Programming Fundamental

Struct File

Struct

การสร้าง data type ใหม่ โดยประกอบจาก data type ที่มีอยู่เดิม

```
จากการกำหนดตัวแปรสำหรับการแสดงผลกระสุนของยาน
int bullet_x;
int bullet_y;
int bullet_status;
```

การกำหนดข้อมูลสำหรับการแสดงผลกระสุน 1 นัดจำเป็นต้องใช้ตัวแปร 3 ตัว ซึ่ง
ทั้ง 3 ตัวถือว่าเป็นองค์ประกอบของข้อมูล "กระสุน"
การแยกตัวแปรออกเป็นส่วนๆ ทำให้เขียนโปรแกรมได้ยาก ถ้าต้องการยิงกระสุน
30 นัด ต้องกำหนดตัวแปร ก็ต้องมีตัวแปรทั้งหมด 90 ตัวแปร
และต้องเขียน Code ในการทำงานของกระสุนเป็นรายนัด

จากกรณีของการเขียนโปรแกรมควบคุมยานหรือยิงกระสุน ยาน 1 ลำ หรือกระสุน 1 ลูก จะมีรายละเอียดข้อมูลที่เก็บมากกว่า 1 ค่า เช่น พิกัด (x,y), ลักษณะยาน, จำนวนกระสุนที่ยิงได้ และพิกัด (x,y) ของกระสุนแต่ละลูก ซึ่งปัญหา ที่พบคือ ภาษา C ไม่มี Data Type ใดที่สามารถเก็บข้อมูลในลักษณะนี้ได้ ภาษา C จึงให้ programmer สามารถสร้าง data type ใหม่ จาก data type ที่มีอยู่เดิม ได้ .. โดยใช้คำสั่ง struct

การเก็บข้อมูลของสิ่งใดสิ่งหนึ่งที่มีรายละเอียดมากกว่า ตัวเลข/ตัวอักษร 1 ตัว จึง ควรมีการสร้าง data type สำหรับเก็บข้อมูลของสิ่งนั้นๆ โดยเฉพาะ

ควรใช้ตัวแปรเพียง 1 ตัว เก็บข้อมูลที่เกี่ยวข้องกับกระสุน 1 นัด

Struct = การสร้าง Data Type ชุดใหม่ จากการเอา Data Type ที่มีอยู่มา ประกอบกับ

int bullet_x;

int bullet y;

int bullet_status

int x;
int y;
int status;

ขั้นตอนการใช้งาน struct

1. สร้าง data type ใหม่ โดยใช้คำสั่ง struct

- 2. สร้างตัวแปร ตาม data type ที่สร้างขึ้น
- 3. ใช้งานตัวแปรย่อยต่างๆ โดยมองว่า [ตัวแปร].[Attribute] คือตัวแปรตัว หนึ่งตามชนิดที่ประกาศ

```
struct bullet {
    int x;
    int y;
    int status;
};
```

```
bullet b;
```

```
b.x = 30;
b.y = 5;
b.status = 0;
```

ตัวอย่างโปรแกรม เพื่อเก็บชื่อผู้เล่น (name) ด่านที่เล่น (Level) และ คะแนน (Score) โดยชื่อมีความยาวไม่ เกิน 50 ตัวอักษร Level และ Score เป็นตัวเลข โดยมีการสร้าง Struct เพื่อ สร้าง data type เฉพาะสำหรับเก็บค่า ต่างๆ

```
#define CRT SECURE NO WARNINGS 1
#include<string.h>
#include<stdio.h>
int main()
         struct player {
                   char name[50];
                   int level;
                   int score;
          player p;
          strcpy(p.name, "Anonymous");
          p.level = 1;
          p.score = 100;
          printf("name : %s\n", p.name);
          printf("level : %d\n", p.level);
          printf("score : %d\n", p.score);
          return 0;
```

data type ที่สร้างขึ้นโดยคำสั่ง struct จะสามารถใช้งานได้เหมือนกับ int , float , char .. จึงสามารถใช้งานร่วมกับ Array และ Pointer ได้เช่นกัน

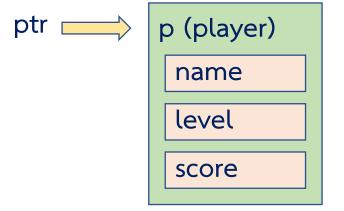
Array of structure

```
#define CRT SECURE NO WARNINGS 1
#include<string.h>
#include<stdio.h>
int main()
  int i;
  struct player {
    char name[50];
    int level;
    int score;
  };
  player p[10];
  for (i = 0; i < 10; i++){
    strcpy(p[i].name, "Anonymous");
    p[i].level = i;
    p[i].score = i*100;
```

```
for (i = 0; i < 10; i++){
    printf("name : %s\t", p[i].name);
    printf("level : %d\t", p[i].level);
    printf("score : %d\n", p[i].score);
}
return 0;
}</pre>
```

```
name : Anonymous
                          level
                                           score:
       Anonymous
                          level
                                           score
       Anonymous
                          level
                                                   200
                                           score
       Anonymous
                          level
                                                   300
                                           score
       Anonymous
                          level
                                           score
       Anonymous
                          level
                                                   500
                                           score
       Anonymous
                                                   600
                          level
                                           score
       Anonymous
                                                   700
                          level
                                           score
name :
      Anonymous
                         level
                                                   800
                                           score
                         level:9
name:
       Anonymous
                                                   900
                                           score
```

Pointer to structure



```
p.name = ptr->name
p.level = ptr->level
p.score = ptr->score
```

```
#define _CRT_SECURE_NO_WARNINGS 1
#include<string.h>
#include<stdio.h>
int main()
{
          int i;
          struct player {
                    char name[50];
                    int level;
                    int score;
          };
          player p, * ptr;
          ptr = &p;
          strcpy(p.name, "Anonymous");
          p.level = 1;
          p.score = 100;
          printf("name : %s\n", ptr->name);
          printf("level : %d\n", ptr->level);
          printf("score : %d\n", ptr->score);
          return 0;
```

Pointer to Array of structure

```
#define CRT SECURE NO WARNINGS 1
#include<string.h>
#include<stdio.h>
int main()
  int i;
  struct player {
    char name[50];
    int level;
    int score;
  };
  player p[10], *ptr;
  for (i = 0; i < 10; i++){
    strcpy(p[i].name, "Anonymous");
    p[i].level = i;
     p[i].score = i*100;
```

```
ptr = p;
for (i = 0; i < 10; i++) {
    printf("name : %s\t", ptr->name);
    printf("level : %d\t", ptr->level);
    printf("score : %d\n", ptr->score);
    ptr++;
    }
    return 0;
}
```

```
Anonymous
                          level
                                           score:
name :
       Anonymous
                          level
                                           score:
       Anonymous
                                           score:
       Anonymous
                                           score
       Anonymous
name
                          level
                                           score
       Anonymous
                          level
                                                   500
name
                                           score
       Anonymous
                          level
                                                   600
                                           score
       Anonymous
                          level
                                                   700
                                           score:
name
       Anonymous
                          level
                                           score
       Anonymous
                          level
                                           score:
```

File

Text file / Binary file การอ่าน / เขียนไฟล์

File

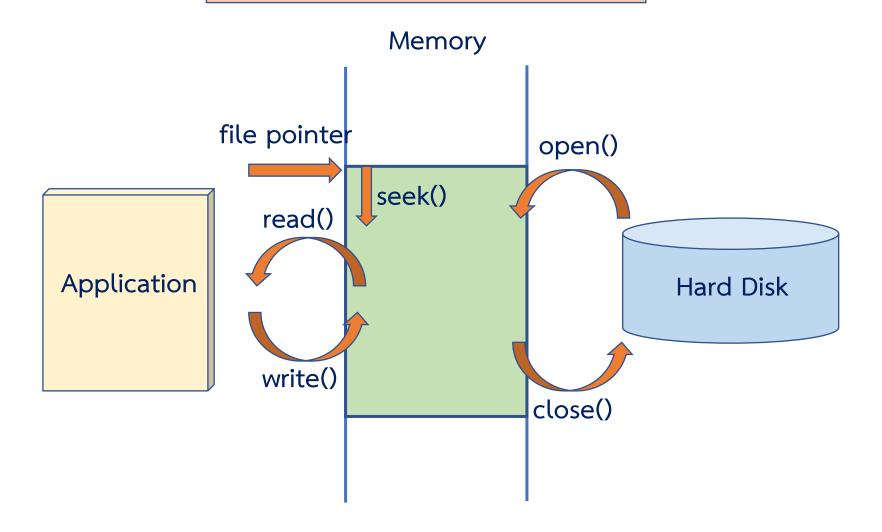
Text

- เป็นไฟล์ที่มีการตีความข้อมูลเป็นราย byte
- ไฟล์ชนิดนี้สามารถอ่านเขียนได้โดยใช้ editor ทั่วไปเช่น notepad ได้

Binary

- เป็นไฟล์ที่มีการตีความข้อมูลเป็นราย block โดยมี การกำหนด struct ในการอ่านเขียนข้อมูล
- ไม่สามารถอ่านได้โดยใช้ editor ทั่วไปได้ ต้องใช้ การอ่านข้อมูลแบบเฉพาะ

การทำงานเกี่ยวกับไฟล์



การทำงานเกี่ยวกับไฟล์

1. การเปิดไฟล์ : fopen()

- OS นำไฟล์จาก disk เตรียมไว้ใน memory แล้วส่ง file pointer มาให้ application เป็นตำแหน่งที่ ข้อมูลไฟล์นั้นวาง
- ทำการ lock ไฟล์ตามรูปแบบการเปิดไฟล์ (อ่าน/เขียน)
- มี mode การเปิดไฟล์ 3 mode (อ่าน, เขียนทับ, เขียนเพิ่ม)

2. การปิดไฟล์ : fclose()

- OS ทำการเขียนข้อมูลจาก memory ลงที่ไฟล์ใน hard disk
- ปลดการ lock ไฟล์

3. การอ่านไฟล์ : fscanf() , fread()

- ดึงข้อมูลจาก memory ในตำแหน่งที่ file pointer ชื่อยู่มาเก็บไว้ในตัวแปร แล้วเลื่อน pointer ไป ตำแหน่งถัดไป

4. การเขียนไฟล์ : fprintf() , fwrite()

- การนำข้อมูลจากตัวแปร หรือหน่วยความจำ ไป copy ไว้ในหน่วยความจำที่ file pointer ชื่อยู่ แล้ว เลื่อน pointer ไปตำแหน่งถัดไป

5. การเลื่อนจุดที่จะอ่าน/เขียนไฟล์ : fseek()

- ทำการย้าย file pointer ไปยังตำแหน่งที่ต้องการ
- ใช้ในการอ่าน/เขียนข้อมูลในตำแหน่งที่กำหนด

การทำงานเกี่ยวกับ Text File

เปิดไฟล์ :

fp = fopen ("filename.txt","mode");

เขียนไฟล์:

fprintf(fp, "string....");

fprintf(fp, "format string", variables);

อ่านไฟล์ :

fscanf(fp, "format string", variables);

ปิดไฟล์ :

fclose(fp);

Mode:

"r" : อ่านไฟล์

"w" : เขียนไฟล์

"a" : เขียนเพิ่มท้ายไฟล์

เหมือน printf , scanf แต่ ไปอ่านเขียนที่ไฟล์แทน

ตัวอย่างการเขียน Text File

```
#define _CRT_SECURE_NO_WARNINGS 1
#include<stdio.h>
int main()
  FILE* fp;
  fp = fopen("mytestfile.txt", "w");
  fprintf(fp, "%d %d %d\n", 100, 200, 500);
  fclose(fp);
  return 0;
                                   mytestfile.txt - Notepad
                                                                    X
                                  <u>File Edit Format View Help</u>
                                   100 200 500
```

Activity

ให้นักศึกษาทดลอง รันโปรแกรมตัวอย่างซ้ำ 2-3 ครั้ง แล้วแต่ละครั้งให้เปิดไฟล์ mytestfile.txt เพื่อตรวจสอบการเปลี่ยนแปลง

ให้นักศึกษาทดลอง เปลี่ยน mode ให้เป็นการเขียนต่อท้ายไฟล์ แล้วรันโปรแกรมอีก 2-3 ครั้ง แล้วแต่ละครั้งให้เปิดไฟล์ mytestfile.txt เพื่อตรวจสอบการเปลี่ยนแปลง

ตัวอย่างการอ่าน Text File

```
#define CRT SECURE NO WARNINGS 1
#include<stdio.h>
int main()
  FILE* fp;
  int a, b, c, suma=0, sumb=0, sumc = 0;
  fp = fopen("mytestfile.txt", "r");
  while (!feof(fp)){
  a = 0; b = 0; c = 0;
  fscanf(fp, "%d %d %d", &a, &b, &c);
  suma += a; sumb += b; sumc += c;
  printf("%d %d %d\n", suma, sumb, sumc);
  fclose(fp);
                                           Microsoft Visual Studio Debug Console
  return 0;
```

400 800 2000

C:\Users\robin\Source\Repos\Project12\Debug\Project12.exe (process 2704) exited with code 0.

To automatically close the console when debugging stops, enable Tools->Options->De



ให้นักศึกษาทดลอง เขียนโปรแกรมเก็บข้อมูล
ชื่อผู้เล่น , level และ คะแนนของผู้เล่น จำนวน 5 คน
เก็บไว้ใน Text File

*Hint : ชื่อผู้เล่นไม่ควรมีเว้นวรรค

การทำงานเกี่ยวกับ Binary File

เปิดไฟล์:

fp = fopen ("filename.txt","mode");

เขียนไฟล์ :

int fwrite(ptr,size,items,stream);

อ่านไฟล์ :

int fread(dest,size,items,stream);

เลื่อนตำแหน่ง pointer :

int fseek(stream, offset, where);

ปิดไฟล์:

fclose(fp);

Mode:

"r" : อ่านไฟล์

"w" : เขียนไฟล์

"a" : เขียนเพิ่มท้ายไฟล์

การเขียน Array & Structure ลงไฟล์ (binary file)

int fwrite(ptr,size,items,stream);

ptr คือ ตำแหน่งของตัวแปร struct หรือ array ในหน่วยความจำ

(address) ที่ต้องการจะเขียนลงไฟล์

size คือ ขนาดของตัวแปร struct หรือ array

items คือ จำนวนของ struct หรือ array ที่ต้องการจะเขียนลงไฟล์

stream คือ File Pointer

fwrite จะ return จำนวน bytes ที่ได้เขียนลงในไฟล์ และจะ return 0 เมื่อไม่สามารถเขียนข้อมูลลงไฟล์ได้

การอ่าน Array & Structure จากไฟล์

int fread(dest, size, items, stream);

dest คือ ตำแหน่งของตัวแปร struct หรือ array ที่ใช้เก็บค่าที่อ่านจากไฟล์

size คือ ขนาดของตัวแปร struct หรือ array ที่ต้องการอ่านเข้ามา

items คือ จำนวนของ struct หรือ array ที่ต้องการจะอ่านจากไฟล์

stream คือ File Pointer

fread จะ return ค่าเป็นจำนวน bytes ที่อ่านได้ และจะ return 0 มันทำการอ่านจนถึงตำแหน่งสุดท้ายของไฟล์

การใช้คำสั่ง fseek

int fseek(stream, offset, where);

stream คือ File Pointer

Offset คือ ระยะที่ต้องการให้ตัวชี้ตำแหน่งในไฟล์เคลื่อนไปในแต่ละครั้ง

where เป็นกำหนดตำแหน่งของตัวชี้ตำแหน่งในไฟล์ มีได้ 3 ค่าคือ

0: ตัวชี้ตำแหน่งในไฟล์เริ่มต้นที่ตำแหน่งแรกสุดของไฟล์

1: ตัวชี้ตำแหน่งในไฟล์เริ่มต้นที่ตำแหน่งปัจจุบัน

2: ตัวชี้ตำแหน่งในไฟล์เริ่มต้นที่ตำแหน่งสุดท้ายของไฟล์

fseek จะ return 0 ถ้าสามารถเลื่อนตัวชี้ตำแหน่งในไฟล์ได้สำเร็จ และจะ return ค่า int ใด ๆ ที่ไม่เท่ากับ 0 เมื่อดำเนินการตาม parameter ไม่สำเร็จ

โปรแกรมเขียน Structure ลงไฟล์

```
int main()
                                                          StRecord - Notepad
                      Name : Kmitl
                                        File Edit Format View Help
  FILE *fptr;
                                       K<u>mitl ;6?‼งย6?‼ว#↑_ 48010000 -</u>
                      ID: 48010000
  struct student {
                      Age : 45
     char name[20];
     char ID[9];
     int age; } s;
  fptr = fopen("StRecord.txt", "w");
  printf("Name : "); scanf("%s",s.name);
  printf("ID : "); scanf("%s",s.ID);
  printf("Age : "); scanf("%d", &s.age);
  fwrite(&s, sizeof(struct student), 1, fptr);
  fclose(fptr);
  return 0;
```

โปรแกรมอ่าน Structure จากไฟล์

```
int main()
{ FILE *fptr;
                        Name: Kmitl
  struct student
                        ID: 48010000 and Age: 45
  { char name[20];
     char ID[9];
     int age;
  }s;
  fptr = fopen("StRecord.txt", "r");
  if (fptr == (FILE *)NULL) printf("Cannot open file\n");
  else
     while (fread(&s, sizeof(struct student), 1, fptr)!=0)
     { printf("Name: %s\n", s.name);
       printf("ID: %s and Age: %d\n", s.ID,s.age);
       printf("----\n");
  fclose(fptr);
  return 0;
                                                   26
```

```
FILE *fptr; int noffset;
                                                           StRecord - Notepad
struct student {
                                   File Edit Format View Help
   char name[20];
                                   Student A 🛚,?∢ย6,ฎ"48010000 ∢
   char ID[9];
                                   Student B 🗓,?┫ย6,ฏ"48012000 🕽
                                   Student C 🗓,?∢ย6,ฏ"48015000 ¶
   int age;
} s;
fptr = fopen("StRecord.txt", "w");
strcpy (s.name, "Student A"); strcpy (s.ID, "48010000");
s.age = 17;
fwrite(&s, sizeof(struct student), 1, fptr);
strcpy (s.name, "Student B"); strcpy (s.ID, "48012000");
s.age = 18;
fwrite(&s, sizeof(struct student), 1, fptr);
strcpy (s.name, "Student C"); strcpy (s.ID, "48015000");
s.age = 20;
fwrite(&s, sizeof(struct student), 1, fptr);
fclose(fptr);
```

```
fptr = fopen("StRecord.txt", "r");
noffset = 0 * sizeof(struct student);
if (fseek( fptr, noffset, 0) == 0 )
{    if (fread(&s,sizeof(struct student),1,fptr) != 0)
      {       printf("Name: %s\n", s.name);
            printf("ID: %s and Age: %d\n", s.ID,s.age);
            printf("-----\n");
      }
}
fclose(fptr);
```

```
Strecord - Notepad
File Edit Format View Help

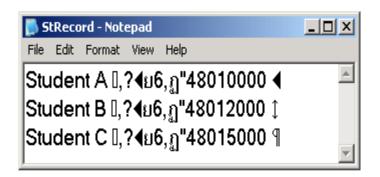
Student A [],? ◀ย6,ฏ"48010000 ↓

Student B [],? ◀ย6,ฏ"48012000 ↓

Student C [],? ◀ย6,ฏ"48015000 ¶
```

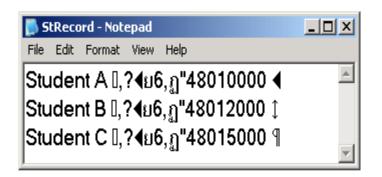
```
Name: Student A
ID: 48010000 and Age: 17
```

```
fptr = fopen("StRecord.txt", "r");
noffset = 1 * sizeof(struct student);
if (fseek( fptr, noffset, 0) == 0 )
{    if (fread(&s,sizeof(struct student),1,fptr) != 0)
        {        printf("Name: %s\n", s.name);
            printf("ID: %s and Age: %d\n", s.ID,s.age);
            printf("-----\n");
        }
}
fclose(fptr);
```



```
Name: Student B
ID: 48012000 and Age: 18
```

```
fptr = fopen("StRecord.txt", "r");
noffset = 2 * sizeof(struct student);
if (fseek( fptr, noffset, 0) == 0 )
{    if (fread(&s,sizeof(struct student),1,fptr) != 0)
      {       printf("Name: %s\n", s.name);
            printf("ID: %s and Age: %d\n", s.ID,s.age);
            printf("-----\n");
      }
}
fclose(fptr);
```



```
Name: Student C
ID: 48015000 and Age: 20
```

คำสั่งการอ่านข้อมูลจากไฟล์ และเขียนข้อมูลลงไฟล์

```
fscanf (fp,"format",&variable);
     var = getc(fp);
     fgets(var,length,fp);
```



ให้นักศึกษาทดลอง เขียนโปรแกรมเก็บข้อมูล
ชื่อผู้เล่น , level และ คะแนนของผู้เล่น จำนวน 5 คน
เก็บไว้ใน Binary File

*Hint : ชื่อผู้เล่นไม่ควรมีเว้นวรรค