

บทที่ 11

การจัดการไฟล์





4 ขั้นตอนการทำงานของ Software

❖ Input

❖ Processing

❖ Output

❖ Storage 



ทำไมต้องใช้ไฟล์

❖ ข้อมูลที่เก็บอยู่ในตัวแปร ไม่สามารถเก็บข้อมูลได้อย่างถาวร จะหายไปหลังจากปิดโปรแกรม





ชนิดของไฟล์

❖ ไฟล์ข้อความ (Text File)

- เก็บข้อมูลแบบอักขระ
- แยกแต่ละบรรทัดด้วย \n
- สามารถเปิดดูเนื้อหาในไฟล์ได้ด้วย text editor

❖ ไฟล์ฐานสอง (Binary File)

- เก็บข้อมูลในรูปแบบของภาษาเครื่อง
- **ไม่สามารถ**เปิดดูเนื้อหาในไฟล์ได้ด้วย text editor



รูปแบบการเปิดไฟล์

โหมด	การทำงาน
r (read)	เปิดไฟล์เพื่ออ่านอย่างเดียว - ถ้าไฟล์ที่กำหนดมีอยู่จริง จะเริ่มอ่านข้อมูลที่ตำแหน่งต้นไฟล์ - ถ้าไฟล์ที่กำหนดไม่มีจริง จะแสดงข้อผิดพลาด
w (write)	เปิดไฟล์เพื่อเขียน - ถ้าไฟล์ที่กำหนดมีอยู่แล้ว จะเริ่มเขียนข้อมูลใหม่ ทับข้อมูลเดิม - ถ้าไฟล์ที่กำหนดไม่มีจริง จะสร้างไฟล์นั้นขึ้นมาใหม่
a (append)	เปิดไฟล์ เพื่อเขียนข้อมูลต่อท้าย - ถ้าไฟล์ที่กำหนดมีอยู่แล้ว จะเริ่มเขียนข้อมูลที่ตำแหน่งสุดท้ายของไฟล์ - ถ้าไฟล์ที่กำหนดไม่มีจริง จะสร้างไฟล์ขึ้นมาใหม่

- หากเป็นไฟล์แบบ binary จะต่อท้ายด้วย b เช่น rb, wb, ab
- หากต้องการเปิดไฟล์ทั้งอ่านและเขียนจะต่อท้ายด้วยเครื่องหมาย + เช่น r+, w+, a+



File Pointer

- ❖ File Pointer คือ ตัวแปร Pointer ที่ใช้สำหรับชี้ไปยังข้อมูลในส่วนต่างๆ ของไฟล์
- ❖ ส่วนท้ายไฟล์ เรียกว่า EOF ย่อมาจาก End of File เป็นคำสั่งในภาษาซี ซึ่งใช้แทนเลข -1
- ❖ นักพัฒนาไม่จำเป็นต้องใช้สัญลักษณ์ + หรือ - เพื่อเลื่อน File Pointer เพราะคำสั่งที่เกี่ยวข้องกับไฟล์ทุกคำสั่งจะเลื่อนไฟล์ไปยังตำแหน่งถัดไปให้อัตโนมัติ

ต้นไฟล์

ตัวอย่างไฟล์

เด็กภาคคอม คณะอะไรสักอย่าง มีความตั้งใจจริงในการเรียนเขียนโปรแกรม ทำการบ้านโดยไม่ copy ไฟล์จากใคร เรียนบรรยายก็ไม่คุย ไม่แอบหลับ ไม่ selfie ไม่ live เรียนแล้วก็จดจ่อที่หน้าจอ ไม่เปิดเฟส ไม่เปิด YouTube ความฝันอันสูงสุดของพวกเขา คือ ผ่านวิชา Calculus 1 และ 2

ท้ายไฟล์

EOF (End of File)



ขั้นตอนการทำงานกับไฟล์

1. ประกาศตัวแปร File Pointer
2. เรียกฟังก์ชันเปิดไฟล์
3. ตรวจสอบว่าเปิดไฟล์สำเร็จหรือไม่
4. อ่าน หรือ เขียนไฟล์
5. ปิดไฟล์



การประกาศตัวแปร File Pointer

รูปแบบ

FILE *ชื่อตัวแปร;

ตัวอย่าง

FILE *fp;



การเรียกฟังก์ชันเปิดไฟล์

❖ รูปแบบฟังก์ชันเปิดไฟล์

ฟังก์ชัน fopen จะ return ตำแหน่งของไฟล์
กลับมาจึงต้องนำตัวแปร file pointer มารอรับ

```
ตัวแปร _File_Pointer = fopen("Pathและชื่อไฟล์", "mode");
```

❖ ตัวอย่าง

```
fp = fopen("d:/myprogram/eiei/data.txt", "w");
```

❖ สามารถใช้คำสั่งเปิดได้หลายไฟล์พร้อมๆกัน แต่ควรประกาศตัวแปร File Pointer ใหม่



การตรวจสอบว่าเปิดไฟล์สำเร็จหรือไม่

❖ การเปิดไฟล์ที่ไม่สำเร็จ มี 2 กรณี

- Path ที่ระบุไม่ถูกต้อง หรือไม่มีอยู่จริง
- สั่งเปิดไฟล์เพื่ออ่านอย่างเดียว (r) แต่ไม่มีไฟล์นั้นอยู่

❖ ฟังก์ชัน fopen หากไม่สามารถเปิดไฟล์สำเร็จจะ return ค่าเป็น NULL ซึ่งหมายถึง ไม่มีค่า

❖ ตัวอย่างการตรวจสอบด้วยเงื่อนไข if

```
fp = fopen("d:/data.txt", "w");  
if (fp==NULL) {  
    printf("Error cannot open file");  
    exit(1); // คำสั่งหยุดการทำงานของโปรแกรมทั้งหมด  
}
```



การปิดไฟล์

❖ การปิดไฟล์เป็นการปลดล็อกการทำงานของไฟล์ เพื่อให้โปรแกรมอื่นๆ สามารถมาทำงานกับไฟล์นั้นได้

❖ รูปแบบ

```
fclose(ตัวแปร_File_Pointer);
```

❖ ตัวอย่าง

```
fclose(fp);
```



โค้ดการทำงานกับไฟล์ข้อมูล

```
#include <stdio.h>
void main() {
    // 1. ประกาศตัวแปร File Pointer
    FILE *fp;

    // 2. เปิดไฟล์
    fp = fopen("d:/data.txt", "w"); // กำหนดเป็นโหมดเขียน

    // 3. ตรวจสอบว่าเปิดไฟล์สำเร็จหรือไม่
    if (fp==NULL) {
        printf("Error cannot open file"); // แสดงข้อผิดพลาดออกทางหน้าจอ
        exit(1); // คำสั่งหยุดการทำงานของโปรแกรมทั้งหมด
    }

    // 4. ประมวลผลไฟล์ โดยการอ่านหรือเขียนที่นี่

    // 5. ปิดไฟล์
    fclose(fp);
}
```



การเขียน Character ลงไฟล์

❖ รูปแบบ

```
fputc(อักขระหรือตัวแปรอักขระ, ตัวแปร_File_Pointer);
```

❖ ตัวอย่าง

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



ตัวอย่าง

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

```
void main() {
```

```
    FILE *fp;
```

```
    fp = fopen("d:/data.txt", "w");
```

```
    if (fp==NULL) {
```

```
        printf("Error cannot open file");
```

```
        exit(1);
```

```
    }
```

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

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

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

```
    fclose(fp);
```

```
}
```

เปิดไฟล์ data.txt ในโหมดเขียนไฟล์

fp

KKU

เขียน Character ลงไฟล์

ทีละตัว

ไฟล์ d:/data.txt



การอ่าน Character จากไฟล์

❖ รูปแบบฟังก์ชัน

```
fgetc(ตัวแปร_File_Pointer);
```

❖ ค่าส่งกลับ

อักขระตัวต่อไป

❖ ตัวอย่าง

```
c = fgetc(fp);
```



ตัวอย่าง

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

```
void main() {
```

```
FILE *fp;
```

```
fp = fopen("d:/data.txt", "r");
```

```
char c; // l = true
```

```
while (1) {
```

```
    c = fgetc(fp);
```

```
    if (c == EOF)
```

```
        break;
```

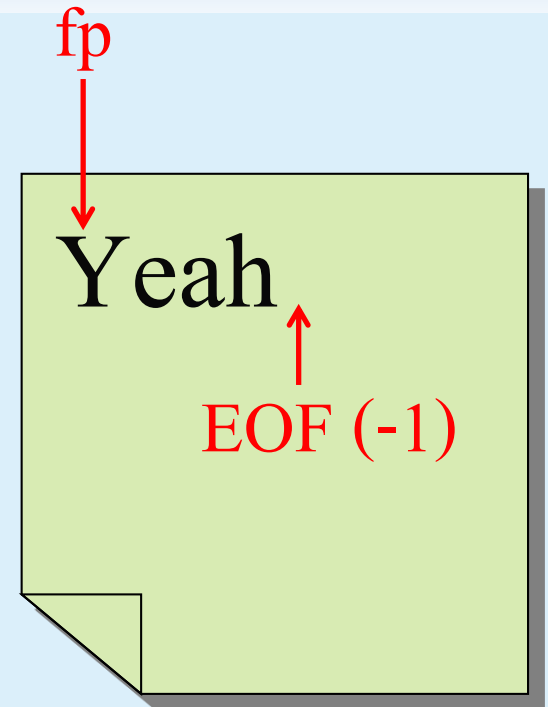
```
    printf("%c\n", c);
```

```
}
```

```
fclose(fp);
```

```
}
```

-1
c



ไฟล์ d:/data.txt



ผลลัพธ์ที่แสดงทางหน้าจอ



การเขียนไฟล์ตามรูปแบบที่กำหนด

❖ รูปแบบ

```
fprintf(ตัวแปร_File_Pointer, รูปแบบการเขียน, รายการตัวแปร);
```

❖ ตัวอย่าง

```
fprintf(fp, "%d %s %s\n", i, fname, lname);
```



ตัวอย่าง

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

```
void main() {
```

โหมดเขียนไฟล์



```
FILE *fp;
```

```
fp = fopen("d:/data.txt", "w");
```

```
int i;
```

```
char fname[10], lname[10];
```

```
for (i=1; i<=3; i++) {
```

```
printf("First Name :");
```

```
scanf("%s", fname);
```

```
printf("Last Name :");
```

```
scanf("%s", lname);
```

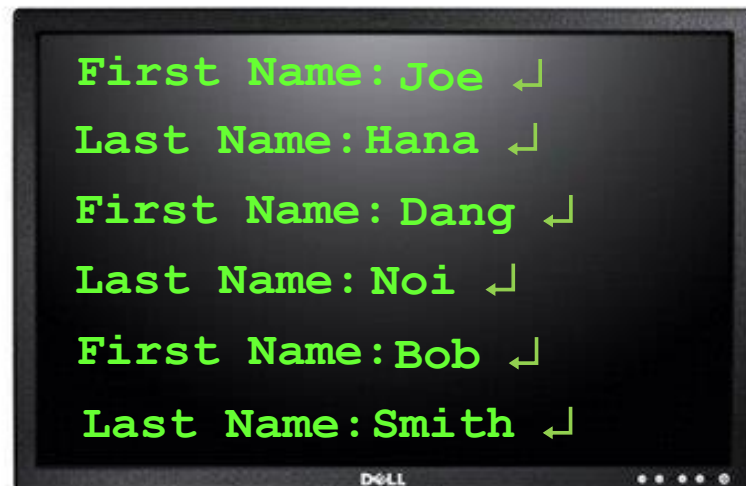
```
fprintf(fp, "%d %s %s\n", i, fname, lname);
```

```
}
```

```
fclose(fp);
```

เขียนลงไฟล์ตามรูปแบบที่กำหนด

```
}
```

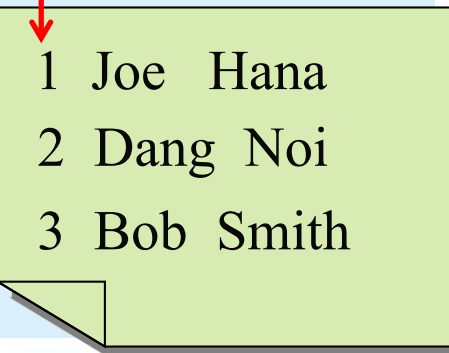


รับค่าทางคีย์บอร์ด

4

i

fp



ไฟล์ d:/data.txt



การอ่านไฟล์ตามรูปแบบที่กำหนด

❖ รูปแบบ

```
fscanf(ตัวแปร_File_Pointer, รูปแบบการอ่าน, รายการ address ของตัวแปร);
```

❖ ตัวอย่าง

```
fscanf(fp, "%d %s %s\n", &id, &fname, &lname);
```



ฟังก์ชันตรวจสอบ EOF

❖ รูปแบบ

feof (ตัวแปร_File_Pointer)

❖ ค่าส่งกลับ

0 - File Pointer **ไม่ได้ชี้** อยู่จุดสิ้นสุดของไฟล์

1 - File Pointer **ชี้** อยู่ที่จุดสิ้นสุดของไฟล์



ตัวอย่าง

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

```
void main() {
```

โหมดอ่านไฟล์

```
FILE *fp;
```

```
fp = fopen("d:/data.txt", "r");
```

fp

1 Joe Hana
2 Dang Noi
3 Bob Smith

ไฟล์ d:/data.txt

```
int id;
```

```
char fname[10], lname[10];
```

อ่านไฟล์ตาม
รูปแบบที่กำหนด

```
while (!feof(fp)) {
```

```
fscanf(fp, "%d %s %S\n", &id, &fname, &lname);
```

```
printf("%d\t%s\t%s\n", id, fname, lname);
```

```
}
```

ผลลัพธ์ที่แสดงทางหน้าจอ

```
fclose(fp);
```

```
}
```

3

id

Bob

fname

Smith

lname





การเขียนข้อมูลลง Binary File

❖ รูปแบบ

```
fwrite(address ของตัวแปร, ขนาดตัวแปร, ขนาดที่ต้องการเขียน, ตัวแปร_File_Pointer);
```

❖ ตัวอย่าง

```
fwrite(&std, sizeof(Student), 1, fp);
```

ตัวอย่าง

```
typedef struct {  
    char id[12];  
    char name[50];  
    int age;  
    float grade;  
} Student;
```

```
Student getStudent() {  
    Student std;  
    printf("ID:"); scanf("%s", std.id);  
    printf("Name:"); scanf("%s", std.name);  
    printf("Age:"); scanf("%d", &std.age);  
    printf("Grade:"); scanf("%f", &std.grade);  
    return std;  
}
```

ฟังก์ชันรับค่าเก็บลง
ในตัวแปร structure

```
void main() {  
    FILE *fp = fopen("d:/data.dat", "wb");  
    Student std;  
    std = getStudent();  
    fwrite(&std, sizeof(Student), 1, fp);  
  
    std = getStudent();  
    fwrite(&std, sizeof(Student), 1, fp);  
    fclose(fp);  
}
```

เขียนข้อมูลในตัวแปร
structure ลงไฟล์



การอ่านข้อมูลจาก Binary File

❖ รูปแบบ

fread(Addressของตัวแปร, ขนาดของข้อมูล, จำนวนที่ต้องการอ่าน, ตัวแปร_File_Pointer);

❖ ตัวอย่าง

```
fread(&std, sizeof(Student), 1, fp);
```



ตัวอย่าง

```
typedef struct {  
    char id[12];  
    char name[50];  
    int age;  
    float grade;  
} Student;
```

```
void printCard(Student std) {  
    printf("%s %s %d\n", std.id, std.name, std.age);  
}
```

```
void main() {  
    FILE *fp = fopen("d:/data.dat", "rb");
```

```
    Student std;
```

```
    while(1) {  
        fread(&std, sizeof(Student), 1, fp);  
        if (feof(fp)) break;  
        printCard(std);  
    }
```

```
    fclose(fp);
```

```
}
```

← อ่านไฟล์ตามขนาดของตัว
แปร structure Student
และเก็บลงในตัวแปร std



กิจกรรม

สำนักงานแห่งหนึ่งเก็บจำนวนผู้รับบริการ ในแต่ละวันในไฟล์แบบ Text จงเขียนโปรแกรมอ่านข้อมูลจากไฟล์เพื่อนำมาแสดงผลทางหน้าจอ โดยใช้เครื่องหมาย * แทนผู้รับบริการ 1 คน

วันที่	ช่องว่าง	จำนวนผู้รับบริการ
30/12/15		6
31/12/15		4
01/01/16		12
02/01/16		11
03/01/16		7

ไฟล์ customer.txt

ผลลัพธ์ที่แสดงทางหน้าจอ





การเลื่อนตัวแปร File Pointer ไปจุดเริ่มต้น

❖ การเรียกฟังก์ชันเกี่ยวกับไฟล์ทั้งการอ่านและเขียน จะถูกเลื่อนตำแหน่งของ File Pointer ไปยังจุดที่อ่านหรือเขียนล่าสุดเสมอ ดังนั้นหากเขียนหรืออ่านไฟล์แล้ว และต้องการเลื่อน File Pointer กลับไปยังจุดเริ่มต้นจะต้องเรียกฟังก์ชันดังนี้

❖ รูปแบบ

```
rewind(ตัวแปร_File_Pointer);
```

❖ ตัวอย่าง

```
rewind(fp);
```



Assignment#11

ข้อ 1. จงเขียนโปรแกรมรับข้อมูลจาก
คีย์บอร์ด เพื่อบันทึกข้อมูลของเพื่อนใน
สาขาจำนวน 3 คน ประกอบด้วย ชื่อ
นามสกุล และชื่อเล่น หลังจากนั้นให้เปิด
ไฟล์เพื่อเขียนข้อมูลเหล่านี้ลงไฟล์ และ
เปิดไฟล์อีกครั้งเพื่ออ่านข้อมูลที่เขียนแล้ว
แสดงที่หน้าจอ

```
Enter First Name: Kotchaphan ↵  
Enter Last Name : Muangsan ↵  
Enter Nick Name : We ↵
```

```
Enter First Name: Chinapun ↵  
Enter Last Name : Janluksa ↵  
Enter Nick Name : Bom ↵
```

```
Enter First Name: Vachawat ↵  
Enter Last Name : Thongsombun ↵  
Enter Nick Name : Mark ↵
```

```
=====  
=====  
Kotchaphan Muangsan (We)  
Chinapun Janluksa (Bom)  
Vachawat Thongsombun (Mark)
```

ตัวอย่างการทำงาน



Assignment#11

ข้อ 2. จงเขียนโปรแกรมเพื่ออ่านไฟล์ที่มีข้อความดังกล่าว (เก็บในไฟล์ชื่อ input.txt) และทำการตรวจสอบว่าข้อความที่รับเข้ามามีสระที่เป็นตัว a e i o u อย่างละกี่ตัว โดยแสดงผลลัพธ์โดยการเขียนลงบนไฟล์ชื่อ output.txt

Input (input.txt)	Output (output.txt)
As his kingdom is being threatened by the Turks, young prince Vlad Tepes must become a monster feared by his own kingdom in order to obtain the power needed to protect his own family, and the families of his kingdom.	Character a : 9 Character e : 22 Character i : 15 Character o : 15 Character u : 3
When Tony Stark's world is torn apart by a formidable terrorist called the Mandarin, he starts an odyssey of rebuilding and retribution.	Character a : 11 Character e : 9 Character i : 8 Character o : 8 Character u : 2
The world's most famous monster is pitted against malevolent creatures who, bolstered by humanity's scientific arrogance, threaten our very existence.	Character a : 9 Character e : 17 Character i : 8 Character o : 9 Character u : 4