

程式設計 (**Programming**)

真理大學 資訊工程系 吳汶涓老師

CH10

結構、Unions、位元
處理以及列舉型別



本章綱要

10-1 簡介

10-2 結構定義

10-3 結構的初始值設定

10-4 存取結構的成員

10-5 使用結構與函式

10-6 Typedef

10-7 範例：高效率洗牌和發牌模擬器

10-8 Unions

10-9 位元運算子

~~10-10 位元欄位~~

10-11 列舉型別常數

10.9 位元運算子

- 所有資料在電腦內部皆以一連串的位元 (0/1) 表示

- 連續8 位元會構成一個位元組 (Byte)
- 各型別的記憶體位元量

int (4 位元組)	float (4 位元組)	char (1 位元組)
unsigned int (4 位元組)	double (8 位元組)	
short int (2 位元組)	long double (12 位元組)	

- 位元運算子

&		^	<<	>>	~
AND	OR	XOR	左移	右移	1 的補數

- & (AND) 運算子

Bit 1	Bit 2	Bit 1 & Bit 2
0	0	0
1	0	0
0	1	0
1	1	1

- | (OR) 運算子

Bit 1	Bit 2	Bit 1 Bit 2
0	0	0
1	0	1
0	1	1
1	1	1

- ^ (XOR) 運算子

Bit 1	Bit 2	Bit 1 ^ Bit 2
0	0	0
1	0	1
0	1	1
1	1	0

- 範例: 利用<<左移運算子，將一個十進位值轉成二進位值

```
Enter an unsigned integer: 65000
65000 = 00000000 00000000 11111101 11101000
```

```
3 #include <stdio.h>
5 void displayBits( unsigned value );
7 int main( void ) {
9     unsigned x;
11    printf( "Enter an unsigned integer: " );
12    scanf( "%u", &x );
14    displayBits( x );
16    return 0;
18 }
```

左移運算子會將它的左運算元向左移動數個位元，空白的位元以0填滿。

```
21 void displayBits( unsigned value )
22 {
23     unsigned c;
26     unsigned displayMask = 1 << 31;
28     printf( "%10u = ", value );
31     for ( c = 1; c <= 32; c++ ) {
32         putchar( value & displayMask ? '1' : '0' );
33         value <<= 1;
35         if ( c % 8 == 0 ) {
36             putchar( ' ' );
37         }
39     }
41     putchar( '\n' );
42 }
```

displayMask 是1
後面跟著31個0



常見的程式設計錯誤 10.10

將邏輯 AND 運算子 (&&) 與位元 AND 運算子 (&) 混用



常見的程式設計錯誤 10.11

將邏輯 OR 運算子 (||) 與位元 OR 運算子 (|) 混用，會造成錯誤

```
8  int main( void )
9  {
10     unsigned number1;
11     unsigned number2;
12     unsigned mask;
13     unsigned setBits;
16     number1 = 65535;
17     mask = 1;
22     displayBits( number1 & mask );

43     number1 = 21845;
47     displayBits( ~number1 );
49     return 0;
50 }
```



65535 = 00000000 00000000 11111111 11111111
1 = 00000000 00000000 00000000 00000001
using the bitwise AND operator & is
1 = 00000000 00000000 00000000 00000001



The one's complement of
21845 = 00000000 00000000 01010101 01010101
is
4294945450 = 11111111 11111111 10101010 10101010

練習

- 輸入兩個十進位，計算這兩個十進位的AND, OR, XOR的結果，並以二進位表示。
- 輸入某個十進位，計算這十進位左移、右移8 位元的結果，並以二進位表示。

```
displayBits( number1 >> 8 );
```

右移運算子會將它的左運算元向右移數個位元，空位充填方式依不同機器而異。

- 撰寫函式 `power2(number, pow)` 並使用左移運算子計算 $\text{number} * 2^{\text{pow}}$ 的結果，此函式需將結果以值與位元印出。

10.11 列舉型別常數 (enumeration)

■ 列舉為一種自定型別

- 使用識別字來代表整數常數
- 以enum起頭，enum內的值由0開始，逐漸遞增1

如：

```
enum months {JAN, FEB, MAR, APR, MAY};  
//enum months {JAN=1, FEB, MAR, APR, MAY};  
enum months M;  
  
for(M=JAN; M<=MAY; M++){  
    printf("%d\n", M);  
}
```

- 列舉型別常數的名稱最好只使用大寫字母


```

1  /* Fig. 10.18: fig10_18.c */
3  #include <stdio.h>
6  enum months {
7      JAN = 1, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC };
8
9  int main( void )
10 {
11     enum months month;
14     const char *monthName[] = { "", "January", "February", "March",
15         "April", "May", "June", "July", "August", "September", "October",
16         "November", "December" };
19     for ( month = JAN; month <= DEC; month++ ) {
20         printf( "%2d%11s\n", month, monthName[ month ] );
21     }
23     return 0;
24 }

```

列舉行別設定常數JAN為1，其後為2, 3, 4...

列舉常數類似符號常數，它們的值會在編譯時期被替換

```

1  January
2  February
3  March
4  April
5  May
6  June
7  July
8  August
9  September
10 October
11 November
12 December

```