

高級中學科學課程試用教材

電子計算機簡介

第四章 計算機內部資料的表示方法

從操作計算機的經驗中，我們知道計算機能執行命令、處理文字和數字等資料。各種資料如何儲存在計算機的內部呢？對這個問題的了解，將有助於日後學習程式語言、診斷程式錯誤、解釋計算機輸出的資料，進而更有效率地利用計算機。本章將逐一討論數字和文字資料的表示方法及其特性。

4-1 資料的儲存單位

計算機的記憶體可比做郵局裏的信箱，每一個信箱是一個記憶單位，每一個記憶單位都有編號，以便互相區別，稱為位址（Address）。記憶單位儲存資料的原理，與錄音機上的計數器或汽車的里程錶儲存數字的原理相似。計數器和里程錶是利用一組上面刻有阿拉伯數字的齒輪來表示數字的。圖 4-1 上面的四個齒輪表示 4816，下面的四個齒輪表示 4827。每一個齒輪的位置代表一個數位（digit），四個齒輪便能表示 0～9999 的所有數字，若要表示更大的數，必需用更多的齒輪。當我們將一組齒輪撥到 4816 時，我們便是將 4816 這個數字存入這一組齒輪。如果我們要將 4827 存入同一組齒輪，就將最右邊兩個齒輪各撥動一位。在存入 4827 的同時，原先儲存的 4816 便不見了，所以存入

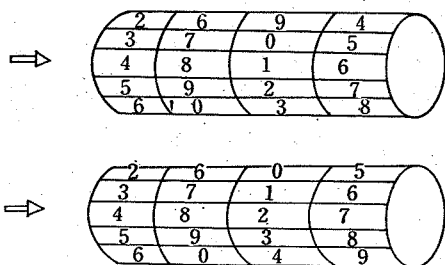


圖 4-1 齒輪計數器

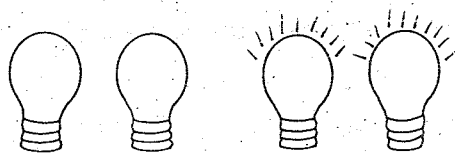


圖 4-2 表示二進位 0011 的一組燈泡

的動作有破壞性。我們知道後來儲存在齒輪組的數字是4827，是因為我們讀出這資料才知道的，讀出的動作不會改變儲存的資料，所以讀出的動作不具破壞性。

上面的齒輪記憶單位所存的是十進位數字。然而計算機的記憶體是電磁裝置，通常電磁裝置只有兩種狀態，例如電流通或不通，磁場指向某個方向或相反方向。如圖4-2所示，4個燈泡組成一儲存單位，電流通時燈泡亮，電流不通時燈泡不亮，每個燈泡可以有亮或不亮兩種狀態，四個燈泡在一起就有 $2^4 = 16$ 個不同的組合，若每一種組合表示一個數目，16種組合便可表示16個不同的數目。例如要表示由0至15的數目，可讓每個燈泡代表一個數位，亮表示“1”，不亮表示“0”，則16個組合如表4-1所示。

表4-1

狀 態	數 字	狀 態	數 字	狀 態	數 字	狀 態	數 字
0000	0	0100	4	1000	8	1100	12
0001	1	0101	5	1001	9	1101	13
0010	2	0110	6	1010	10	1110	14
0011	3	0111	7	1011	11	1111	15

這樣的數字表示法稱為二進位表示法，因為每一個數位只能有0和1兩種符號，二進位的數位又叫做位元（bit，是binary digit的簡稱）。

像十進位的表示法一樣，要表示較大的數字需要較多的位元，大多數計算機的記憶單位，由8個、16個或32個位元所組成。8個位元可稱為一個位元組（Byte）。計算機的記憶體容量都很大，通常都在數千、數萬、甚至數百萬個Byte，故習慣上以多少K Byte來表示其容量。在十進位中K表示千，但是在此處是指 2^{10} ，因為 $2^{10} = 1024$ ，接近1000。準此，一個16 K的電腦，其記憶體有 $16 \times 1024 = 16384$ Byte。

4-2 整數的表示法

計算機用二進位表示正整數的方法如前節所述，以一個Byte可表示的最大數字是1111 1111，即 $1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 2^8 - 1 = 255$ ，最小的數是0，記做0000 0000₂。若用2個Byte則可表示最大的數為1111 1111 1111 1111₂ = $2^{16} - 1 = 65535$ 。但是整數包含正數和負數，計算機常用第一個位元來表示數字的正負號，通常0表示正數，1表示負數，其餘的位元才用來表示數值。例如以兩個Byte來表示整數時，最大的數是0111 1111 1111 1111₂ = $2^{15} - 1 = +32767$ 。

【練習】 從鍵盤輸入

```
10    INPUT "NUMBER=" ; A %
20    PRINT A %
30    GOTO 10
RUN
```

螢光幕會出現

NUMBER=

立即輸入適當的數值，螢光幕會在下一行出現剛輸入的數值，然後又出現

NUMBER=

如果輸入的數值太大電腦就會有不同的反應。試著用最少次數輸入，猜測電腦用多少 Byte 儲存整數。

4-3 浮點表示法

科學上的數據有的很大，例如光速大約為 $30000000000 \text{ cm/sec}$ ，有的却很小，例如氫原子的半徑大約為 0.000000005 cm ，數值的範圍這麼大，用一般定點表示法很不方便。如果用指數來表示，不但寫起來省事，也可表示數值的有效數字。例如把光速寫成 $3 \times 10^{11} \text{ cm/sec}$ ，把氫原子半徑寫成 5×10^{-9} 或 $0.05 \times 10^{-7} \text{ cm}$ ，這樣的表示法是將數值分成兩部分，一為假數 (mantissa)，由有效數字組成，另一為指數；指數的作用在指示改寫成一般表示法時，應將假數的小數點向左或向右移動幾位，例如將 5×10^{-8} 寫成 0.000000005 時，要將 $.5$ 的小數點向左移動 8 位；改寫 $.05 \times 10^{-7}$ 時，則要將 $.05$ 的小數點向左移動 7 位，因此這種表示法稱為浮點表示法。

以浮點表示法，可將 26700 寫成 26.7×10^3 ，或 2.67×10^4 或 $.267 \times 10^5$ ，或 $.0267 \times 10^6$ 等形式，其中 $.267 \times 10^5$ 這種寫法稱為常態式。十進位的浮點常態式中，假數滿足

$$0.1 \leq \text{假數} < 1$$

也就是假數必須是純小數，且小數點後面第一位不能是 0。若以 8 個數位來表示浮點數字，可依圖 4-3 的格式以第一個數位表示數值的正負號，其後兩個數位表示指數，最後五個數位表示假數，小數點在指數與假數之間。

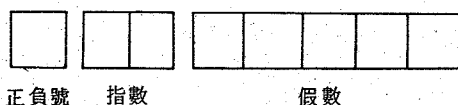


圖 4-3 十進制的 8 個數位表示浮點數的格式

指數的兩個數位，若只用來表示正整數，可以表示 $0 \sim 99$ ，但指數也可為負數，如用一個數位表示正負號，就僅能表示 -9 到 $+9$ ，範圍小而不實用；如果以 50 表示指數是 0 ，大於 50 的數表示正指數，小於 50 的數表示負指數，如此可存入此格式的指數範圍便擴大到 $-50 \sim +49$ 。下列是以這種方式存入浮點數字的例子

數	值	存 入 之 形 式								
30000000000	$= .30000 \times 10^{11}$	<table><tr><td>+</td><td>6</td><td>1</td><td>3</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	+	6	1	3	0	0	0	0
+	6	1	3	0	0	0	0			
.000000005	$= .50000 \times 10^{-8}$	<table><tr><td>+</td><td>4</td><td>2</td><td>5</td><td>0</td><td>0</td><td>0</td><td>0</td></tr></table>	+	4	2	5	0	0	0	0
+	4	2	5	0	0	0	0			
-12.48	$= -.1248 \times 10^2$	<table><tr><td>-</td><td>5</td><td>2</td><td>1</td><td>2</td><td>4</td><td>8</td><td>0</td></tr></table>	-	5	2	1	2	4	8	0
-	5	2	1	2	4	8	0			
-.0014762	$= -.14762 \times 10^{-2}$	<table><tr><td>-</td><td>4</td><td>8</td><td>1</td><td>4</td><td>7</td><td>6</td><td>2</td></tr></table>	-	4	8	1	4	7	6	2
-	4	8	1	4	7	6	2			

二進位的浮點數字表示法與此相似。不同的計算機所採用的位數常有不同，若用四個 Byte 來表示，且其格式如圖 4-4，以一個位元表示正負號，7 個位元表示指數，24 個位元表示假數。由於 7 個位元可表示 0 ~ 127 共 128 個正數，故可讓 64 表示指數 0，而使指數範圍在 -64 與 +63 之間，則可存入數值的範圍在 2^{-64} 到 2^{63} 之間，相當於 10^{-19} 到 10^{19} 的數值，而有效數字可達八位。

【練習】 由鍵盤輸入

10 INPUT "NUMBER=" , A

20 PRINT A

30 GOTO 10

RUN

仿照練習 1，輸入不同的實數，推測計算機用多少 bit 存指數，多少 bit 存假數，總共用多少 Byte 存浮點數字

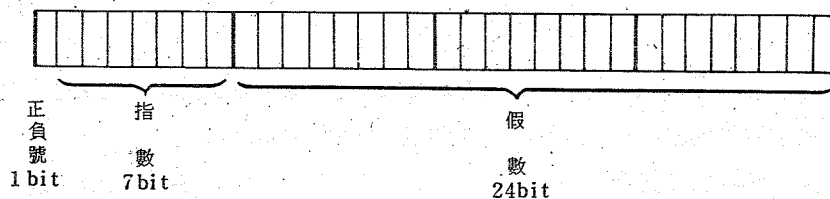


圖 4-4 以四個 Byte 表示浮點數的一種格式

4-4 數字系統之換算

由於計算機是二進位的機器，任何資料在計算機內部都是以二進位的數字系統來表示。但是我們已習慣於十進位的數字系統，對於二進位數字系統所表示的數值，常需換成十進位系統才容易接受，所以輸入輸出計算機的數值資料，大部份是採用十進位的數字系統，有時也採用十六進位數字系統。本節要介紹這些數字系統間的換算方法(十六進位數字系統每一數位可有 0 至 15 共 16 個數值，其符號為 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F)。

1. 整數換算

二進位或十六進位數字換算成十進位數字時，只要把各數位符號所代表的數值相加即得。

例 $11101_2 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
 $= 1 \times 16 + 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1$
 $= 29_{10}$

例 $30B6_{16} = 3 \times 16^3 + 0 \times 16^2 + 11 \times 16^1 + 6 \times 16^0$
 $= 3 \times 4096 + 0 \times 256 + 11 \times 16 + 6 \times 1$
 $= 12470_{10}$

十進位數字換算成二進位或十六進位時可用除法求得。

例 將 1455_{10} 換成十六進位

$$\begin{array}{r} 16 \overline{) 1455} \qquad \text{餘數} \\ 16 \overline{) 90} \cdots \cdots 15 = F \\ \quad 5 \cdots \cdots 10 = A \end{array}$$

$$\therefore 1455_{10} = 5AF_{16}$$

例 將 150_{10} 換算成二進位數字

$$\begin{array}{r} 2 \overline{) 150} \qquad \text{餘數} \\ 2 \overline{) 75} \cdots \cdots 0 \\ 2 \overline{) 37} \cdots \cdots 1 \\ 2 \overline{) 18} \cdots \cdots 1 \\ 2 \overline{) 9} \cdots \cdots 0 \\ 2 \overline{) 4} \cdots \cdots 1 \\ 2 \overline{) 2} \cdots \cdots 0 \\ \quad 1 \cdots \cdots 0 \end{array}$$

$$\therefore 150_{10} = 10010110_2$$

二進位與十六進位間的換算較為簡便，因為 $16 = 2^4$ ，一組四個二進位數位可表示 0 至 15 的數值，相當於十六進位系統的一個數位。二進位數字換成十六進位數字時，只需將二進位數字，從右至左，每四個數位直接換成一個十六進位，若最左邊一組不足四個數位，可用零補足，例如：

$$1011110110_2 = \underline{0010} \quad \underline{1111} \quad \underline{0110}_2 = 2F6_{16}$$

若要將一個十六進位數字換成二進位數字時，只需將十六進位數字的每個數位依序用四個二進位數位表示，例如：

$$2A5_{16} = \underline{0010} \quad \underline{1010} \quad \underline{0101}_2$$

2. 小數換算

二進位或十六進位數字換成十進位數字時，與整數換算相同，只將各數位符號所表示之數值相加即可。

$$\begin{aligned} \text{例 } .1111_2 &= 1 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} + 1 \times 2^{-4} \\ &= 0.5 + 0.25 + 0.125 + 0.0625 \\ &= 0.9375_{10} \end{aligned}$$

$$\begin{aligned} \text{例 } .AB_{16} &= 10 \times 16^{-1} + 11 \times 16^{-2} \\ &= 10 \times 0.0625 + 11 \times 0.00390625 \\ &= 0.66796875_{10} \end{aligned}$$

十進位數字換成二進位數字時，其方法以下例說明：

例 0.2 換算成二進位數字

假設二進位之形式為 $.A B C D \cdots$

$$\text{則 } \frac{1}{5} = \frac{A}{2} + \frac{B}{2^2} + \frac{C}{2^3} + \frac{D}{2^4} + \cdots$$

兩邊乘 2 得

$$\frac{2}{5} = A + \frac{B}{2} + \frac{C}{2^2} + \frac{D}{2^3} + \frac{E}{2^4} + \cdots$$

因為等號兩邊整數部分相等，分數部分相等

所以 $A = 0$

$$\frac{2}{5} = \frac{B}{2} + \frac{C}{2^2} + \frac{D}{2^3} + \frac{E}{2^4} + \cdots$$

將兩邊再乘 2 得

$$B = 0$$

$$\frac{4}{5} = \frac{C}{2} + \frac{D}{2^2} + \frac{E}{2^3} + \frac{F}{2^4} + \cdots$$

將兩邊再乘 2 得

$$\frac{8}{5} = 1 + \frac{3}{5} = C + \frac{D}{2} + \frac{E}{2^2} + \frac{F}{2^3} + \cdots$$

所以 $C = 1$

$$\frac{3}{5} = \frac{D}{2} + \frac{E}{2^2} + \frac{F}{2^3} + \cdots$$

將兩邊再乘 2 可得

$$D = 1$$

$$\frac{1}{5} = \frac{E}{2} + \frac{F}{2^2} + \frac{G}{2^3} + \cdots$$

如此繼續下去得

$$.2_{10} = .00110011 \cdots_2$$

注意： 0.2 換成二進位時為一無窮小數。

十進位數字換算十六進位數字可仿此進行。二進位數字與十六進位數字間之互換與整數換算相似，僅在分組時由左而右而已，例如：

$$.110101_2 = \underline{.1101} \quad \underline{0100}_2 = .D4_{16}$$

$$.333_{16} = \underline{.0011} \quad \underline{0011} \quad \underline{0011}_2 = .001100110011_2$$

4-5 文字符號資料

文字符號（簡稱字符）資料在計算機的內部也是以二進位數字來表示，只要讓每一個字符對應一個二進位數字（稱為字碼），計算機便可處理字符了。

常用的字符包括英文字母A~Z、阿拉伯數字0~9、空白、標點符號、運算符號、括號及少數幾個特別符號如%、\$等，共62個常用的字符。從4-2節知道每一個byte可表示256個數字，若以不同的數字表示不同的字符，一個byte便足夠表示全部的字符。

計算機處理文字資料時常需判斷字母順序。而字母順序可用字碼的大小來代表，那麼判斷字母順序的工作就變成比較數字大小的運算。字碼如果不作統一規定，使用起來一定很不方便。目前用得最普遍的一種字碼是“美國資訊交換標準碼”，英文全名是American Standard Code for Information Interchange，簡稱ASCII，共含128個字符。表4-2是ASCII碼的部分對照表，由空白符號的字碼開始（以0010 0000表示，化成十進制是32），其次依序為標點符號、阿拉伯數字、英文字母等。

文字資料例如單字、片語、句子，甚至一段文章，通常由許多字符所組成，且字符的個數（常稱做長度）常不相同，這一類不定長度的文字資料稱為字串。計算機儲存字串時，仍然是用一個byte儲存一個字符。

4-6 錯誤診斷

我們知道整數、浮點數、字符等在計算機內部都是以二進位數來表示。實際上，包括指令在內的任何資料都是以二進位數儲存在計算機內部。例如：四個位元組的狀態為

0100 0001 0100 0010 0100 0011 0100 0100

若表示整數，則為+1094861636，若表示字符則為“ABCD”，若表示浮點數（左邊第1個bit表正負號，第2至第8bit表指數，第9—16bit表假數），則為+ .51767777，顯然一個二進位數若解釋不同，所代表的意義便不相同，因此要計算機正確地處理資料，必須先決定資料的種類，讓計算機知道記憶單位中所代表的是何種資料。

當計算機預期接受整數資料時，如果輸入的資料是文字資料或浮點數，便發生錯誤，計算機能採取適當反應指出錯誤。我們可重做4-2的練習，以實數或英文字母輸入，觀察計算機的反應。

除了資料種類的錯誤，計算機也能診斷其他類型的錯誤，例如當計算機做算術運算時，若所得的數值太大或太小無法以既定數目的位元來表示，也會指出錯誤。數值太大稱為溢位（overflow），數值太小稱為虧位（underflow），其他類型的錯誤，將於學過程式語言時再加以討論。

4-7 數值的誤差

計算機內部以有限的位元來表示數值，基本上便有兩種限制，一是上節提到的溢位現象，另一是本節要討論的數值誤差。整數表示法所表示的數值是準確的，整數經過加、減、乘的運算也得到整數，只要不溢位，其結果也是準確的。但是整數相除結果可能不準確。計算機做整數除法時，其結果以整數表示法儲存，也就是只取整數部分。例如： $8/3=2$ $2/3=0$ 。

浮點表示法所表示的數值常常是不準確的，因為以有限的數位不能表示無限個數值，所以有許多實數不能表示出來。在十進制中也是同樣的情形，如果用五位數來表示實數，像2.3416，2.3417

表4-2 ASCII碼部分對照表

字 符	二進位數碼	十進位數碼	字 符	二進位數碼	十進位數碼
!	0010 0000	32	Q	0101 0001	81
"	0010 0001	33	R	0101 0010	82
#	0010 0010	34	S	0101 0011	83
\$	0010 0011	35	T	0101 0100	84
%	0010 0100	36	U	0101 0101	85
&	0010 0101	37	V	0101 0110	86
'	0010 0110	38	W	0101 0111	87
(	0010 0111	39	X	0101 1000	88
)	0010 1000	40	Y	0101 1001	89
*	0010 1001	41	Z	0101 1010	90
+	0010 1010	42	[	0101 1011	91
,	0010 1011	43	\	0101 1100	92
-	0010 1100	44	]	0101 1101	93
.	0010 1101	45	^	0101 1110	94
/	0010 1110	46	_	0101 1111	95
0	0010 1111	47	`	0110 0000	96
1	0011 0000	48	a	0110 0001	97
2	0011 0001	49	b	0110 0010	98
3	0011 0010	50	c	0110 0011	99
4	0011 0011	51	d	0110 0100	100
5	0011 0100	52	e	0110 0101	101
6	0011 0101	53	f	0110 0110	102
7	0011 0110	54	g	0110 0111	103
8	0011 0111	55	h	0110 1000	104
9	0011 1000	56	i	0110 1001	105
:	0011 1001	57	j	0110 1010	106
;	0011 1010	58	k	0110 1011	107
<	0011 1011	59	l	0110 1100	108
=	0011 1100	60	m	0110 1101	109
>	0011 1101	61	n	0110 1110	110
?	0011 1110	62	o	0110 1111	111
@	0011 1111	63	p	0111 0000	112
A	0100 0000	64	q	0111 0001	113
B	0100 0001	65	r	0111 0010	114
C	0100 0010	66	s	0111 0011	115
D	0100 0011	67	t	0111 0100	116
E	0100 0100	68	u	0111 0101	117
F	0100 0101	69	v	0111 0110	118
G	0100 0110	70	w	0111 0111	119
H	0100 0111	71	x	0111 1000	120
I	0100 1000	72	y	0111 1001	121
J	0100 1001	73	z	0111 1010	122
K	0100 1010	74	{	0111 1011	123
L	0100 1011	75		0111 1100	124
M	0100 1100	76	}	0111 1101	125
N	0100 1101	77	~	0111 1110	126
O	0100 1110	78			
P	0100 1111	79			
	0101 0000	80			

，2.3418 都能準確的表示出來，但是像 2.3417689 便無法表示，這時，我們可以用 2.3417 或 2.3418 來表示 2.3417689 的近似值。若用 2.3417，我們是把第五位以後的數位捨去，這樣產生的誤差稱為 Truncation error；若用 2.3418，我們是將第六位數字四捨五入，這樣產生的誤差稱為 Rounding error，計算機處理實數，通常都採用四捨五入 (Rounding) 的方式。既然浮點數可能是近似值，浮點數運算的結果也可能是近似值。

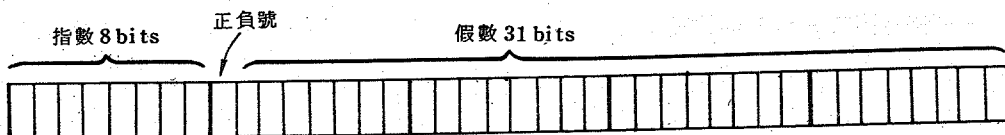
【練習】

下面的程式計算 500 個 0.1 的總和，將其輸入計算機執行，檢查結果若發現奇特現象請解釋之。

```
10 X1=.1
20 X=0
30 I=0
40 IF I=500 GOTO 80
50 X=X+X1
60 I=I+1
70 GOTO 40
80 PRINT X
```

第四章 問 題

1. 若計算機用兩個位元組存整數，則十進位數 1023 的表示法為何？
2. 將十六進位數 3F2 化成十進位數。
3. 依圖 4-4 之格式寫出下列數字之浮點常態式。
 - a. 127
 - b. $3/8$
 - c. $1/3$
4. 如果一部計算機用五個位元組儲存浮點數，其格式如下：



則可存的最大數值 (十進制) 為何？有效數字有幾位？

5. 有 1000 公斤米要裝成 10 袋，應如何裝法，當客人要買 1000 斤以內的任何數量時，都不必再稱。