File Processing
(การประมวลผลแฟ้มข้อมูล)

ที่มา : ทัศนวรรณ ศูนย์กลาง :: http://www.cs.su.ac.th/~tasanawa/cs517111/files.ppt

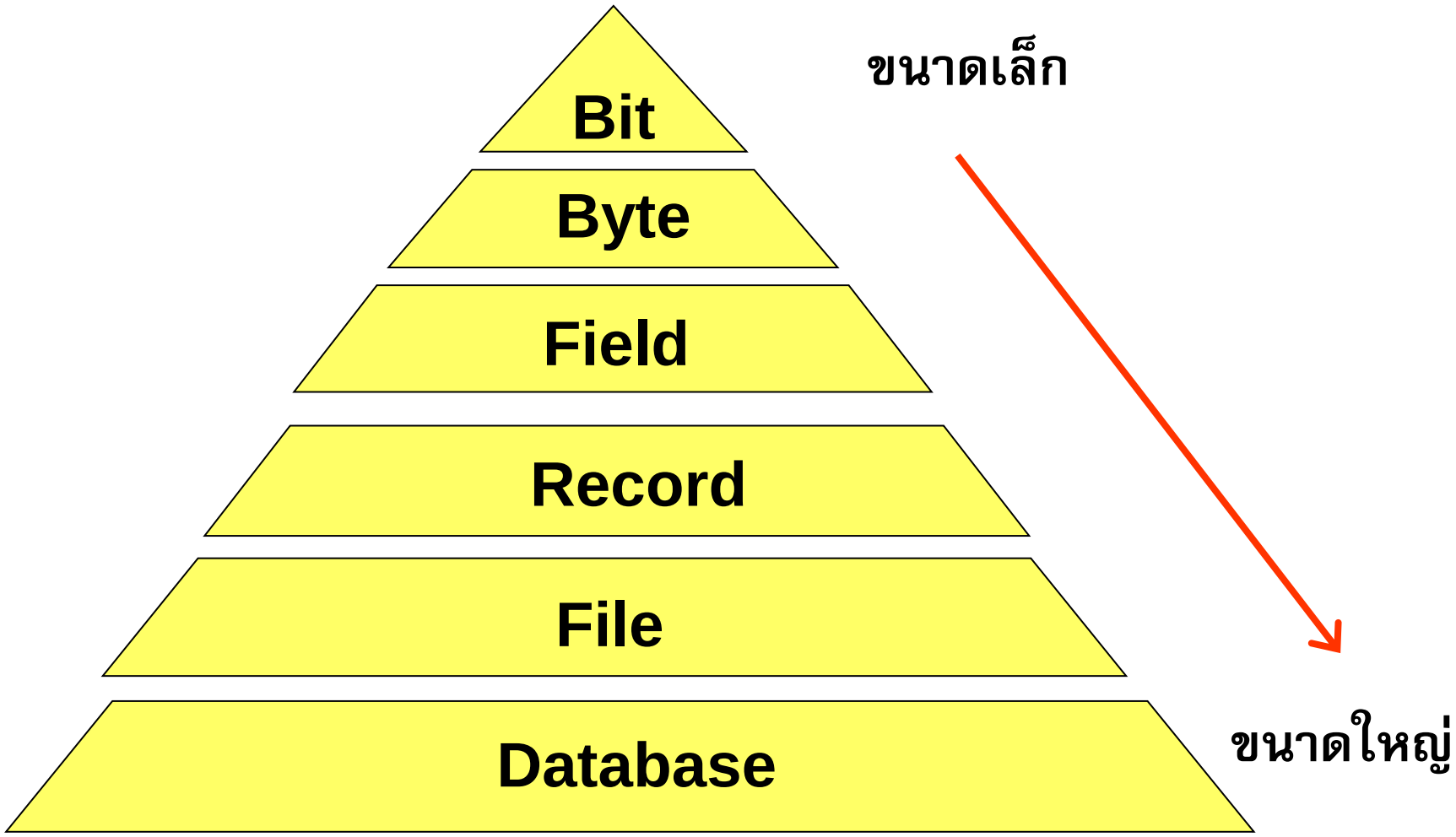
เนื้อหาที่จะเรียนรู้

- ทำไมจึงต้องมีการติดต่อแฟ้มข้อมูล
- ลำดับข้อมูล (Data Hierarchy)
- การเปิดและปิดแฟ้มข้อมูล
- การอ่านและเขียนแฟ้มข้อมูล
- ฟังก์ชันที่ใช้ประมวลผลแฟ้มข้อมูล

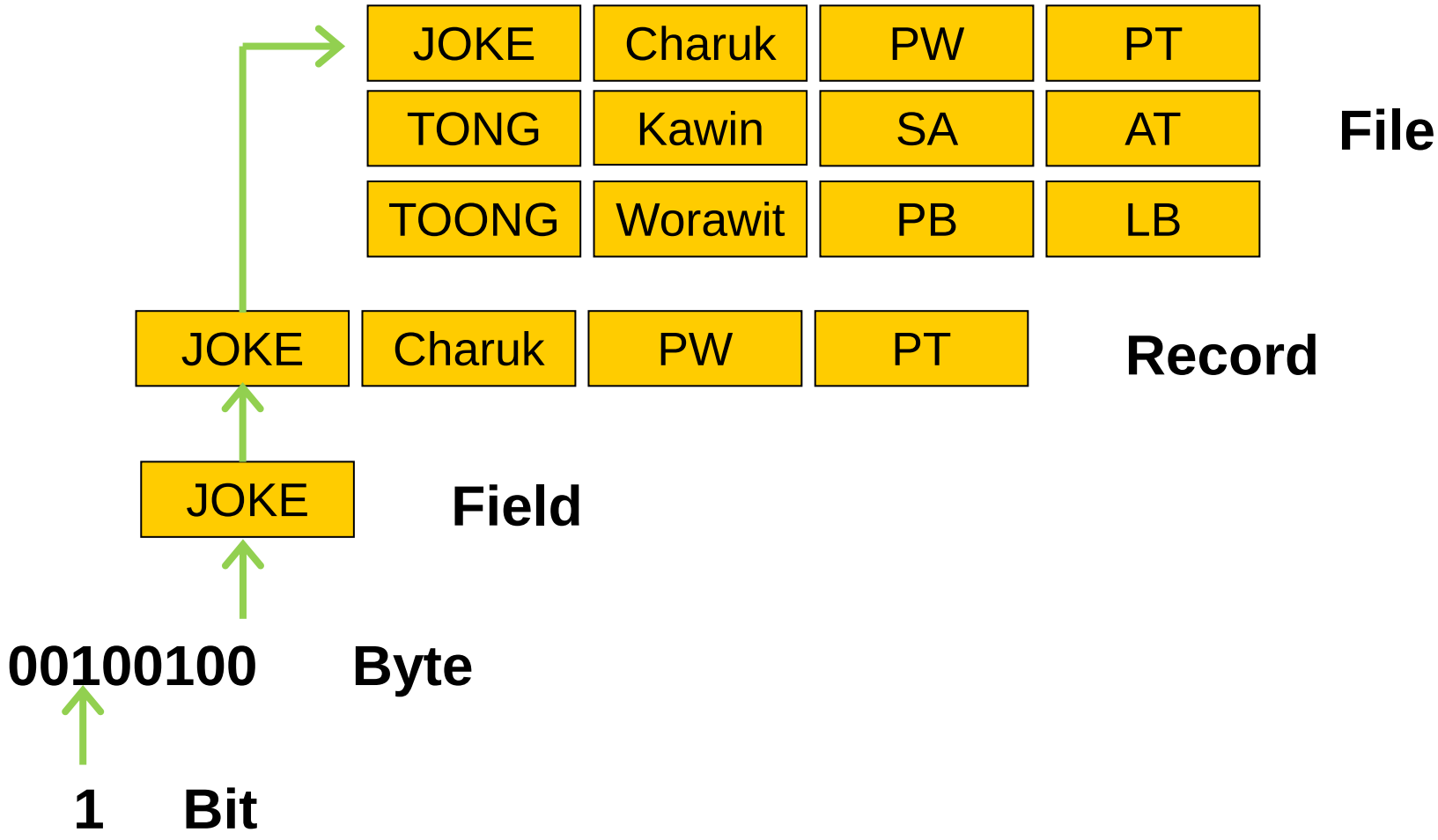
ทำไมจึงต้องมีการติดต่อแฟ้มข้อมูล

- ข้อมูลไม่หายเมื่อโปรแกรมจบการทำงาน
- เป็นการเก็บข้อมูลแบบถาวร อยู่ในหน่วยความจำรอง
- สามารถนำข้อมูลมาใช้ประโยชน์ได้หลังจากโปรแกรมจบการทำงาน
- ประมวลผลข้อมูลที่ถูกเก็บในแฟ้มข้อมูล โดยใช้โปรแกรมคอมพิวเตอร์

ลำดับของข้อมูล (data hierarchy)



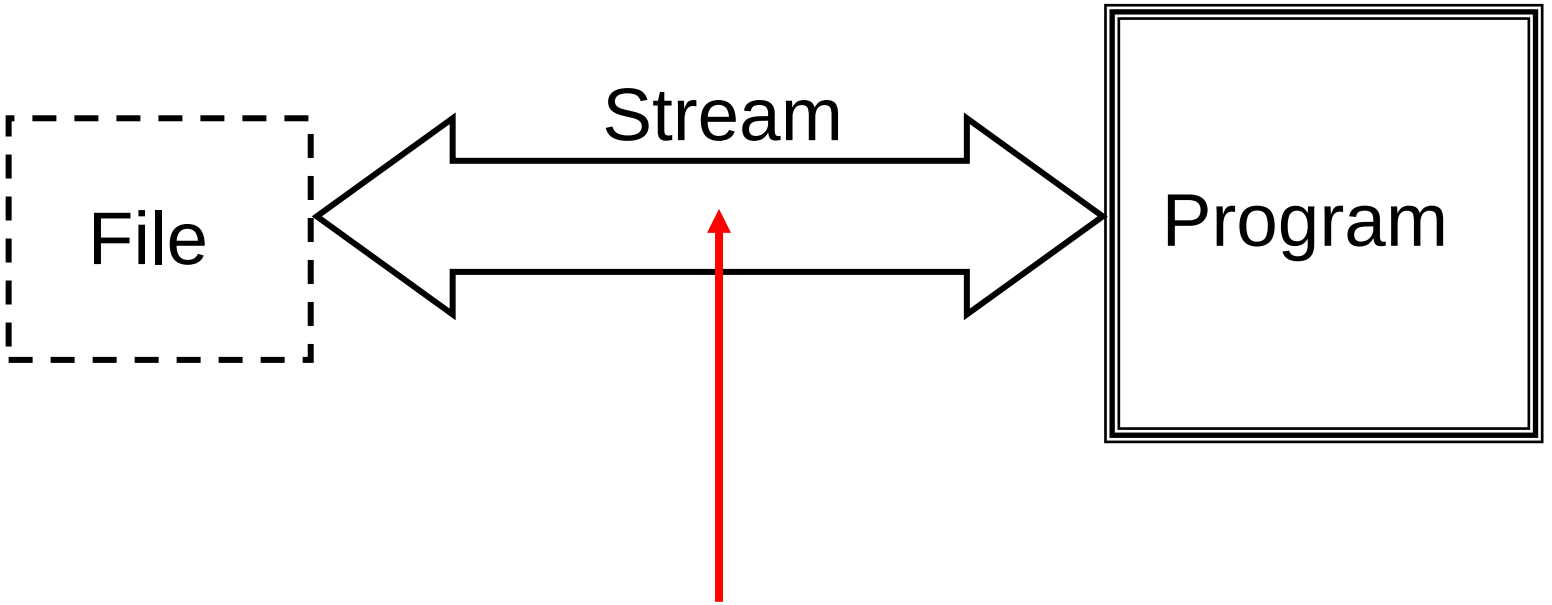
ลำดับของข้อมูล



การติดต่อแฟ้มข้อมูล

Stream เป็นหน่วยของข้อมูลที่เรียงติดกัน จบด้วยรหัส EOF

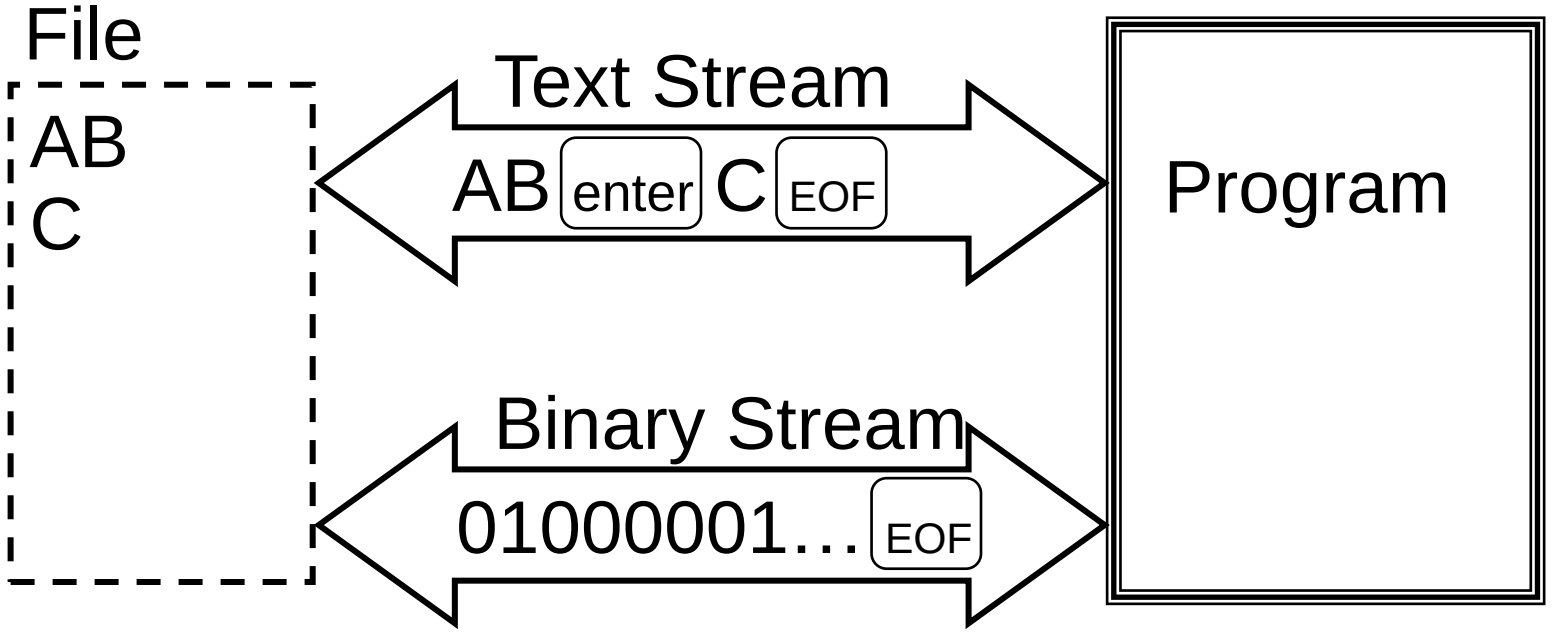
เป็นตัวเชื่อมต่อระหว่างแฟ้มข้อมูลกับโปรแกรม



ข้อมูลจะมีการไหลเข้า-ออกผ่าน Stream

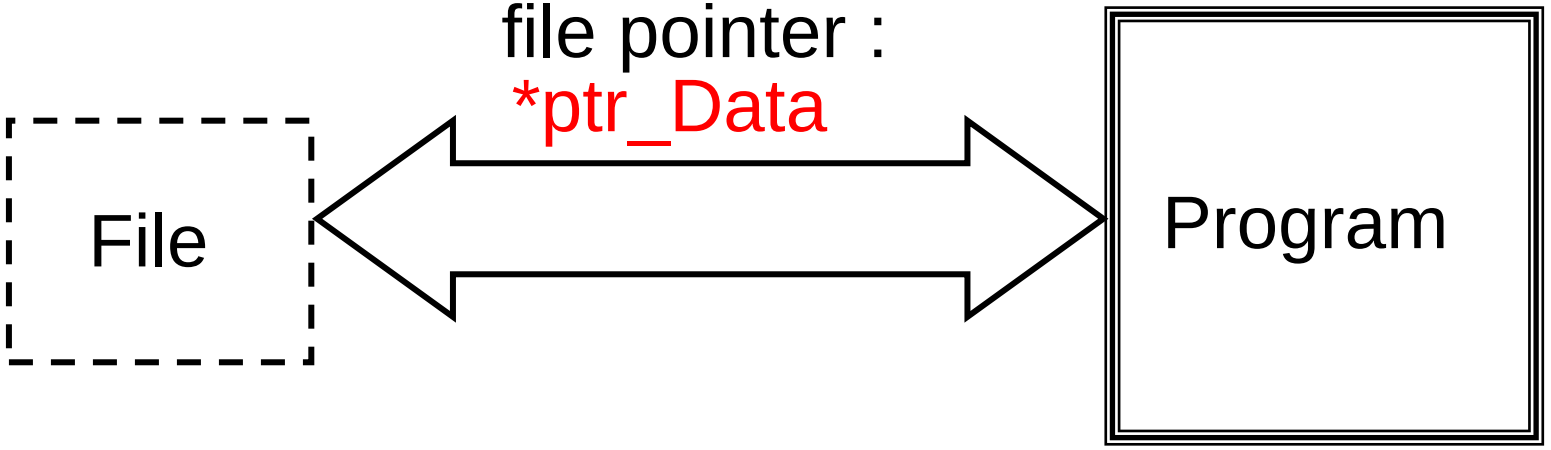
การติดต่อแฟ้มข้อมูล

- Text Stream เก็บข้อมูลในรูปแบบของรหัสแอสกี
- Binary Stream เก็บข้อมูลเป็นเลขฐานสองของข้อมูลนั้นๆ



การติดต่อแฟ้มข้อมูล

Stream จะถูกแทนด้วย ตัวแปรชี้แฟ้มข้อมูล (file pointer)
ที่กำหนดเป็นโครงสร้างข้อมูลชนิด **FILE ซึ่งถูกกำหนดไว้ใน stdio.h**



ตัวแปรชี้แฟ้มข้อมูล มีสัญลักษณ์ * นำหน้าชื่อตัวแปร เช่น *ptr_Data

กระบวนการกระทำกับไฟล์

1) เปิดไฟล์ : ติดต่อกับไฟล์ใด ติดต่อรูปแบบใด

2) กระทำกับไฟล์ : อ่านข้อมูลจากกไฟล์/
เขียนข้อมูลลงไฟล์

3) ปิดไฟล์ : ตัดการเชื่อมต่อกับไฟล์

ประกาศตัวแปรสำหรับตัวชี้แฟ้มข้อมูล

- แฟ้มข้อมูลที่ทำการศึกษานี้ เป็นชนิดแฟ้มข้อความ (text file)
- ตัวแปรที่เกี่ยวกับแฟ้มข้อมูลใช้โครงสร้างข้อมูลชนิด FILE

จะมีการใช้ตัวชี้แฟ้มข้อมูล เพื่ออ้างอิงถึงพื้นที่ดิสก์ที่เก็บแฟ้มข้อมูล

```
FILE  *ชื่อตัวแปรชี้แฟ้มข้อมูล;
```

ตัวอย่าง

```
FILE *ptrData;
```

การเปิดแฟ้มข้อมูล

ตัวชี้แฟ้ม = fopen(“ชื่อแฟ้มข้อมูล”,mode);

เงื่อนไขการคืนค่าของฟังก์ชัน fopen()

if **openfile complete**

return ***address***

if **openfile <u>not</u> complete !!**

return **NULL**

ตัวอย่าง ptrData = fopen(“c:\\student.txt”, “r”);

รูปแบบการเปิดแฟ้มข้อมูล

mode	ความหมาย
“r”	อ่านอย่างเดียว (read) ถ้าไม่มีแฟ้มอยู่จะเปิดไม่ได้
“w”	เปิดแฟ้มใหม่เขียนอย่างเดียว (write) ถ้าไม่มีแฟ้มอยู่จะสร้างแฟ้มใหม่
“a”	เขียนต่อท้าย (append) ถ้าไม่มีแฟ้มอยู่จะสร้างแฟ้มใหม่
“r+”	อ่านและเขียน (read/write) ถ้าไม่มีแฟ้มอยู่จะเปิดไม่ได้
“w+”	เปิดแฟ้มใหม่อ่านและเขียน ถ้าไม่มีแฟ้มอยู่จะสร้างแฟ้มใหม่
“a+”	อ่านและเขียนต่อท้าย (read/append) ถ้าไม่มีแฟ้มอยู่จะสร้างใหม่

การปิดแฟ้มข้อมูล

fclose(ตัวชี้แฟ้มข้อมูล);

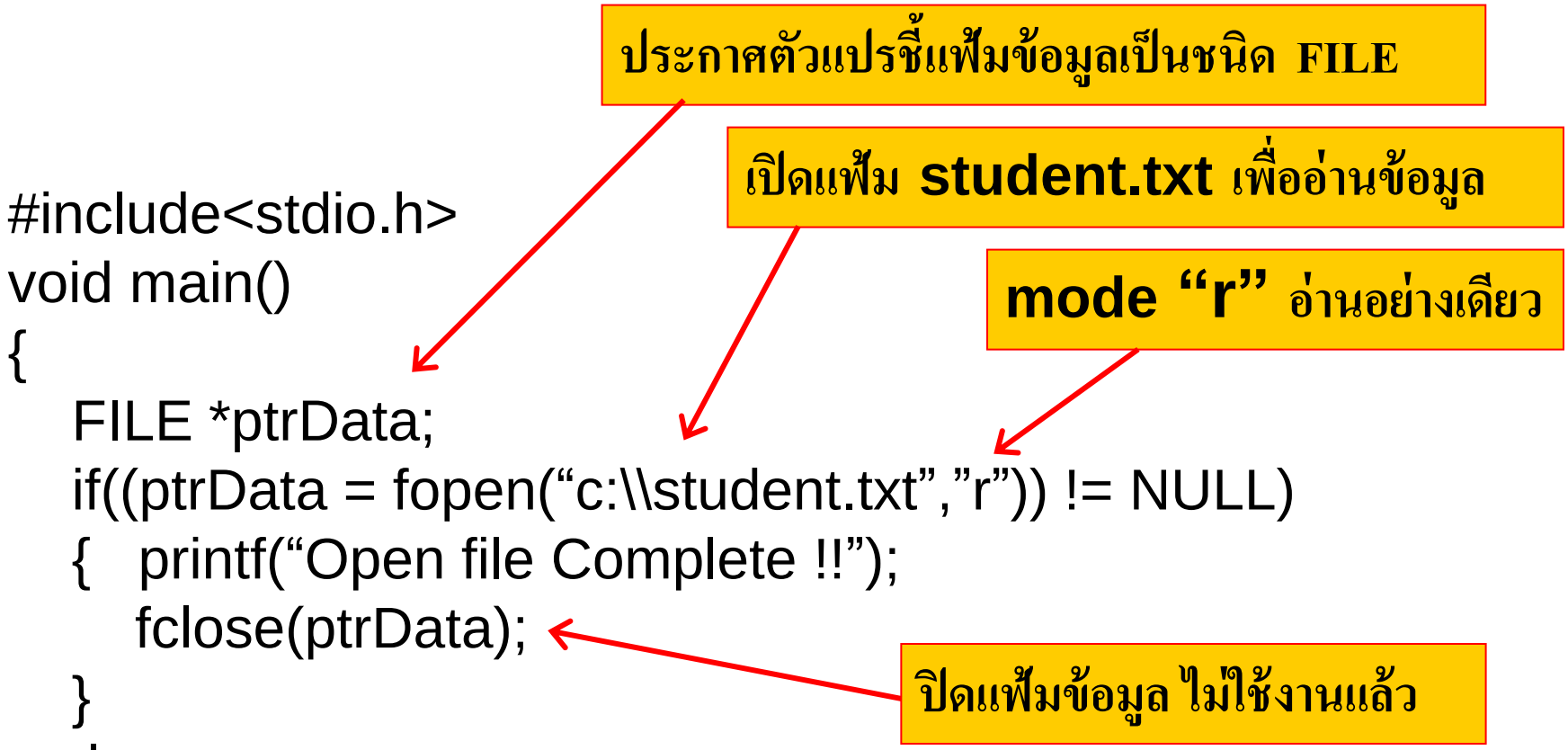
เงื่อนไขการคืนค่าของฟังก์ชัน fclose()

if **close file complete**

return **0**

fclose(ptrData);

ตัวอย่างโปรแกรมการเปิดและปิดแฟ้มข้อมูล




```
#include<stdio.h>
void main()
{
    FILE *ptrData;
    if((ptrData = fopen(“c:\\student.txt”,”r”)) != NULL)
    {   printf(“Open file Complete !!”);
        fclose(ptrData);
    }
   else
       printf(“Can not Open file”);
}
```

การอ่านและเขียนแฟ้มข้อมูลแบบ Sequential Access

เมื่อสามารถเปิดแฟ้มข้อมูลสำเร็จแล้ว สามารถนำข้อมูลในแฟ้มออกมาใช้งาน หรือเขียนข้อมูลไปยังแฟ้ม

ฟังก์ชันในการอ่าน/เขียนแฟ้มข้อมูล คล้ายกันกับฟังก์ชันที่เคยเรียนรู้

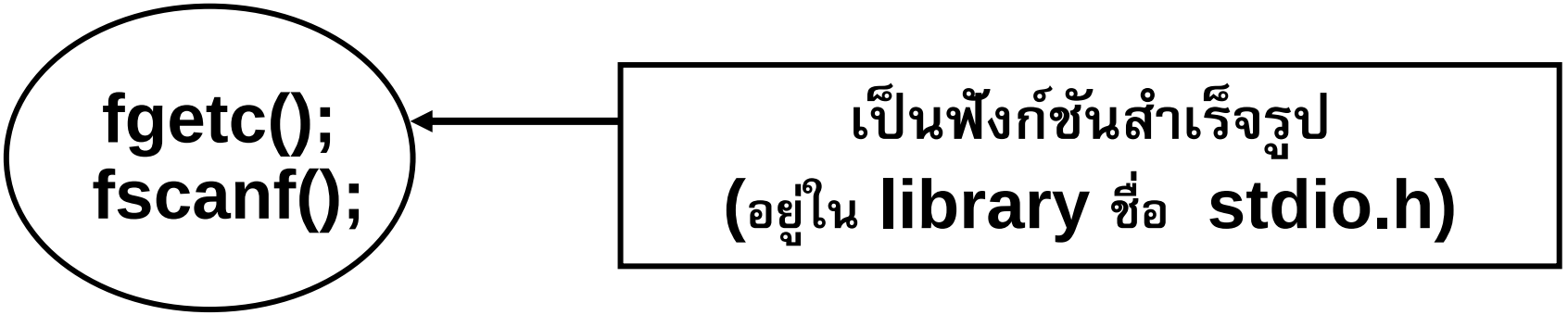
scanf( ); // รับค่าจากคีย์บอร์ด (Standard input)

printf( ); // ส่งค่าที่ต้องการแสดงผลออกหน้าจอ (Standard Output)

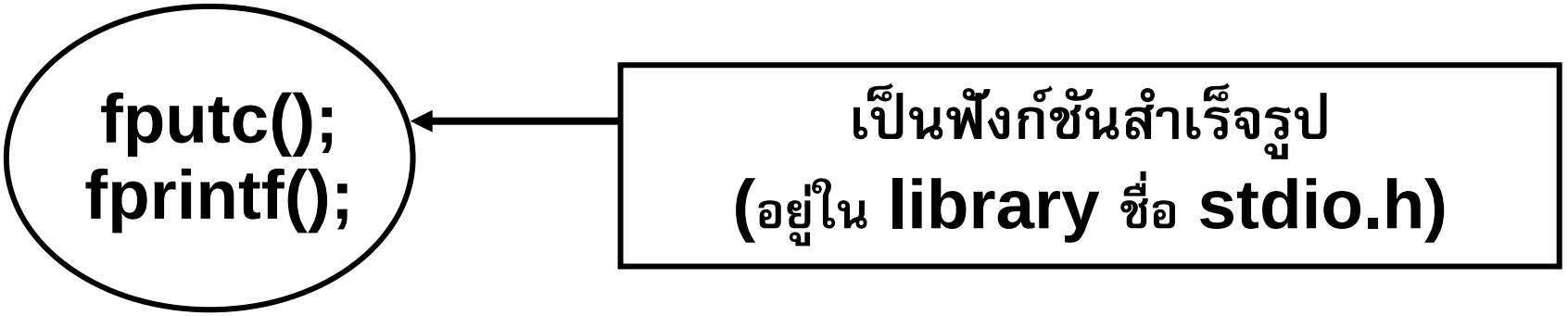
ฟังก์ชันสำหรับอ่าน/เขียนแฟ้มข้อมูลจะมี f นำหน้าชื่อฟังก์ชัน

ฟังก์ชันสำหรับอ่านและเขียนแฟ้มข้อมูล

- ฟังก์ชันที่ใช้ในการอ่านข้อมูลในแฟ้ม



- ฟังก์ชันที่ใช้ในการเขียนข้อมูลไปยังแฟ้ม



การเขียนข้อมูลในแฟ้มแบบ Sequential Access

fputc(**ตัวแปรชนิดอักขระ, ตัวชี้แฟ้ม**);

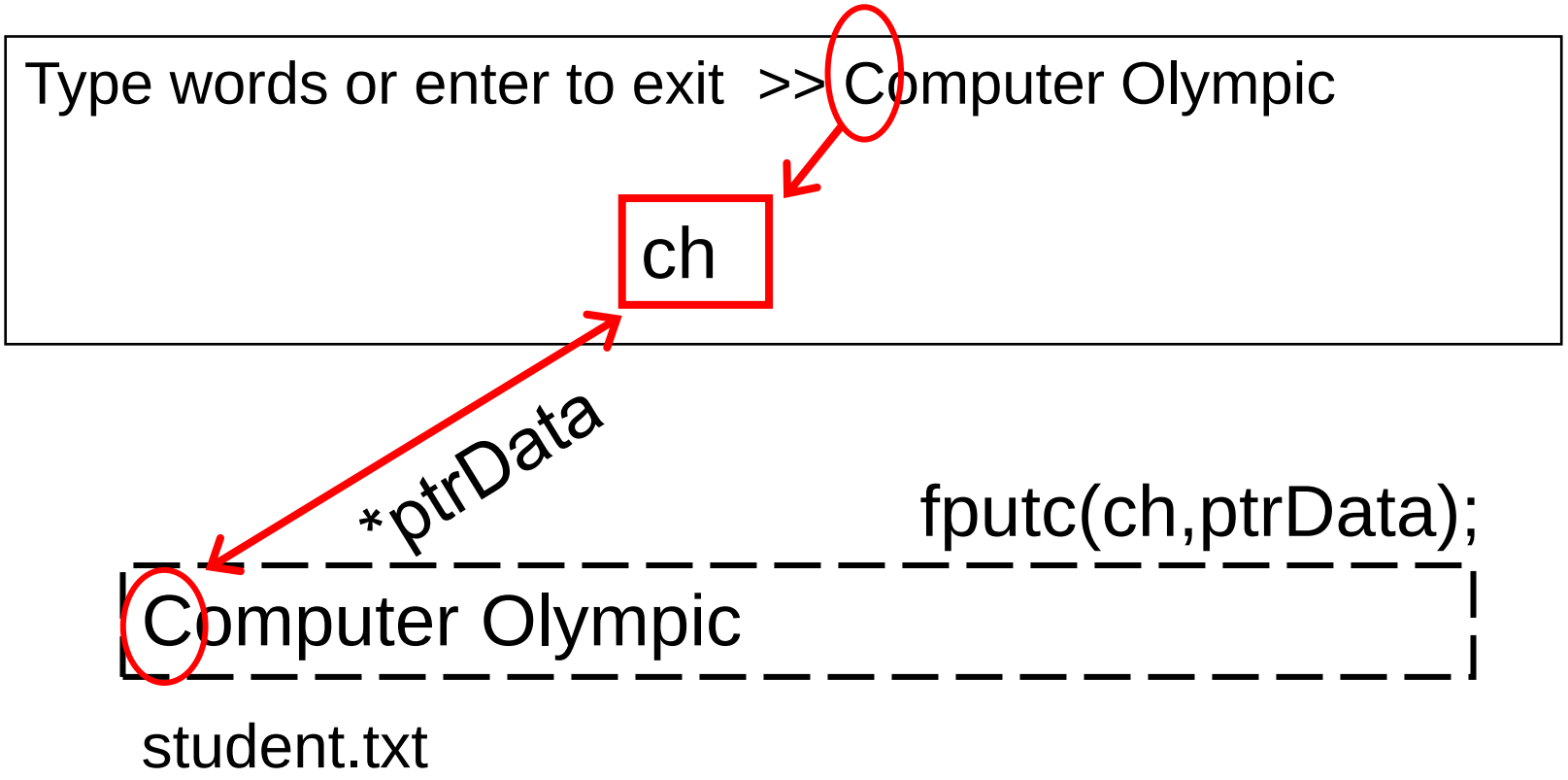
ทำงานคล้ายฟังก์ชัน putchar() เขียนข้อมูลทีละ 1 ตัว อักษรลงในแฟ้มข้อมูล

ตัวชี้แฟ้มข้อมูล

```
char output = 'Y';
fputc(output,ptrData);
```

ตัวแปรชนิดอักขระ

การเขียนข้อมูลในแฟ้มแบบ Sequential Access



```
#include <stdio.h>
void main() {
    FILE *ptrData;
    char ch;
    ptrData=fopen(“student. txt","w");
    if (ptrData != NULL) {
        printf(“Type words  or enter to exit >>");
        while(       (ch = getchar()          ) != '\n')
                fputc(ch,ptrData);
    }  else    printf(“Open file Error”);
    fclose(ptrData);
}
```

ประกาศตัวแปรชี้แฟ้มข้อมูลเป็นชนิด FILE

สร้างตัวแปร ch สำหรับเก็บตัวอักษร

เปิดแฟ้มข้อมูลเพื่อเขียนข้อมูล

รับข้อมูลมาเก็บทีละตัวอักษร

เขียนข้อมูลลงแฟ้มข้อมูลทีละตัวอักษร

ปิดแฟ้มข้อมูล

การอ่านข้อมูลในแฟ้มแบบ Sequential Access

ตัวแปรชนิดอักขระ = fgetc(ตัวชี้แฟ้ม);

ทำงานคล้ายฟังก์ชัน getchar() หรือ getch() อ่านข้อมูลจากแฟ้ม ทีละ 1 ตัวอักษร เมื่ออ่านไปถึงจุดจบแฟ้มแล้วจะคืนค่า EOF

```
char input;
input = fgetc(ptrData);
```

ตัวแปรชนิดอักขระ

ตัวชี้แฟ้มข้อมูล

```
#include <stdio.h>
void main() {
        FILE *ptrData;
        char ch;
        ptrData=fopen(“student.txt”,”r”);
        if (ptrData != NULL) {
            while (     (ch=fgetc(ptrData)        ) != EOF)
                    printf(“%c”,ch);
        } else    printf(“Open file Error”);
        fclose(ptrData);
}
```

การเขียนข้อมูลในแฟ้มแบบ Sequential Access

fprintf(ตัวชี้แฟ้ม, “รูปแบบข้อมูล”, ตัวแปร);

ทำงานคล้ายฟังก์ชัน printf() เขียนข้อมูลลงแฟ้มโดยจัดรูปแบบของข้อความและข้อมูลให้อยู่ในชนิดที่ต้องการได้

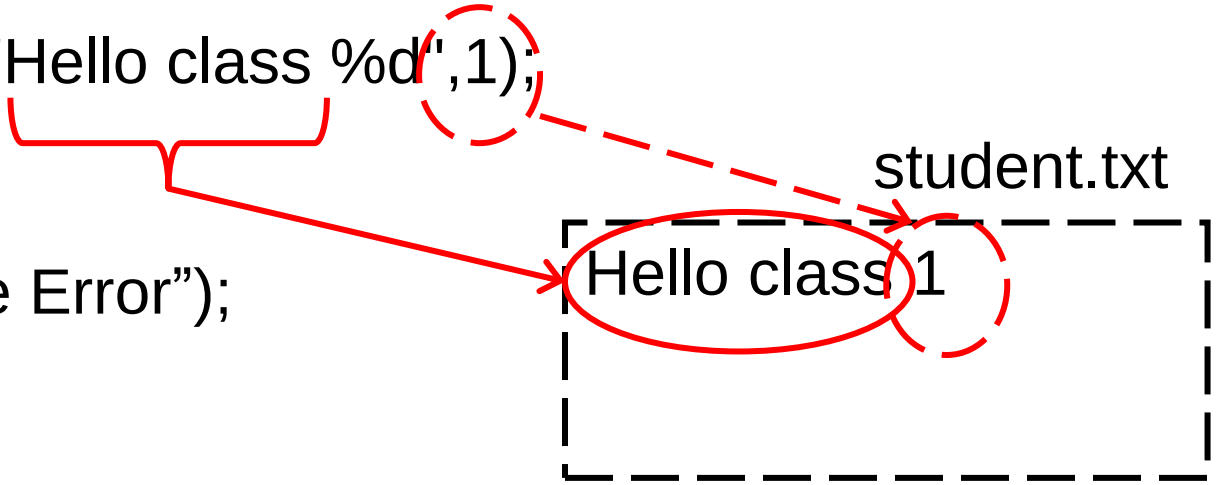
fprintf(ptrData,“N’ MaM is %d years old”,23);

ตัวชี้แฟ้ม

รูปแบบข้อความและข้อมูล

ข้อมูล

```
#include <stdio.h>
void main()
{
    FILE *ptrData;
    ptrData=fopen(“student.txt","w");
    if (ptrData != NULL)
        fprintf(ptrData,"Hello class %d",1);
    else
        printf(“Open file Error”);
    fclose(ptrData);
}
```



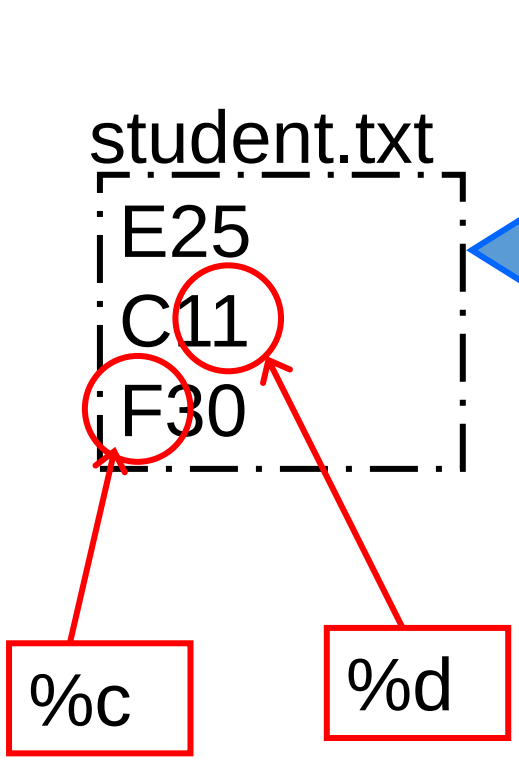
การอ่านข้อมูลในแฟ้มแบบ Sequential Access

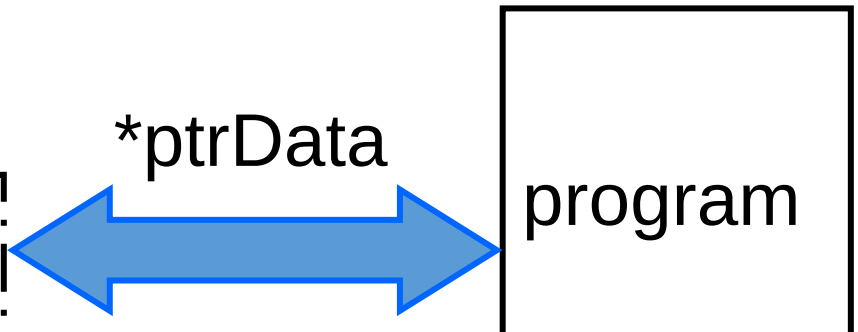
fscanf(ตัวชี้แฟ้ม, “รูปแบบข้อมูล”, &ตัวแปร);

ทำงานคล้ายฟังก์ชัน scanf() อ่านข้อมูลจากแฟ้มทีละ 1 ค่า โดยจัดรูปแบบข้อมูลให้อยู่ในชนิดที่ต้องการได้

fscanf(ptrData,“%c%d%d”,&ch,&num1,&num2);

การอ่านข้อมูลในแฟ้มแบบ Sequential Access



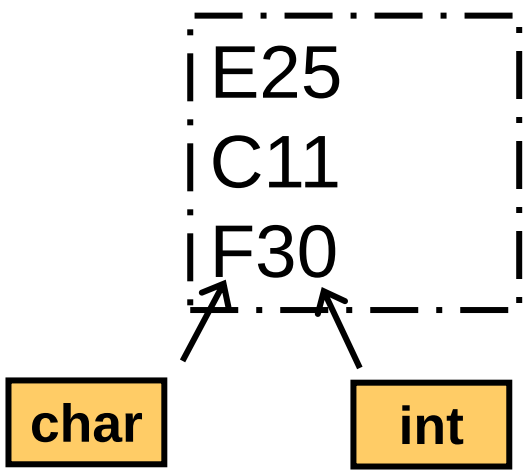


ผลลัพธ์

```
your room is C11
```

```
fscanf(ptrData,"%c%d",&ch,&num);
```

```
#include <stdio.h>
void main() {
    FILE *ptrData;
    char ch;   int num;
    ptrData=fopen(“student.txt",“r");
    if (ptrData != NULL)  {
        while(feof(ptrData) ==0)  {
            fscanf(ptrData,"%c%d",&ch,&num);
            if (ch==‘C')
                printf("your room is %c%d",ch,num);
        }
    }    else    printf(“Open file Error”);
    fclose(ptrData);
}
```



การตรวจสอบการอ่าน/เขียน แฟ้มข้อมูล

ป้องกันไม่ให้โปรแกรมทำงานผิดพลาด !!!

ฟังก์ชันสำหรับการตรวจสอบก่อนการอ่าน/เขียน แฟ้มข้อมูล

ferorr();ตรวจสอบสถานะความผิดพลาดของการประมวลผลแฟ้ม

feof(); ตรวจสอบจุดสิ้นสุดของแฟ้มข้อมูล

อาจจะใช้ตัวแปรเฉพาะ EOF (End Of File) นำมาตรวจสอบจุดสิ้นสุดของแฟ้มข้อมูล

การตรวจสอบการอ่าน/เขียน แฟ้มข้อมูล

retvalue = ferror(ตัวชี้แฟ้ม);

retvalue คือ ตัวแปรที่รับค่าที่ส่งกลับมาจากฟังก์ชัน ferror

ค่าตัวแปรที่ส่งกลับมา มีความหมาย 2 แบบ ดังนี้

1. ถ้า rtvalue = 0 คือ ไม่มีข้อผิดพลาด
2. ถ้า rtvalue > 0 คือ มีข้อผิดพลาดเกิดขึ้น

การตรวจสอบการอ่าน/เขียน แฟ้มข้อมูล

retvalue = feof(ตัวชี้แฟ้ม);

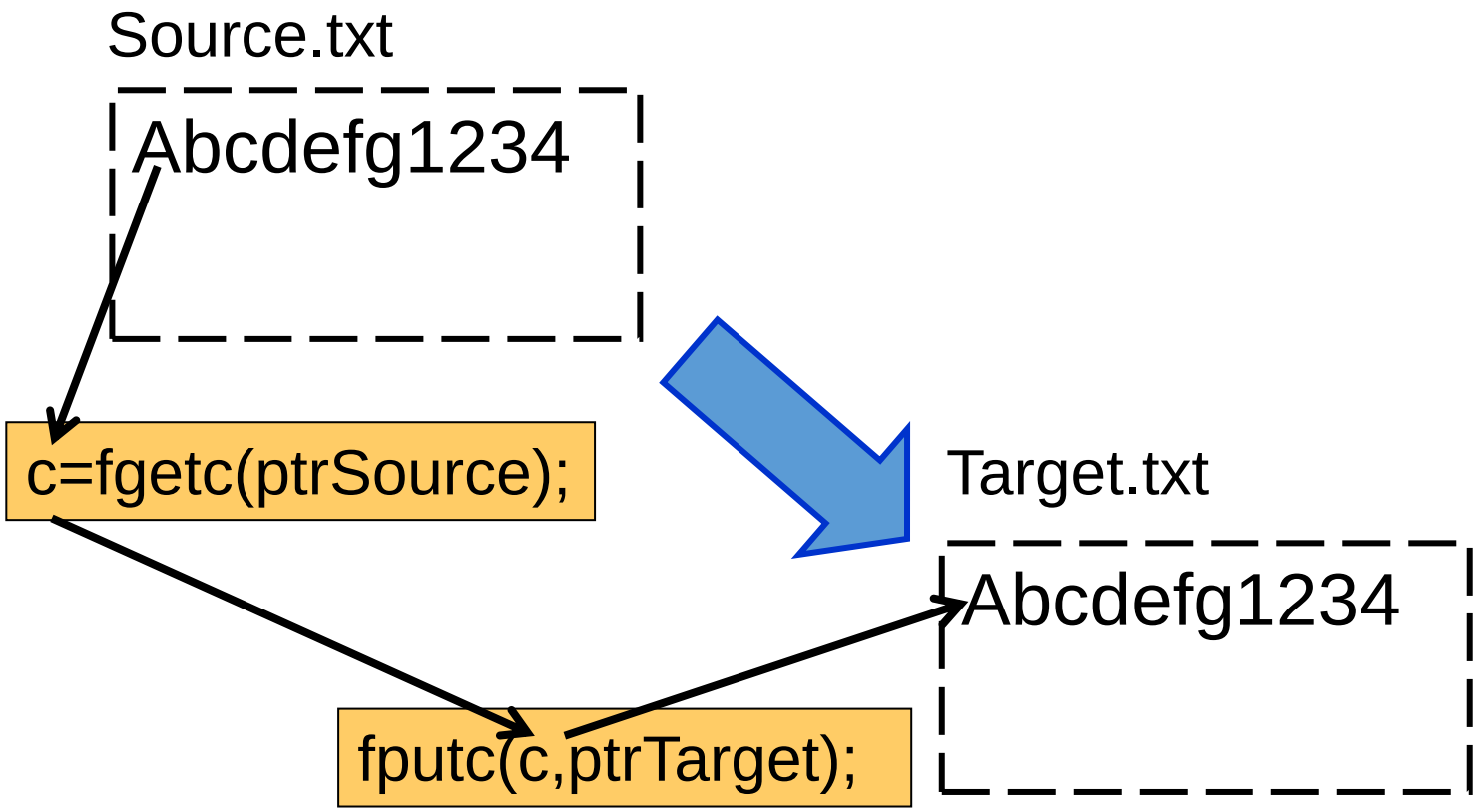
retvalue คือ ตัวแปรที่รับค่าที่ส่งกลับมาจากฟังก์ชัน feof

ค่าตัวแปรที่ส่งกลับมา มีความหมาย 2 แบบ ดังนี้

1. ถ้า rtvalue = 0 คือ ตัวชี้ยังไม่ถึงจุดสิ้นสุดของแฟ้ม (not EOF)
2. ถ้า rtvalue > 0 คือ ตัวชี้ถึงจุดสิ้นสุดของแฟ้ม (EOF)

** rtvalue > 0 เป็นตัวเลขอื่นๆ ได้ ต้องระวังในการตรวจสอบ

การตรวจสอบการอ่าน/เขียน แฟ้มข้อมูล



```
#include <stdio.h>
void main() {
    FILE *ptrSource;
    FILE *ptrTarget;
    char c;
    if(    (ptrSource=fopen("Source.txt","r")     ) != NULL)
    {
        if(    (ptrTarget=fopen("Target.txt","w")     ) != NULL)
        {
          while(!    feof(ptrSource)      ) /* check EOF */
          {
              c=fgetc(ptrSource);
              if(   ferror(ptrSource)    != 0) /* check write data */
              {
                  printf("Error Read data from source file\n");
                  return();
              }
```

```
            fputc(c,ptrTarget); /* write to target file */

             if(     ferror(ptrTarget)      != 0) /* check write data */
                  { printf("Error Writing to Target file\n");
                    return();
                  }
            } /* end while */
          }  /* end if open Target file */
          else
            printf("Can't open Target file");
        }  /* end if open Source file */
        else
            printf("Can't open Source file");

        fclose(ptrSource);
        fclose(ptrTarget);
}  /* end main */
```

แบบฝึกหัด

- กำหนดให้ ในแฟ้มข้อมูล **num.data** มีข้อมูลดังภาพ

1	4	2	8	14
15	40	57	32	84
91	57	5	7	100

จงเขียนโปรแกรมเพื่อหาผลรวม ของข้อมูลจากแฟ้ม **num.txt** แล้วแสดงผลออกหน้าจอและบันทึกผลลัพธ์ที่แฟ้ม **num_out.txt**

แบบฝึกหัด

กำหนดให้แฟ้มข้อมูล friends.data เก็บข้อมูลชื่อและอายุ ดังภาพ

Somchai	15
Devil	14
Satan	20
Rungtip	16

จงเขียนโปรแกรมค้นหา ชื่อของเพื่อนที่ต้องการถ้าพบข้อมูลตรงกับที่ต้องการค้นหาให้แสดงชื่อและอายุของเพื่อนทางจอภาพ

อ่านและเขียนข้อมูลแบบ Sequential Access

ที่ผ่านมา เป็นการอ่าน/เขียนข้อมูล ทีละ 1 ค่าหรือทีละ 1 ตัวอักษร

การอ่าน/เขียนข้อมูล แบบเป็นสายข้อมูล (String)

- จะใช้ตัวแปรแบบสายข้อมูล เรียกว่า String ในการนำมารับค่าข้อมูลที่อ่านจากแฟ้มข้อมูล หรือเขียนลงแฟ้มข้อมูล

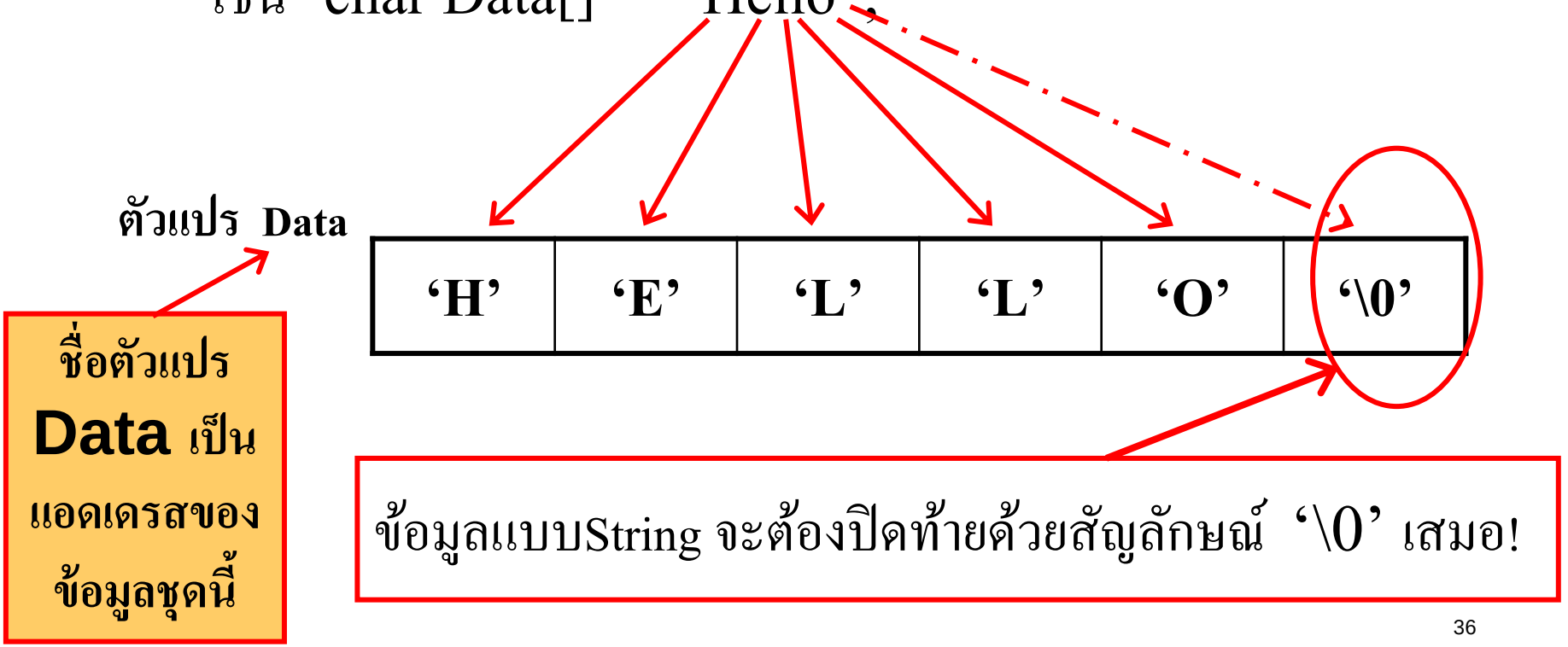
String คือ ชุดของตัวแปรแบบตัวอักษร (Array of Character) ที่มีข้อมูลที่เป็นตัวอักษรเรียงติดต่อกันมากกว่า 1 ตัวอักษร

เช่น “Hello” , “Computer Olympic”

การอ่าน/เขียนข้อมูล แบบ String (ต่อ)

การใช้งานตัวแปรแบบ String

เช่น char Data[] = “Hello”;



การอ่าน/เขียนข้อมูล แบบ String (ต่อ)

ฟังก์ชันในการอ่าน/เขียนแฟ้มข้อมูลแบบ string

- fgets();
- fputs();

การเขียนข้อมูล แบบ String

- การเขียนข้อมูลแบบ string โดยใช้ฟังก์ชัน fputs();
  - เขียนข้อมูลจากตัวแปร string ลงแฟ้มข้อมูล

fputs(แอดเดรสของตัวแปรที่รับข้อมูล, ตัวชี้แฟ้ม);

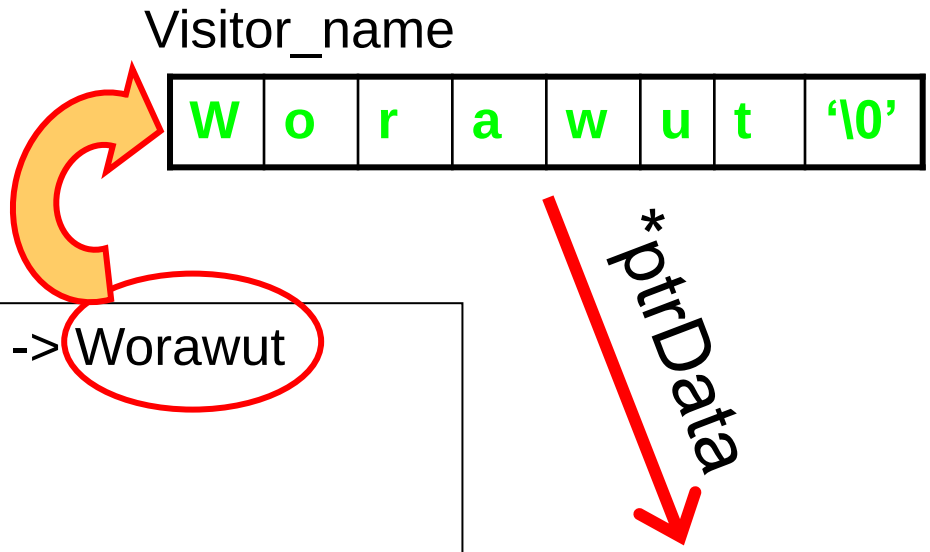
เช่น

fputs(str_out,ptrData);

ตัวแปรแบบ string มีข้อมูล นำไปเขียนลงแฟ้ม

ตัวชี้แฟ้ม

การเขียนข้อมูล แบบ String (ต่อ)



หน้าจอ

Type Visitor name and press “enter” -> Worawut

Press ‘y’ to enter another name or any key to exit-> n

```
fputs(visitor_name,ptrData);
```

```
#include <stdio.h>
#include <string.h>
void main()
{  FILE *ptrData;
  char visitor_name[50];
if(    (ptrData=fopen("visitor.txt",“w“)        ) != NULL)
       {  printf(“Type Visitor name and press “enter” ->”);
          scanf(“%s”,visitor_name);
          fputs(visitor_name,ptrData);}
       } else       printf("\nOpenfile not Complete\n");
   fclose(ptrData);
}
```

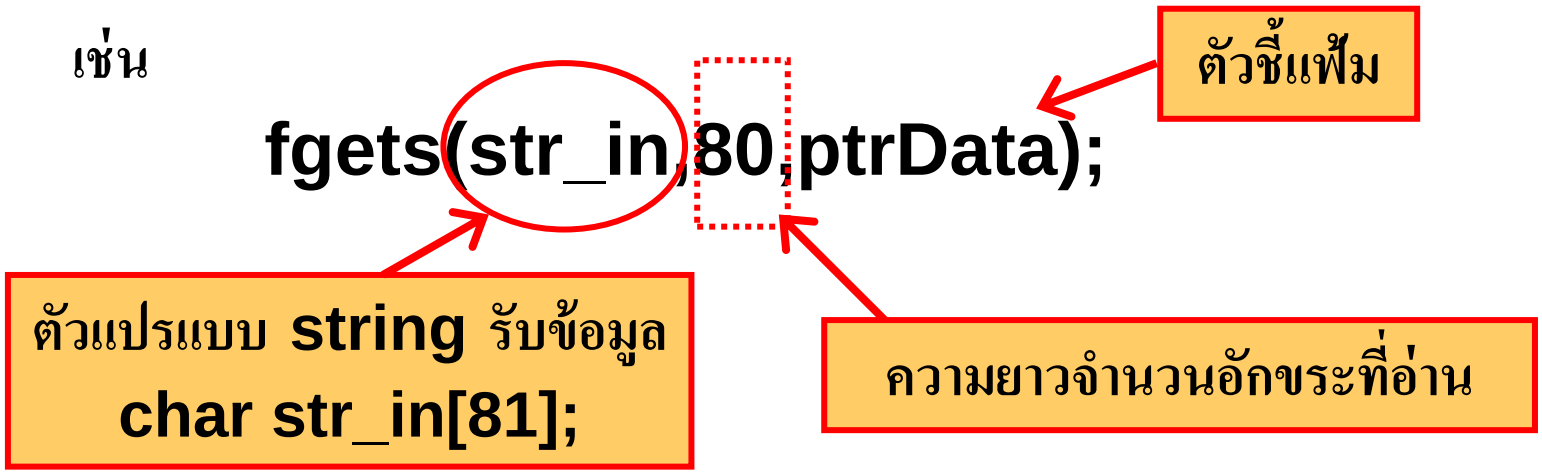
การอ่านข้อมูล แบบ String

- การอ่านข้อมูลแบบ string โดยใช้ฟังก์ชัน fgets();
  - อ่านข้อมูลจากแฟ้มเป็นสายข้อมูลตามความยาวที่ระบุ
  - มีสัญลักษณ์ '\0' ปิดท้ายสายข้อมูลที่อ่านมาด้วยเสมอ

fgets(แอดเดรสของตัวแปรที่รับข้อมูล, จำนวนอักขระ, ตัวชี้แฟ้ม);

เช่น

```
fgets(str_in,80,ptrData);
```



การอ่านข้อมูล แบบ String (ต่อ)

student.txt

25 อักขระ

```
abcdefghijklmnop123456789

opqrstuvwxyz102004507872
```

```
fgets(str_in,25,ptrData);
```

str_in

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	1	2	3	4	5	6	7	8	'\0'
0																								24

** ต้องกำหนดขนาดของ **string** ให้สัมพันธ์กับความยาวอักขระที่รับมา !!! อย่าลืมนับ **'\0'** ด้วย **

```
#include <stdio.h>
#include <string.h>
void main() {
  FILE *ptrData;
  char str_in[25];
  if(      (ptrData=fopen("student.txt","r")     ) != NULL)    {
       fgets(str_in,25,ptrData);
       printf("\n%s",str_in);
   }
    else       printf("\nOpenfile not Complete\n");
   fclose(ptrData);
}
```

ฟังก์ชันที่ใช้เกี่ยวกับการประมวลผลแฟ้มข้อมูล

- ฟังก์ชัน **remove( )**

  - ใช้สำหรับการลบแฟ้มที่ระบุออกจากพื้นที่ดิกส์

syntax: **remove(ชื่อแฟ้ม);**

เช่น ต้องการลบแฟ้ม **student.txt**

remove(student.txt);

ฟังก์ชันที่ใช้เกี่ยวกับการประมวลผลแฟ้มข้อมูล

- ฟังก์ชัน **rewind( )**

  - ใช้สำหรับการกำหนดให้ตัวชี้แฟ้มข้อมูลกลับไปชี้ยังต้นแฟ้ม

***syntax*:** **rewind(ตัวชี้แฟ้ม);**

เช่น

```
rewind(ptrData);
```

ฟังก์ชันที่ใช้เกี่ยวกับการประมวลผลแฟ้มข้อมูล

- ฟังก์ชัน **fflush( )**

  - ใช้สำหรับการชำระค่าให้ตัวชี้แฟ้มข้อมูลไม่ให้มีข้อมูลใดอยู่ภายใน

syntax: **fflush(ตัวชี้แฟ้ม);**

เช่น

fflush(ptrData);

การเปิดและปิดแฟ้มข้อมูล (ชนิดBinary)

- **mode** ของการเปิดแฟ้มข้อมูล สังเกตุจะมี **b** ต่อท้ายโหมดที่รู้จักแล้ว

mode	ความหมาย
“rb”	อ่านอย่างเดียว (read) ถ้าไม่มีแฟ้มอยู่จะเปิดไม่ได้
“wb”	เปิดแฟ้มใหม่เขียนอย่างเดียว (write) ถ้าไม่มีแฟ้มอยู่จะสร้างแฟ้มใหม่
“ab”	เขียนต่อท้าย (update) ถ้าไม่มีแฟ้มอยู่จะสร้างแฟ้มใหม่
“rb+”	อ่านและเขียน (read/write) ถ้าไม่มีแฟ้มอยู่จะเปิดไม่ได้
“wb+”	เปิดแฟ้มใหม่อ่านและเขียน ถ้าไม่มีแฟ้มอยู่จะสร้างแฟ้มใหม่
“ab+”	อ่านและเขียนต่อท้าย (read/update) ถ้าไม่มีแฟ้มอยู่จะสร้างใหม่

การใช้งานแฟ้มข้อมูลชนิด binary

- สามารถนำคำสั่งที่เกี่ยวกับการทำงาน แฟ้มข้อมูลชนิด text มาจัดการได้ fgetc(), fscanf(), fgets(), fputc(), fprintf(), fputs()