

Pesky Pointers

首先，此篇開頭先介紹了何謂指標，指標就是一個可以儲存記憶體位址的變數，所以在 main 程式中我們呼叫指標實質上是在使用該變數的「記憶體位址」而不是變數真實的「值」，並且文章也補充到 Java 這款高階程式語言也有類似指標的功能，叫作 `reference`，比起 C 語言的指標，`reference` 是比較嚴謹且一致的，幾乎是不會出錯的，因為在使用 C 語言的指標的時候，它並不會防止我們在運程式時意外的使用到其他的東西，但是 Java 的 `reference` 有明確的規定，它只會對它自己本身的 `object` 去做 `reference`，並且只能通過 `reference` 去操作用於保存的 `object` 的記憶體空間。

接下來就是一些指標的應用，例如我從 `swap` 的例子中學到如果只是單純的傳值進去 `swap` 的話，改變後的值是無法回傳給 main 程式的，還有我學到一個副程式是無法回傳 1 個以上的值的，所以應該傳這些值的記憶體位址進去 `swap` 才對，並且 `&` 代表的就是取得位址，另外，我也學到指標與矩陣之間的關係，事實上我認為指標就是矩陣，只是宣告的方式不同而已，如果當初宣告的是矩陣的形式，則程式中只能用矩陣來做運算，但是如果當初宣告的是指標，則不管是以指標或是矩陣的形式都可以在程式中做運算，並且 $a[i] = *(a + i)$ ，意思就是矩陣 `a` 的第 `i` 項的值其實就是指標指向起始位址後向後數第 `i` 項的值。

在 `Pointer Arithmetic` 中，我注意到 `i++` 或是 `++i` 的重要性以及 `atoi()` 函式的用法，以 `strcpy.c` 為例，我發現程式跑 `src[len++]` 時，因為 `len++` 的緣故，

所以導致多算到空字元，使得 len 為 6 但正確值應該為 5，所以應該改為

++len 才不會多算到空字元，但是我又發現最後輸出應該還是錯的，因為關於

其中一個輸出“printf("dst array %s and last element %c\n", dst, dst[len]);”，

dst[len] = dst[5] = 空格，但是 dst 的 last element 應該為 dst[11] = 空字

元，所以我又新增一個 len_dst 去計算 dst 的字串長度，並把輸出改成

dst[len_dst]，這樣 last element 才是對的。

程式圖如下：

```
1  #include<stdio.h>
2  #include<stdlib.h>
3
4  char *my_strcpy(char *s1, const char *s2) {
5      register char *d = s1;
6      printf("s2 address %p, its contents is a pointer %p to first char %c \n", (void *)&s2, (void *)s2, *s2);
7      printf("s1 address %p, its contents is a pointer %p to first char %c \n", (void *)&s1, (void *)s1, *s1);
8      while (*d++ = *s2++);
9      return(s1);
10 }
11
12 int main(void) {
13     char src[] = "cs23!";
14     char dst[] = "Hello hello";
15     char *curdst;
16     int len_src = 0, len_dst = 0;
17
18     printf("src address %p and first char %c \n", (void *)&src, src[0]);
19     printf("dst address %p and first char %c \n", (void *)&dst, dst[0]);
20     //計算字串長度
21     while(src[++len_src]);
22     while(dst[++len_dst]);
23     //atoi無有效的轉換可以執行，則回傳0
24     printf("src array %s and last element %d\n", src, atoi(&src[len_src]));
25     printf("dst array %s and last element %c\n", dst, dst[len_dst]);
26
27     curdst= my_strcpy(dst, src);
28     //因為copy src 最後一個字元(src[5])為'\0'空字元，所以輸出只輸出到dst[5]但事實上dst[6]以後仍是hello
29     printf("dst array %s and last element %d\n", dst, atoi(&dst[len_dst]));
30
31     return 0;
32 }
```

```
src address 0061FF0E and first char c
dst address 0061FF02 and first char H
src array cs23! and last element 0
dst array Hello hello and last element
s2 address 0061FEF4, its contents is a pointer 0061FF0E to first char c
s1 address 0061FEF0, its contents is a pointer 0061FF02 to first char H
dst array cs23! and last element 0
```