<Adv C & App/>

Advanced C Programming And It's Application

Struct II

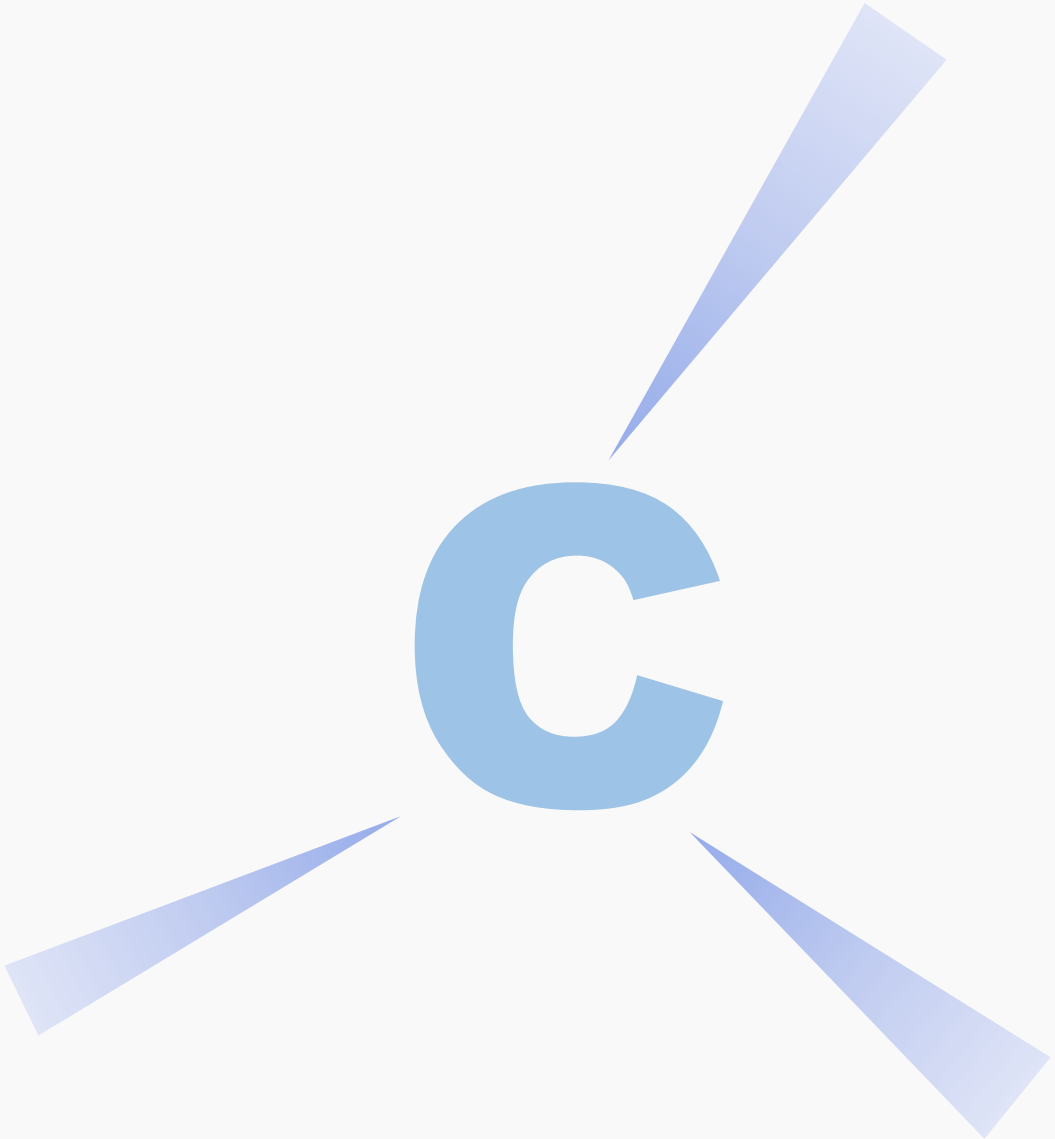
Assistant Prof. Chan, Chun-Hsiang

Department of Artificial Intelligence, Tamkang University

Dec. 29, 2021

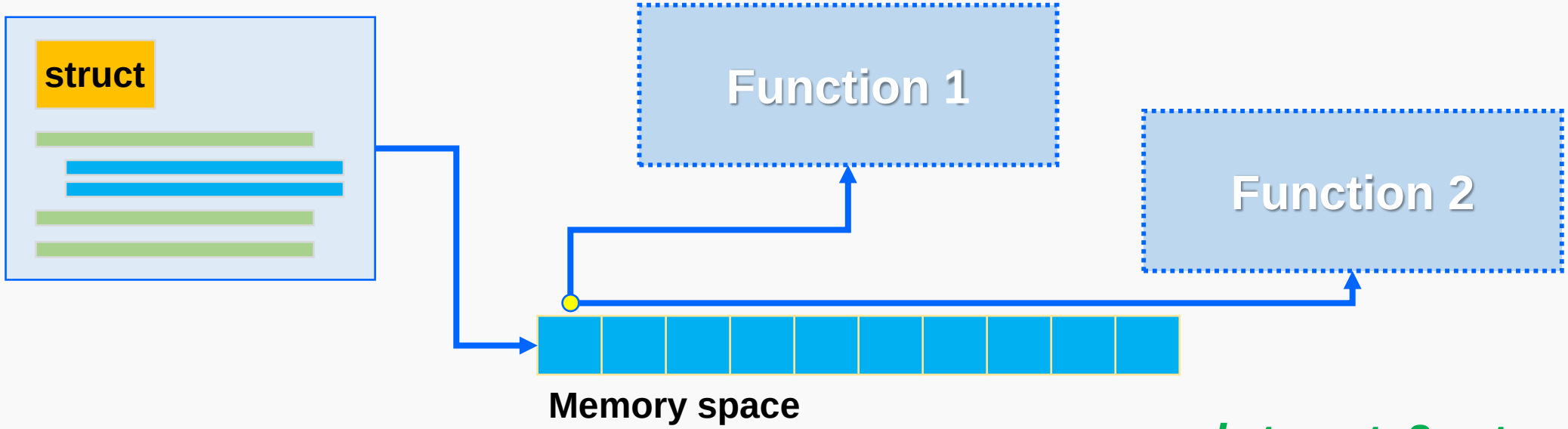
</ Adv C & App >

大綱



struct and pointer – function

既然有了struct，就一定會提到pointer啦!
我們可以利用pointer去存取struct變數的記憶體位置；換句話說，
就可以在function直接存取struct變數的資料並做加值的動作。



struct and pointer

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int main(){
        /*Ex 13-13: pointer to struct*/
        printf("/*Ex 13-13: pointer to struct*/\n");
        Flight EK367;
        Flight EK367 = {"EK367", "Emirates Airline", "TPE", "DXB", 7, 459, 9.917};
        Flight *r = &EK367;
        printf("Flight Number (%s) is operated by %s.\n", (*r).flightNo, (*r).airline);
```

```
/*Ex 13-13: pointer to struct*/
Flight Number (EK367) is operated by Emirates Airline.
```

struct and pointer with an arrow

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int main(){
        /*Ex 13-14: pointer to struct with arrow*/
        printf("/*Ex 13-14: pointer to struct with arrow*/\n");
        Flight EK367;
        Flight EK367 = {"EK367", "Emirates Airline", "TPE", "DXB", 7, 459, 9.917};
        Flight *r = &EK367, *s;
        s = r;
        printf("Flight Number (%s) is operated by %s.\n", s->flightNo, s->airline);
```

```
/*Ex 13-14: pointer to struct with arrow*/
Flight Number (EK367) is operated by Emirates Airline.
```

struct and function

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int get_flightNo(Flight F){
        printf("Flight Number is %s.\n", F.flightNo);}
int get_flightNo_arrow(Flight *F){
        printf("Flight Number is %s.\n", F->flightNo);}
int main(){
        /*Ex 13-15: struct to function*/
        printf("/*Ex 13-15: struct to function*/\n");
        Flight EK367 = {"EK367", "Emirates Airline", "TPE", "DXB", 7, 459, 9.917};
        Flight *p = &EK367;
        get_flightNo(EK367);
        get_flightNo_arrow(p);}
```

```
/*Ex 13-15: struct to function*/
Flight Number is EK367.
Flight Number is EK367.
```

struct with pointer and function

Lab 13-4:

將以下資訊以Lab13-3的Car結構變數，設計一個function。

```
void printCarDetails( Car C ){
        ...
        // 將所有資訊印出來!
        ...
}
```

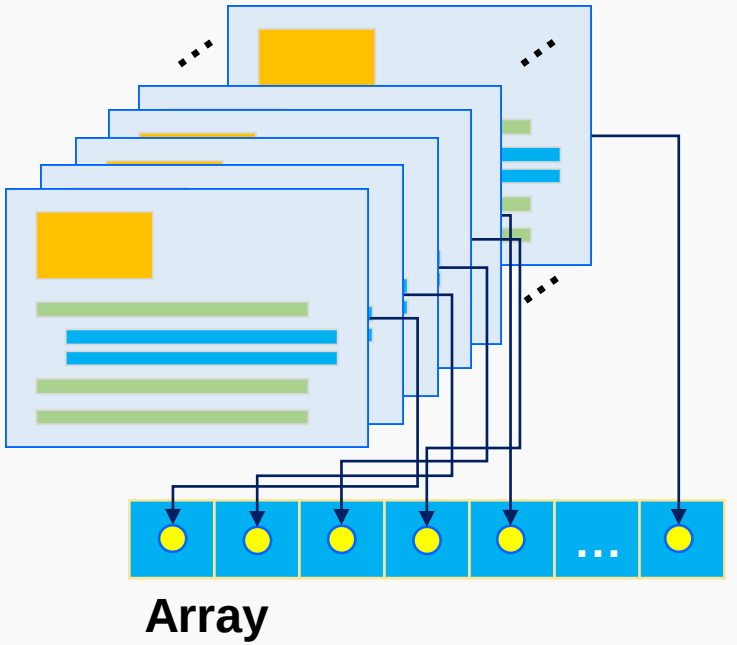
為了證明可以用，請任印出兩個Car結構變數。

struct Array

既然我們可以用pointer存取struct變數的記憶體位置，那麼應該就會想到下個問題，如果我今天需要建立1000筆航班資料的屬性匯入，總不可能宣告1000個變數吧! 所以說這個時候就可以用struct array的方式並在一起囉!

用法其實很簡單:

```
Flight EK[1000];
for (…){…};
```



struct Array

```c
#include <stdio.h>
#include <string.h>
struct flight{... SKIP ...};
typedef struct flight Flight;
int main(){
        /*Ex 13-16: struct array*/
        printf("/*Ex 13-16: struct array*/\n");
        int i; Flight Emirates[10];
        strcpy(Emirates[0].flightNo, "EK367");
        strcpy(Emirates[1].flightNo, "EK366");
        strcpy(Emirates[2].flightNo, "EK362");
        for (i=0; i<3; i++){
                printf("Flight Number: %s\n", Emirates[i].flightNo);
        }
}
```

```
/*Ex 13-16: struct array*/
Flight Number: EK367
Flight Number: EK366
Flight Number: EK362
```

struct Array

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
int put_flightNo(Flight *F, char flightNoList[][6], int size){
        int idx;
        for (idx=0; idx<size; idx++){
                strcpy(F[idx].flightNo, flightNoList[idx]);
        }
}
int print_flightNo(Flight *F, int size){
        int idx;
        for (idx=0; idx<size; idx++){
                printf("Flight Number is %s.\n", F[idx].flightNo);
        }
}
```

struct Array

```
int main(){
        /*Ex 13-17: struct array in function*/
        printf("/*Ex 13-17: struct array in function*/\n");
        int i;

        char EmiratesFlightInfo[3][6] = {"EK367", "EK366", "EK362"};
        Flight Emirates[10];

        put_flightNo(Emirates, EmiratesFlightInfo, 3);
        print_flightNo(Emirates, 3);
}
```

```
/*Ex 13-17: struct array in function*/
Flight Number is EK367.
Flight Number is EK366.
Flight Number is EK362.
```

struct array

Lab 13-5:

仿造EX13-17的做法，將Lab13-3資訊利用一個function儲存Cars struct array變數，再利用另一個function將所有資訊印出來。

```
Car cars[7];
put_carInfo(…);
print_carInfo(…);
```

Nested struct

雖然struct array已經可以讓我們方便許多，但如果今天我們有很多家航空公司的資料，就不太可能用array的做法。因為我們沒辦法很快了解這些航班資訊那些是同一個航空公司的班機。

於是這個時候，我們就會用到nested struct，意即一層以上的struct結構，這時候屬性資料就可以有多層次的架構。

這個概念其實在現今的noSQL DB中很常見，像是social media的資料就是這種結構，一篇post會有貼文內容、按讚數，留言數以及分享數；此外底下可能有comment也會有內容、按讚數以及reply數，再加上也會出現comment's reply或是reply's reply。

Nested struct

```
/*Ex 13-18: nested struct*/
Emirates-Emirates-EK366
```

```c
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
struct airlines{
        char airline[10];
        Flight flight[10];
};
typedef struct airlines Airlines;
int main(){
        /*Ex 13-18: nested struct*/
        printf("/*Ex 13-18: nested struct*/\n");
        int i;
        Airlines Emirates;
        char EmiratesFlightInfo[6][6] = {"EK367", "EK366", "EK362", "EK031", "EK943", "EK358"};
        strcpy(Emirates.airline, "Emirates");
        strcpy(Emirates.flight[0].airline, "Emirates");
        strcpy(Emirates.flight[0].flightNo, "EK366");
        printf("%s-%s-%s\n", Emirates.airline, Emirates.flight[0].airline, Emirates.flight[0].flightNo);
}
```

Nested struct

```
#include <stdio.h>
#include <string.h>
struct flight{… SKIP …};
typedef struct flight Flight;
struct airlines{… SKIP …};
typedef struct airlines Airlines;
int put_flightNo(Airlines *A, char flightNoList[][6], int size){
        int idx;
        for (idx=0; idx<size; idx++){
                strcpy(A->flight[idx].flightNo, flightNoList[idx]);
        }
}
int print_flightNo(Airlines A, int size){
        int idx;
        for (idx=0; idx<size; idx++){
                printf("Flight Number is %s.\n", A.flight[idx].flightNo);
        }
}
```

Nested struct

```
/*Ex 13-19: nested struct*/
Flight Number is EK367.
----------------------
Flight Number is EK367.
Flight Number is EK366.
Flight Number is EK362.
Flight Number is EK031.
Flight Number is EK943.
Flight Number is EK358.
```

```
int main(){
        /*Ex 13-19: nested struct*/
        printf("/*Ex 13-19: nested struct*/\n");
        int i;
        Airlines Emirates;
        char EmiratesFlightInfo[6][6] = {"EK367", "EK366", "EK362",
                                         "EK031", "EK943", "EK358"};
        put_flightNo(&Emirates, EmiratesFlightInfo, 6);
        printf("Flight Number is %s.\n", Emirates.flight[0].flightNo);
        printf("----------------------\n");
        print_flightNo(Emirates, 6);
}
```

Nested struct

Lab 13-6:

仿造EX13-19的做法，將Lab13-3資訊中的廠牌設為第一層結構。再利用一個function儲存資料匯入，而另一個function將所有資訊印出來。

```
struct Car{…};
struct CARS{
        char carBrand[…]; //品牌名稱
        Car car[…]; //如同前面的屬性資料
};
```

```
// In main…
put_carInfo(…);
print_carInfo(…);
```

參考資料

1. https://zh.wikipedia.org/wiki/%E7%BB%93%E6%9E%84%E4%BD%93_(C%E8%AF%AD%E8%A8%80)
2. https://openhome.cc/Gossip/CGossip/StructABC.html
3. https://ithelp.ithome.com.tw/questions/10190575
4. https://blog.xuite.net/tzeng015/twblog/113271886
5. 蔣宗哲教授講義