค่า profile
  - .c ไฟล์ซอร์สโค้ดโปรแกรมภาษา C
  - .php ไฟล์ซอร์สโค้ดโปรแกรมภาษา PHP

นอกจากนี้ยังรวมไปถึงซอร์สโค้ดของโปรแกรมภาษาต่างๆ ซึ่งจะเก็บอยู่ในรูป Text File ทั้งหมด โดยระบบนามสกุลเป็นภาษานั้นๆ

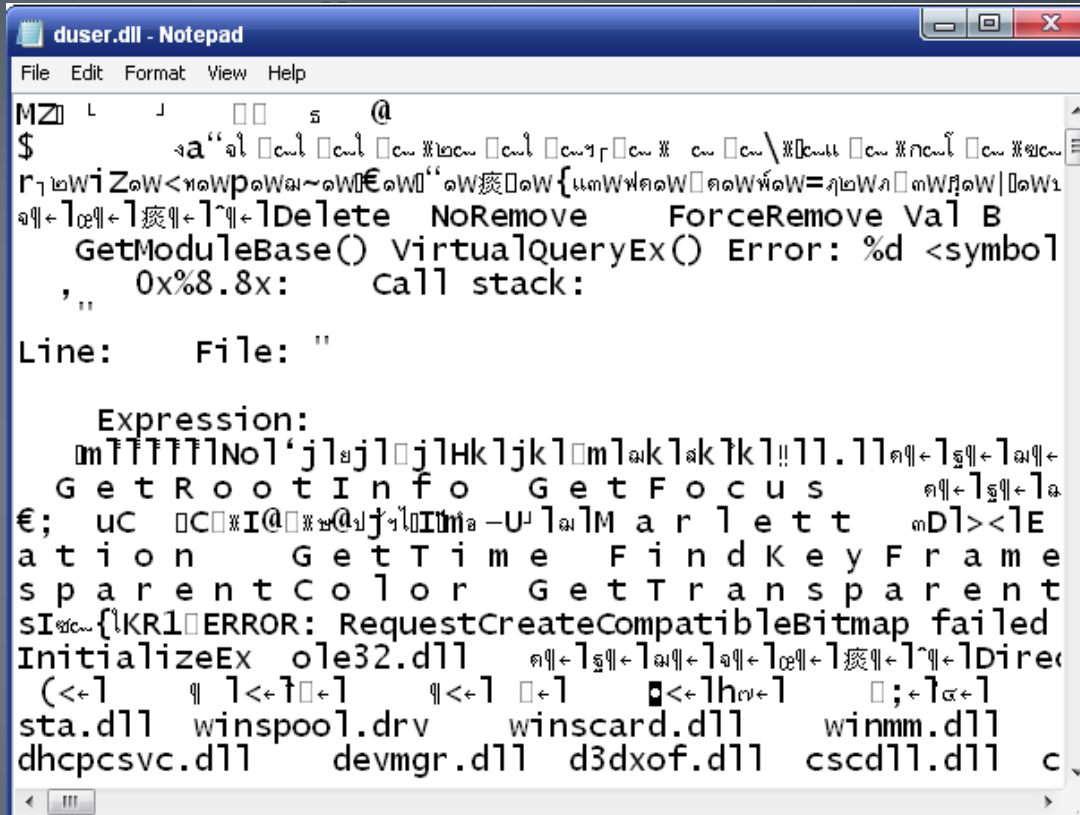
# ชนิดของแฟ้มข้อมูล

## 2. Binary file

- ❖ เป็นการเก็บแฟ้มข้อมูลในระบบเลขฐานสอง หรือ Binary Code
- ❖ ไม่ต้องแปลงเหมือนกับ Text file เพราะข้อมูลในระบบคอมพิวเตอร์ถูกจัดเก็บในระบบเลขฐานสองอยู่แล้ว

- ตัวอย่างไฟล์ที่เก็บข้อมูลแบบไบนารี โดยจะเก็บข้อมูลเป็น ไบต์ เช่น
  - ไฟล์รูปภาพ เช่น bmp,jpg,gif
  - ไฟล์เอกสาร ของโปรแกรมต่างๆ .doc , .xls , .ppt , .rdw
  - ไฟล์หนังและไฟล์เพลง เช่น .wav , .mp3 , .wma , .wmv
  - ไฟล์โปรแกรม เช่น .exe , .dll

# ชนิดของแฟ้มข้อมูล



- ตัวอย่างการเปิดไฟล์แบบ Binary ด้วย notepad

# หลักการเขียน โปรแกรมการจัดแฟ้มข้อมูล

## 1. การเปิดแฟ้มข้อมูล

- ❖ เป็นขั้นตอนแรกที่ต้องทำ ในการเขียนโปรแกรมการจัดแฟ้มข้อมูล
- ❖ เหตุผลที่จะต้องเปิดแฟ้มข้อมูล มีอยู่ 2 ประการ คือ
  - ❖ ประการแรก เพื่อกำหนดให้ File Pointer ชี้ไปที่ตำแหน่งแรก
  - ❖ ประการที่สอง เพื่อสร้างเนื้อที่ความจำชั่วคราว (Buffer) สำหรับทำการอ่านข้อมูล บันทึกข้อมูล และเพิ่มข้อมูล จะทำให้การทำงานทำได้เร็วขึ้น ไม่ต้องกระทำกับดิสก์โดยตรง

## 2. การอ่านข้อมูล / การบันทึกข้อมูล

- ❖ ขั้นตอนต่อจากการเปิดแฟ้มข้อมูล คือ การเรียกใช้ฟังก์ชัน เพื่ออ่านข้อมูล บันทึกข้อมูล และเพิ่มข้อมูล ลงแฟ้มข้อมูล

## 3. การปิดแฟ้มข้อมูล

- ❖ ขั้นตอนนี้ เป็นการนำข้อมูลที่อยู่บนเนื้อที่ความจำชั่วคราว (Buffer) จัดเก็บลงดิสก์ (Disk)

## การเปิดเพิ่มข้อมูล

- ในการประมวลผลร่วมกับเพิ่มข้อมูลนั้น การดำเนินการแรกที่จะต้องทำก่อนงานอื่นๆ คือการเปิดไฟล์เพื่อเลือกไฟล์ที่จะประมวลผล และสร้างตัวชี้ตำแหน่งไฟล์ขึ้น ซึ่งในการเปิดไฟล์จะต้องใช้ 2 คำสั่ง
  - คำสั่งแรก เป็นการสร้างตัวพอยเตอร์ชนิด file เพื่อใช้เป็นตัวชี้ตำแหน่งไฟล์
  - คำสั่งที่สอง คือ คำสั่งเปิดไฟล์และกำหนดตัวชี้ตำแหน่งไฟล์ให้กับตัวแปรพอยเตอร์



## การเปิดเพิ่มข้อมูล (Open file)

(page 1/2)

- ❖ เป็นการบ่งบอกให้คอมไพเลอร์ (Compiler) ทราบว่าจะมีการอ่านและบันทึกข้อมูล
- ❖ โดยมีตัวแปรพอยน์เตอร์ (pointer) เป็นตัวชี้เพิ่มข้อมูล
- ❖ นอกจากนั้นยังเป็นการจองเนื้อที่ความจำชั่วคราว (Buffer)

### ❖ รูปแบบ

```
FILE *fp
```

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

FILE หมายถึง ระบุตัวแปรพอยน์เตอร์เป็นประเภทเพิ่มข้อมูล

fp หมายถึง ตัวแปรพอยน์เตอร์

Fopen() หมายถึง ฟังก์ชันที่ใช้ในการเปิดเพิ่มข้อมูล

FileName หมายถึง ชื่อเพิ่มข้อมูลที่ต้องการเปิดใช้

Mode หมายถึง การเลือกชนิดของการเปิดเพิ่มข้อมูล

## การเปิดแฟ้มข้อมูล

- ในการเขียนคำสั่งเพื่อเปิดไฟล์ จะต้องระบุโหมดของไฟล์ด้วย เพื่อให้ตัวแปลภาษา C เข้าใจว่าเราต้องการเปิดไฟล์เพื่อทำงานใด โดยโหมดของไฟล์แสดงดังตารางต่อไปนี้

โหมดการเปิดไฟล์	ความหมาย
r	เปิดไฟล์เพื่ออ่านข้อมูลอย่างเดียว
w	เปิดไฟล์เพื่อเขียนข้อมูลทับลงไปทับข้อมูลเก่าในไฟล์ โดยข้อมูลเก่าจะถูกลบทิ้ง
a	เปิดไฟล์สำหรับเขียนข้อมูลต่อท้ายข้อมูลเดิมในไฟล์
r+	เปิดไฟล์สำหรับอ่านและเขียนข้อมูลลงในไฟล์
w+	เปิดไฟล์สำหรับอ่านและเขียนข้อมูลใหม่ทับข้อมูลเดิม
a+	เปิดไฟล์สำหรับอ่านและเขียนข้อมูลใหม่ต่อท้ายข้อมูลเดิม

## การเปิดแฟ้มข้อมูล

การเปิดไฟล์ในโหมด `r` ไฟล์ที่จะเปิดต้องมีอยู่จริงจึงจะเปิดได้ ส่วนโหมดอื่นๆ ถ้าไฟล์ที่ระบุชื่อยังไม่ได้อยู่ ระบบจะสร้างไฟล์ใหม่ขึ้นมา

การเขียนคำสั่งเพื่อเปิดไฟล์ ถ้าเปิดไฟล์ไม่สำเร็จ ไม่มีปัญหา หรือข้อผิดพลาดใดๆ สิ่งจะได้กลับมาก็คือ ตำแหน่งตัวชี้ไฟล์ แต่หากไม่สามารถเปิดไฟล์ขึ้นมาได้ ค่าที่ได้กลับมาก็คือ `NULL`

# การเปิดเพิ่มข้อมูล

`FILE *fp;`

← - สร้างตัวแปรพอยเตอร์ชนิด file ชื่อ fp

`fp=fopen("d:/exercise/text.txt", "r");` ← - เปิดไฟล์ชื่อ text.txt ในไดร์ฟ D โฟลเดอร์ exercise เพื่ออ่านข้อมูลจากไฟล์

`fp=fopen("c:/temp.dat", "w");` ←

`fp=fopen("ex_10.exe", "a+");` ←

`fp=fopen("c:/documents/news.txt", "w+");` ←

## การปิดแฟ้มข้อมูล (Close file)

- ❖ หลังจากทำการเปิดแฟ้มข้อมูล อ่านข้อมูลและบันทึกข้อมูลเรียบร้อยแล้ว ขั้นตอนสุดท้ายคือการปิดแฟ้มข้อมูล โดยใช้ฟังก์ชัน `fclose()` เพื่อนำข้อมูลที่อยู่บน Buffer จัดเก็บลงดิสก์
- ❖ หากปิดแฟ้มข้อมูลสำเร็จจะให้ค่าเป็นศูนย์
- ❖ หากปิดแฟ้มข้อมูลไม่สำเร็จ จะส่งค่า EOF (End Of File) กลับคืนมาให้
- ❖ รูปแบบ

```
fclose(fp)
```

fp หมายถึง ตัวแปรพอยน์เตอร์

# ฟังก์ชันในการจัดการไฟล์ในภาษา C

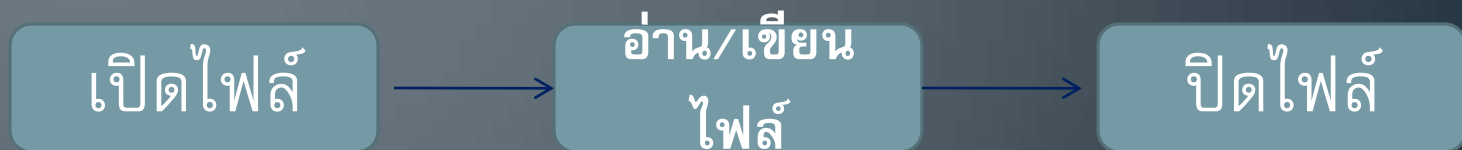
- ฟังก์ชันเปิดและปิดไฟล์
  - fopen ใช้ในการเปิดไฟล์
  - fclose ใช้ในการปิดไฟล์
- ฟังก์ชันที่ใช้ในการอ่านข้อมูลจากไฟล์
  - fgetc อ่านข้อมูลที่ละตัวอักษร
  - fgets อ่านข้อมูลมาเป็นข้อความ
  - fscanf อ่านข้อมูลเป็นข้อความเหมือนกับ scanf
  - fread อ่านข้อมูลแบบไบต์

# ฟังก์ชันในการจัดการไฟล์ในภาษา C

- ฟังก์ชันที่ใช้ในการเขียนข้อมูลลงไฟล์
  - fputc           เขียนข้อมูลที่ละตัวอักษร
  - fputs           เขียนข้อมูลแบบข้อความ
  - fprintf         เขียนข้อมูลเป็นข้อความเหมือนกับ printf
  - fwrite          เขียนข้อมูลเป็น ไบต์
- ฟังก์ชันช่วยในการจัดการไฟล์
  - feof           ใช้ตรวจสอบการสิ้นสุดไฟล์
  - ftell           ใช้ตรวจสอบการอ่านตำแหน่งของไฟล์
  - fseek          ใช้ในการเลื่อนไปยังตำแหน่งที่ต้องการภายในไฟล์

## การเปิดไฟล์

- การตรวจสอบจุดสิ้นสุดของไฟล์สามารถทำได้ 2 วิธี โดยถ้าเป็นไบนารีไฟล์ จะต้องเรียกใช้ฟังก์ชัน **feof()** เพื่อตรวจสอบจุดสิ้นสุดของไฟล์ แต่ถ้าเป็น Text File เมื่ออ่านข้อมูลจนจบแล้วจะได้ค่า **EOF** (End Of File) ออกมา หรือจะเรียกใช้ฟังก์ชัน **feof()** ก็ได้เช่นเดียวกัน





## การลบไฟล์

- สำหรับไฟล์ที่ไม่จำเป็นต้องใช้งานแล้ว สามารถลบไฟล์ดังกล่าวเพื่อประหยัดพื้นที่ในฮาร์ดดิสก์ได้ โดยใช้คำสั่ง **remove()** ซึ่งมีรูปแบบการเรียกใช้ดังนี้

```
remove("ชื่อไฟล์");
```

ตัวอย่าง เช่น

```
remove("c:/data.txt");
```

## การเปลี่ยนชื่อไฟล์

- การเปลี่ยนชื่อไฟล์สามารถทำได้ โดยเรียกใช้คำสั่ง **rename()** เพื่อเปลี่ยนชื่อ โดยมีรูปแบบการเรียกใช้คำสั่งดังนี้

```
rename(“ชื่อไฟล์เดิม”, “ชื่อไฟล์ใหม่”);
```

ตัวอย่าง เช่น

```
rename(“menu.dat”, “content.dat”);
```

```
rename(“c:/doc/data.txt”, “d:/lab/test.txt”);
```

# สรุปเนื้อหาท้ายบท

## ❖ 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)
```

# ตัวอย่างโปรแกรมบันทึกเพิ่มความ

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
main()
{
FILE *fp;
fp=fopen("d:\\temp\\test.txt","w");
char buffer[1024];
do
{
gets(buffer);
fprintf(fp,"%s\n",buffer);

}while(strcmp(buffer,"")!=0);
fclose(fp);

}
```

# ตัวอย่างโปรแกรมเปิดเพิ่มข้อความ

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
main()
{
FILE *fp;
fp=fopen("d:\\temp\\test.txt","r");
char buffer[1024];
while(!feof(fp))
{
    fscanf(fp,"%s",buffer);
    printf("%s\n",buffer);
}
fclose(fp);
}
```

# ตัวอย่างโปรแกรมบันทึกเพิ่มไบนารี

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
main()
{
FILE *fp;
fp=fopen("d:\\temp\\test.bin","w");
int a;
do
{
scanf("%d",&a);
fwrite(&a,sizeof(a),1,fp);

}while(a!=0);
fclose(fp);
}
```

# ตัวอย่างโปรแกรมเปิดแฟ้มไบนารี

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
main()
{
FILE *fp;
fp=fopen("d:\\temp\\test.bin","r");
int a;
while(!feof(fp))
{
fread(&a,sizeof(a),1,fp);
printf("%d\n",a);
}
fclose(fp);
}
```

1. จงเขียนโปรแกรม สร้างไฟล์ **text** และ **binary** เก็บข้อมูล

กรณีกต 1 เก็บตัวเลข 3ค่า โดยให้ทำเป็น ฟังก์ชัน

กรณีกต 2 ให้อ่านไฟล์ ของไฟล์ที่บันทึก

กรณีกต 3 ให้อ่านไฟล์แล้วออกจากโปรแกรม

1) **Save data -> text or binary**

2) **Load data -> text or binary**

3) **Exit**

2. จงเขียนโปรแกรม นับตัวอักษรในไฟล์ว่ามีตัว **a , g , z** กี่ตัว

3. จงเขียนโปรแกรม ฟังก์ชันรับค่าคำนวณหาค่าเฉลี่ย 5ค่า หลังจากนั้นเอาผลลัพธ์ไปเก็บบนไฟล์เป็น **binary** แล้วเขียนโปรแกรมอ่านไฟล์ นั้น ๆ ด้วย

1) **Save data**

2) **Load data**