2017

程式設計

加強班

程式設計與實習(一)

BY 孫茂勛

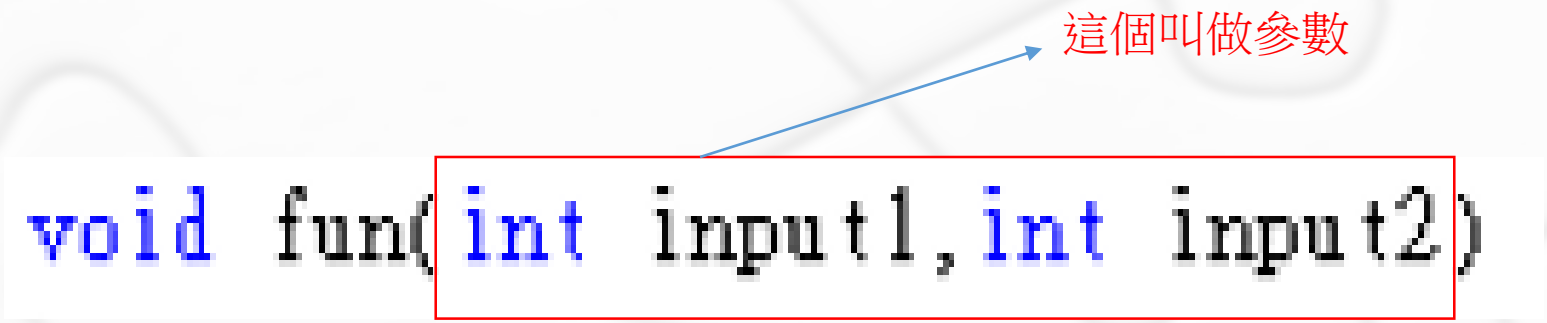
Email:JOHN85051232@GMAIL.COM

花點時間複習一下

- 不然你應該聽不懂今天要講什麼

Pointer

- 大概知道指標是什麼東西後.....
- 我們回頭看下之前學過的Function是如何傳遞參數的.....



Pointer

```c
void swap(int a , int b)
{
    int temp = 0;
    temp = a;
    a = b;
    b = temp;
}
```

```c
int main()
{
    int a = 10;
    int b = 20;
    printf("交換前: a = %d b = %d\n",a,b);
    swap(a,b);
    printf("交換後: a = %d b = %d\n",a,b);
    system("pause");
    return 0;
}
```

執行結果應該是什麼？

Function

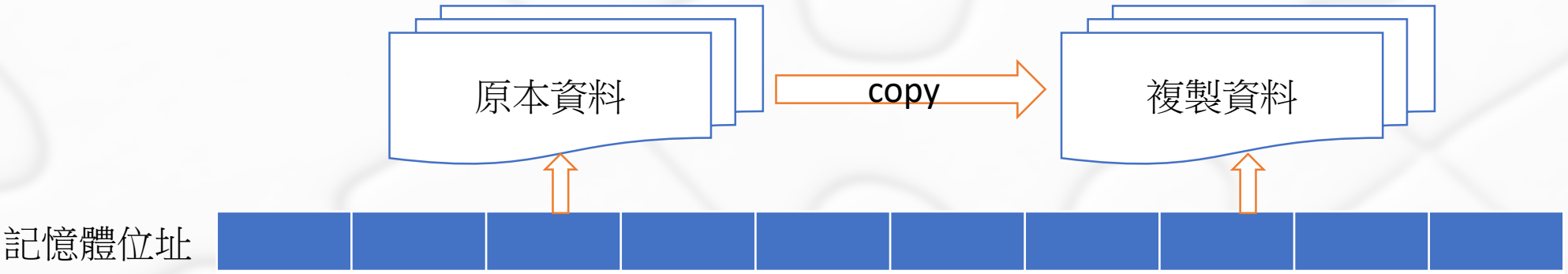
- 參數的傳遞方式（Passing Arguments）：
- Pass By Value
- Pass By Address
- ~~Pass By Reference(you will study in C++)~~

Function

Pass By Value(傳值)：

- 副程式的參數會將要使用的資料複製一份，副程式會以被複製的資料進行操作。
- 不會影響到原來的資料。

副程式會用這份資料去進行操作。



Function

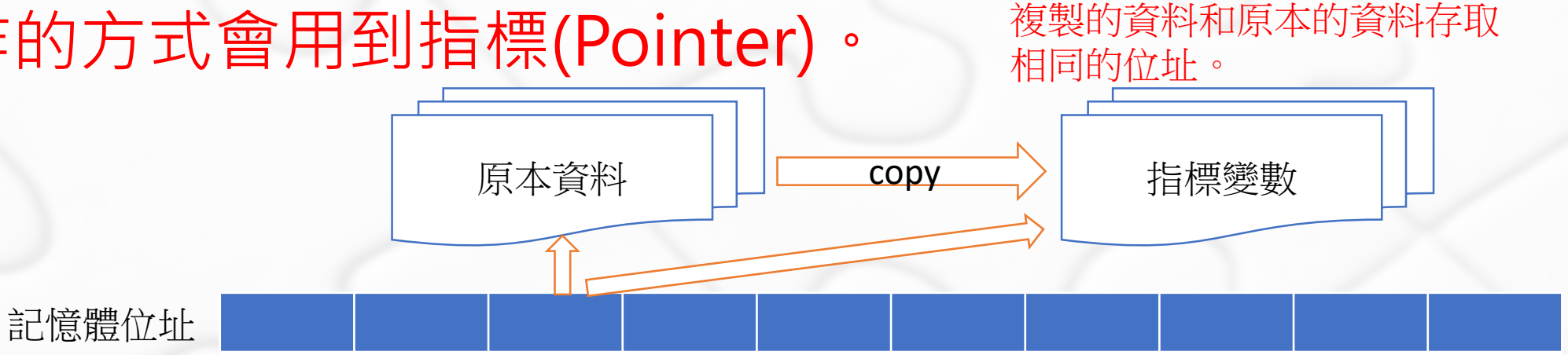
```c
#include <stdio.h>
#include <stdlib.h>
//pass by value
void fun(int input1,int input2)
{
     input1 = input1 + 10;
     input2 = input2 + 10;

     printf("在function中a的值為:%d\n",input1);
     printf("在function中b的值為:%d\n",input2);
}
int main()
{
     int a = 5;
     int b = 10;
     printf("在呼叫function前a的值為:%d\n",a);
     printf("在呼叫function前b的值為:%d\n",b);
     fun(a,b);
     printf("在呼叫function後a的值為:%d\n",a);
     printf("在呼叫function後b的值為:%d\n",b);
     system("PAUSE");
     return 0;
}
```

```
在呼叫function前a的值為:5
在呼叫function前b的值為:10
在function中a的值為:15
在function中b的值為:20
在呼叫function後a的值為:5
在呼叫function後b的值為:10
請按任意鍵繼續 . . .
```

Function

Pass By Address(傳址)：

- 副程式的參數會將要使用資料的**記憶體位置**複製一份，副程式會使用指標再該位置上進行操作。
- 會影響到原來的資料。
- 操作的方式會用到指標(Pointer)。



Function

```c
#include <stdio.h>
#include <stdlib.h>
//pass by address
void fun(int *input1,int *input2)
{
     *input1 = *input1 + 10;
     *input2 = *input2 + 10;

     printf("在function中a的值為:%d\n",*input1);
     printf("在function中b的值為:%d\n",*input2);
}
int main()
{
     int a = 5;
     int b = 10;
     printf("在呼叫function前a的值為:%d\n",a);
     printf("在呼叫function前b的值為:%d\n",b);
     fun(&a,&b);
     printf("在呼叫function後a的值為:%d\n",a);
     printf("在呼叫function後b的值為:%d\n",b);
     system("PAUSE");
     return 0;
}
```

宣告要傳入的參數是指標變數

呼叫時必須給**變數的"記憶體位址"**

```
在呼叫function前a的值為:5
在呼叫function前b的值為:10
在function中a的值為:15
在function中b的值為:20
在呼叫function後a的值為:15
在呼叫function後b的值為:20
請按任意鍵繼續 . . .
```

Binary Search

```c
int main()
{
    int arr[10] = {0,1,2,3,4,5,6,7,8,9};
    printf("%d\n",binary_sort(arr,10,3));
    system("PAUSE");
    return 0;
}
```

一維陣列的參數傳遞需要告知起始位置
(arr代表arr[0]的記憶體位址)

=>arr和 &arr[0] 是一樣的意思

```c
int binary_sort(int arr[],int num,int find)
{
    int min = 0;
    int max = num-1;
    int mid = (min + max) / 2;
    while(min <= max)
    {
        if(arr[mid] == find)
        {
            return find;
        }
        else if(arr[mid] < find)
        {
            min = mid + 1;
        }
        else // arr[mid] > find
        {
            max = mid -1;
        }
        mid = (min + max) / 2;
    }
    return -1;
}
```

告訴function我要傳的參數是一個陣列

Function

Pass By Reference(傳參考)：

- 副程式的參數會新增一個連結，連結至原本的資料，副程式會直接對該資料進行操作。
- 會影響到原來的資料。
- 與Pass By address的差別：傳址在實作上會多新增指標去複製記憶體位置，傳參考則是直接創立連結。

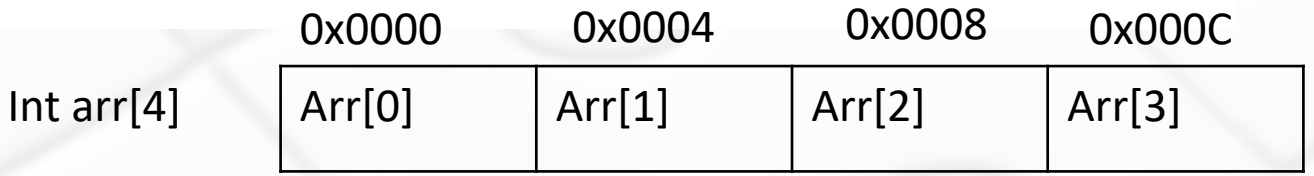
指標的應用

```c
//利用指標變數去更改其他變數的內容
int a = 10;
int b = 20;
char c = 's';

int *ptr = &a; //宣告ptr是一個指標變數，指向a的記憶體位址
*ptr = 99; //將(ptr指向的位址->a)內容改成99
printf("a = %d *ptr = %d\n",a,*ptr);

ptr = &b;
*ptr = 990;
printf("b = %d *ptr = %d\n",b,*ptr);

ptr = &c; //錯誤，int的指標無法指向char的變數
```

Pointer

- 陣列是一連串的記憶體空間。
- 且陣列名稱本身就是一個指向陣列開頭的位址。

```
int arr[5] = {1,2,3,4,5};
printf("arr的位址 = %p \n",arr);//陣列名字是陣列起始的記憶體位址
printf("arr[0]的位址 = %p \n",&arr[0]);
```



所以也可以透過移動指標指向不同的位址來存取陣列

THANK YOU