

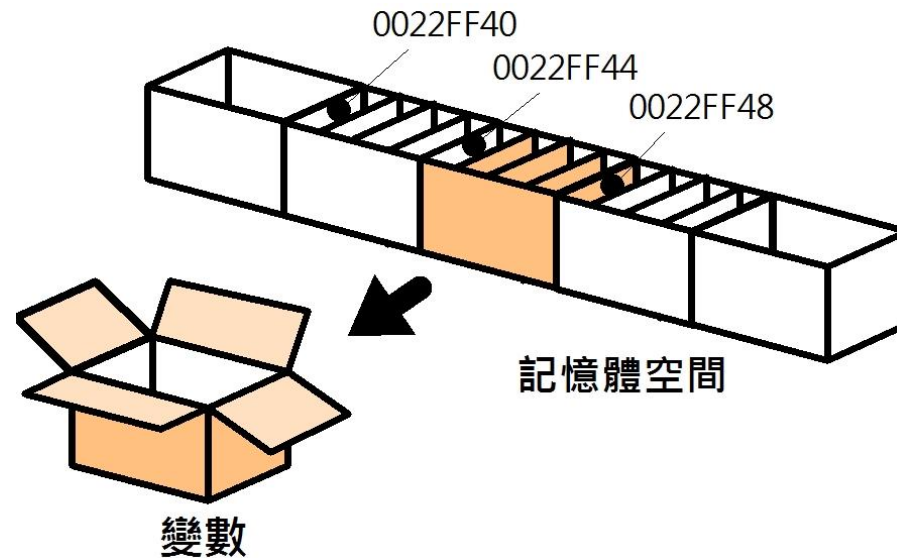
C 程式語言 5

區域與全域變數

- 區域變數（Local Variables）：程式區塊有效範圍的變數是一種區域變數，例如：在函數中宣告的變數或參數，變數只能在宣告的函數中使用，在函數外的程式碼並無法存取此變數。
- 全域變數（Global Variables）：程式檔案有效範圍的變數屬於一種全域變數，例如：在函數外宣告的變數，整個程式檔案都可以存取此變數。

電腦的記憶體位址

- 記憶體空間如同一排大樓信箱，每一個儲存單位擁有數字編號的「位址」（Address），即信箱上的門牌號碼，在每一單位的記憶體內容是資料，儲存資料佔用的記憶體空間大小，需視使用的資料型態而定：



C語言的指標

- 「指標」（**Pointers**）是 C 語言的低階程式處理功能，可以直接存取電腦的記憶體位址。指標是一種變數，只是變數內容不是常數值，而是其他變數的「位址」，其值是指向其他變數的位址，能夠讓我們間接取得其他變數的值。

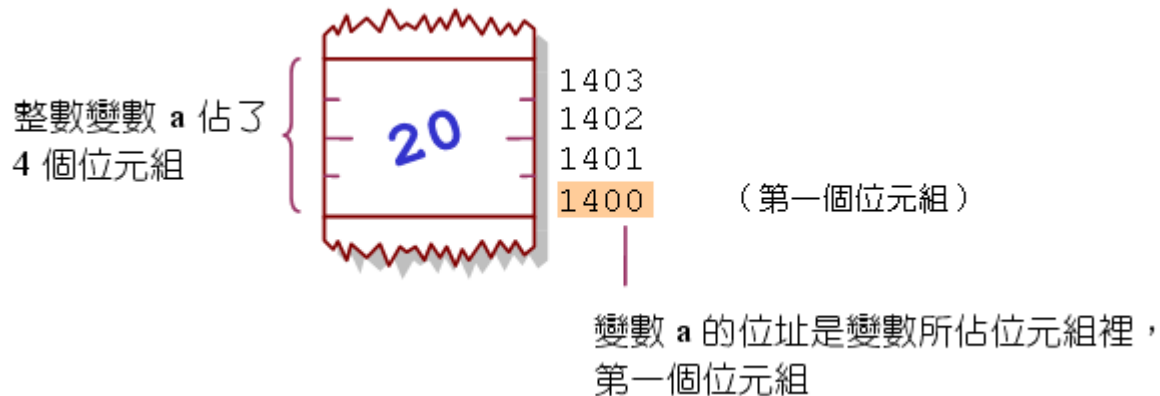
C語言的指標

- 3個變數a、b和ptr，變數值分別為65、30和0022FF44，ptr變數值0022FF44是變數b的記憶體位址，ptr是一個指標，為什麼叫指標？因為它是一個指向其他變數位址的變數，可以引導我們找到其他變數的值，以此例就是變數b的值：



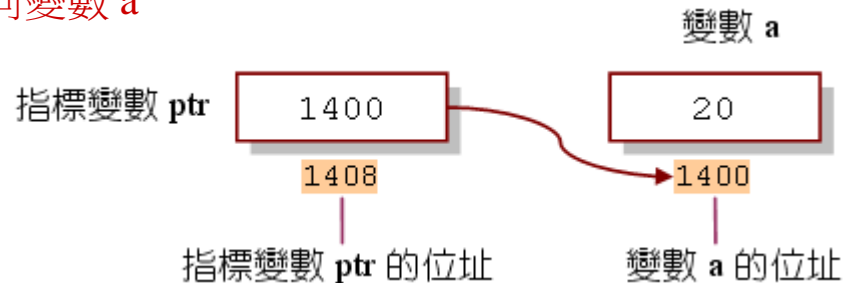
變數的位址

- 變數的位址是它所佔位元組裡，第一個位元組的位址：



- 指標變數是用來存放變數在記憶體中的位址

指標 **ptr** 指向變數 **a**

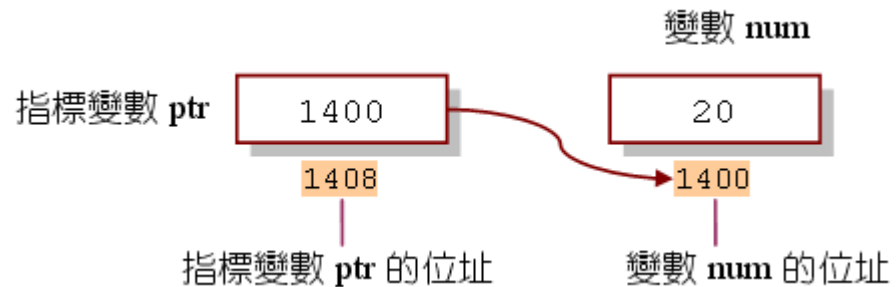


指標變數的宣告

資料型態 *指標變數;
或
資料型態* 指標變數;

— 指標變數使用的範例：

- `int num=20;`
- `int *ptr;`
- `ptr=#`

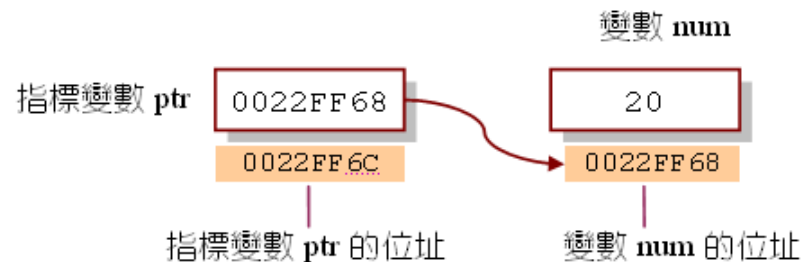


假設 num 的位址為 1400

指標變數的使用範例

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int *ptr, num=20;
    ptr=&num;                      /* 將num的位址設給指標ptr存放 */
    printf("num=%d, &num=%x\n",num,&num);
    printf("*ptr=%d, ptr=%x, &ptr=%x\n",*ptr,ptr,&ptr);

    system("pause");
    return 0;
}
```



傳遞指標到函數

- 接收指標的函數：

接收指標之函數的語法

```
傳回值型態 函數名稱 (資料型態 *指標變數)
{
    /* 函數的本體 */
}
```

```
void func(int *ptr)
{
    /* 函數的本體 */
}
```

傳遞指標到函數

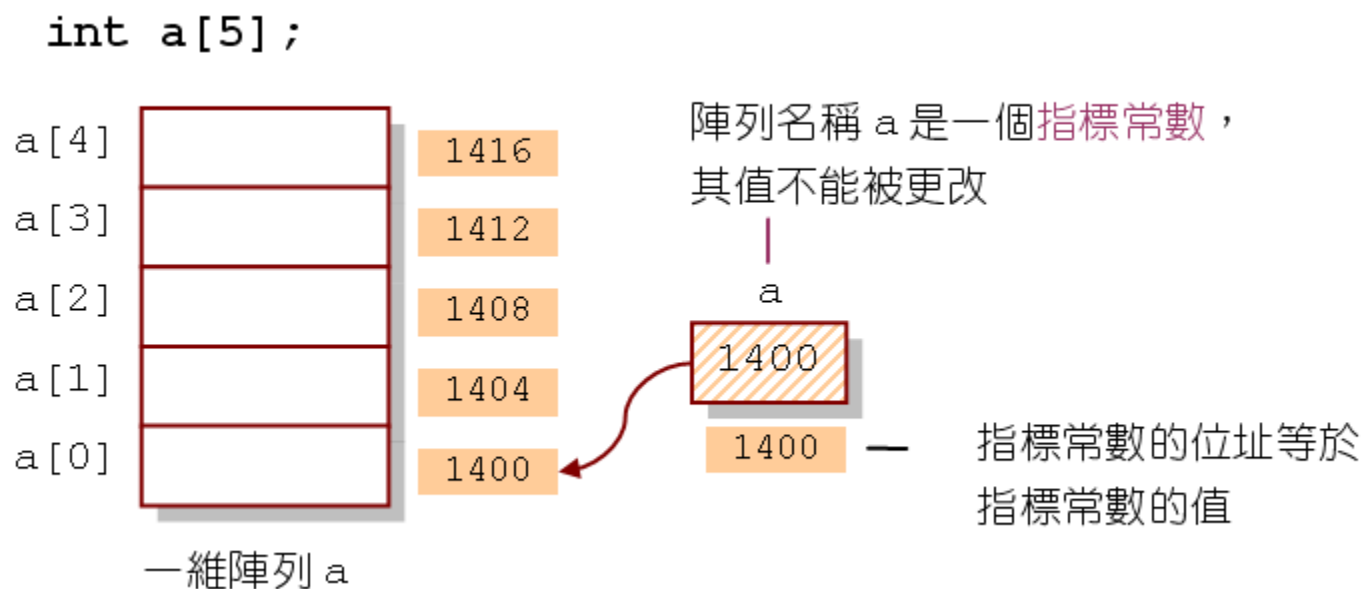
- 傳遞指標的應用：

```
#include <stdio.h>
#include <stdlib.h>
void add10(int *);          /* add10()函數的原型 */
int main(void) {
    int a=5;
    printf("Before call add10(),a=%d\n",a);
    add10(&a);              /* 呼叫add10()函數 */
    printf("After call add10(),a=%d\n",a);
    system("pause");
    return 0;
}

void add10(int *p1)
{
    *p1=*p1+10;
}
```

指標與陣列

- 陣列的名稱是一個**指標常數**，它指向該陣列的位址



陣列名稱的值即陣列的位址

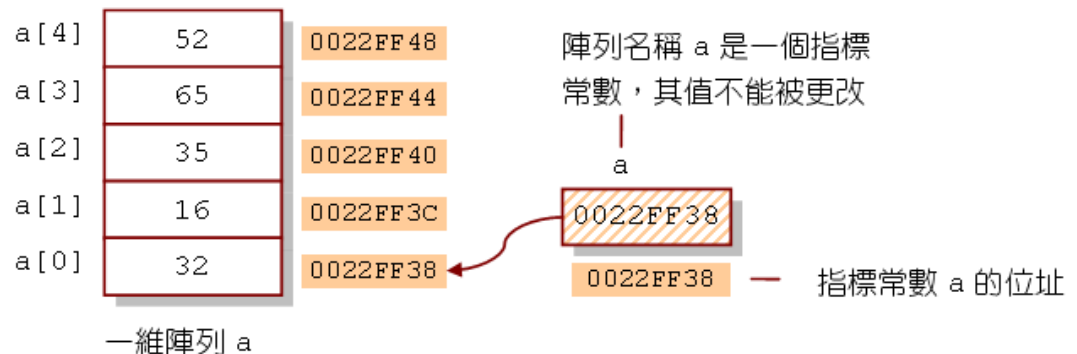
- 驗證陣列名稱是一個指標常數：

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int i,a[5]={32,16,35,65,52};
    printf("a=%p\n",a);
    printf("&a=%p\n",&a);
```

```
/* 印出指標常數a的值 */
/* 印出指標常數a的位址 */
```

```
for(i=0;i<5;i++)
    printf("&a[%d]=%p\n",i,&a[i]); /* 印出陣列a每一個元素的位址 */
```

```
system("pause");
return 0;
}
```



指標的算數運算

- 利用指標存取陣列的內容

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int a[3]={5,7,9};
    printf("a[0]=%d, *(a+0)=%d\n",a[0],*(a+0));
    printf("a[1]=%d, *(a+1)=%d\n",a[1],*(a+1));
    printf("a[2]=%d, *(a+2)=%d\n",a[2],*(a+2));

    system("pause");
    return 0;
}
```

指標的算數運算

- 利用指標計算一維陣列內所有元素的和

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    int a[3]={5,7,9};
    int i,sum=0;
    for(i=0;i<3;i++)
        sum+=*(a+i);    /* 加總陣列元素的總和 */
    printf("sum=%d\n",sum);

    system("pause");
    return 0;
}
```

-The End-