

# 在微電腦DOS下隨機存取 檔案作業之介紹與探討

摘要 電算中心 陳寅龍

近幾年來，微電腦在國內掀起了一陣不小微電腦風，一時之間各種微電腦如小教授、小快樂、小神通…等充斥市面，可說是一種趕上時代的徵兆。然不少微電腦的使用者却把它當成是電動玩具來消遣、娛樂，以致於微電腦之許多功能都被忽略了。基於此，本文僅就微電腦中一項很重要的功能“隨機存取檔案作業方法”作一詳盡之介紹，以便使用者對微電腦有更深一層之了解。

隨機存取檔案作業方法在小型電腦以上可以說使用的相當的普遍及廣泛，然在微電腦之軟體系統下，以目前市面上的各種微電腦之書刊，很少涉及此。以致於少有人利用隨機作業方式來處理檔案，本文在此作多方面之介紹，希望有助於使用者利用此種方式處理各種檔案。

本文第一章介紹磁碟作業系統，在目前所處理的各種商業性的資料大部份都須使用到磁碟片，以便對資料作永久性的保留或更新，即使是使用磁碟片就必須認識磁碟是如何存取資料。第二章談及檔案在目前儲存資料之方法有很多種，但首先我們必須對檔案之定義有個相當之認識，才可進一步了解各種檔案結構在磁碟作業系統下是如何被運用的。

第三章談到隨機存取檔案作業方法，本章不但對該方法做了詳盡的定義，也舉了各種實例、程式來說明隨機存取作業是如何在磁碟作業下建立檔案、更新檔案、查訊…等，做了一個有系統之介紹。

筆者才疏學淺，加以倉促付梓，誤漏之處再所難免，尚祈先進不吝指正。

## 一、……什麼是DOS？

DOS表示“磁碟作業系統”。磁碟被裝在一個小的方形片子裡讓我們插入磁碟機裡。磁碟機是一個機械裝置用來讀和寫磁碟的。磁碟作業系統允許我們存放程式到磁碟裡，以這個簡單的能力作為理由就可以買個磁碟機。用磁碟系統我們就可以存放許多的程式到磁碟裡；而且又可依程式的名稱把程式取出來。應用DOS，我們可以要求電腦把磁碟裡每個程式

的名稱顯示出來，這就大大的改進了用卡式錄音帶來存程式，同時磁碟的速度大約也快了20倍，而且也比較可靠。

甚至於，我們可以很簡便的來儲存資料。在磁碟中儲存資料的能力使我們的電腦變成了更強的資料處理系統。現在我們可以用APPLE來處理成百成千個人的姓名地址資料列示出來。我們可以用一個程式來輸入統計資料，用一個或更多的程式來作不同的分析處理，而不必再各別的給每個程式輸入資料。

資料存在磁碟裡被看成一個“資料檔”(data file)。APPLE叫這些檔案為Text files。實際上，程式存在磁碟裡也是一個檔案，只是這些程式對BASIC來說比較特殊。

DOS實際上是BASIC的延伸。有一群BASIC的關鍵字(keywords)使我們能有效的應用磁碟。當我們應用磁碟作業系統開啓APPLE的時候，這些和磁碟有關的關鍵字就和我們所使用的BASIC相互合作。我們可以選擇使用BASIC；或是DOS的關鍵字。指令INT轉換到Integer BASIC，FP轉換到浮點式BASIC(或APPLESOFT BASIC)。這些都必須在系統裡都有所需要的語言。即使已經在使用APPLESOFT時也可以要求使用FP。這兩個指令都會清除掉現在記憶體裡所使用的程式，所以使用時要小心。

我們可以用DOS的關鍵字“CATALOG”來要求顯示磁碟內所包含的內容，電腦會有所反應。先開啓磁碟機，讀取磁碟目錄內所找到的程式和檔案名稱一個個的顯示出來。這個磁碟內的目錄被DOS一直維護著，沿著每一個程式名稱有一個I，A，B或T的符號分別表示檔案的內容是一個Integer BASIC程式，APPLESOFT程式，Binary(二進位)程式或Text)。二進位式程式是一個不必透過BASIC而直接可以被執行的程式，而註記Text的是資料檔案，是本章所要提到的主題。還有就是顯示上的星號(\*)表示那個檔案被鎖起來。被鎖起來的檔案就不能直接被寫入或從磁碟中清除。如果顯示幕已顯示滿了，DOS就會等到鍵盤上任意打入一個字鍵後，再繼續顯示未顯示完的程式名稱。如果最後的程式名稱都已顯示出來，我們就可以看到(>)或(])符號表示BASIC已經準備接受接下來的指令。

我們可以簡單的鍵入“SAVE FIRST”就可以把程式存放在磁碟片裡，定名為FIRST。磁碟就會轉幾秒鐘，當它停下來，指示燈也熄滅的時候，鍵入“CATALOG”，來確定程式FIRST已經在磁碟片裡。如果我們原先已經有一個程式的名稱叫FIRST存在磁碟片內，則將被存放新的程式，而無法再度取得原先舊版的程式。

當一個程式已經存放在磁碟片裡，我們可以用兩個指令來應用這個程式。“LOAD FIRST”將會把FIRST程式從磁碟片中傳送到APPLE的記憶體裡。如果要執行這個程式，接下來就是鍵入“RUN”。我們也可以用“RUN FIRST”直接的執行磁碟片中的FIRST程式。

如果這個程式，我們不再須要使用時則可用“DELETE FIRST”來把它從磁碟片中清除，而且最好是想成清除掉的程式不可能再取得到。

另外還有一群BASIC關鍵字是我們用來處理磁碟片內的資料用的。這些都是被用在程式中，用來建立檔案，寫入資料，讀取資料或清除檔案用的。本章所要討論的就是這些關鍵字的內容和使用方法。

## 二、……什麼是檔案？

一個檔案就是磁碟片內用來存放資料的一塊區域，如前所述，我們可以存放程式在裡面。DOS幫我們作每件事，讓我們把程式存放在磁碟裡。但是當我們存放資料到資料檔案中，就必須完全靠自己來組織這些資料。

有關資料檔案之另外增加幾分神秘性的原因，在於它是不可見的，就和程式在執行時一樣。我們可以發現在程式中我們在做很多變化的事情，但一直在最後得到列印出來的結果之前，我們根本看不見什麼。我們將要大大的擴展我們的能力用程式來建立檔案和存取資料。我們可以“LIST”指令來列示出程式。但是對我們自己建立的資料檔而言，我們必須自己寫程式來列示資料檔。

資料可以被組織成有序性的格式，或被安排成隨機性存取方式（Random Access），一些指令和技巧對有序性和隨機性檔案處理方式是相同的。最重要的一點是必須記得所有程式和檔案之溝通是透過BASIC的PRINT語法來控制的。值得重複說明：“所有程式和檔案的溝通是透過BASIC的PRINT指令加以控制的”。其次必須知道的部份是所有檔案控制指令必須有CTRL-D先引，絕對不可忘記。否則，PRINT指令將顯示到螢幕上。“所有檔案的控制指令必須有CTRL-D先引”。下面將說明隨機性檔案是如何工作的。

## 三、……隨機存取檔案

既然檔案中的資料段可能變化長度，因此它們也就佔用不同的空間，因此就沒有辦法來

預測第 5 或第 50 個資料段在檔案中開始的地方。所以在有序性檔案中我們要經常從頭開始來讀取檔案。當我們要再寫資料到這種檔案中，最保險的方式是重新再把整個檔案寫一次。如果當檔案變得愈來愈大那麼也就要花費愈多的時間了。

而用隨機存取檔案就改觀了。這種新的檔案結構使得讀取第 25 個資料段（任意）來更改，再重新寫入檔案中，而不會冒什麼風險可能破壞其他的資料段。這是因為預先已訂好固定長度的空間來存放每個資料段。這表示在許多的應用中，會存放許多沒用的空白存放在檔案中。

我們可用隨機存取檔案應用於各種資料錄。對於我們不是每次都是處理每一筆資料的應用上，能夠取得任意一筆資料的能力是相當理想的。相對的在地名接龍遊戲的程式，很顯然的我們應用程式時都必須牽涉整個檔案就不能應用了。一般隨機存取檔案被應用於名字和住址的列表上，財務會計系統和票據管理上。還有家庭資料管理，雜誌文章的參考資料都適合這種檔案的應用。

許多應用上，把數字檔案連繫起來組成系統檔案。一個定單號碼可能被連線到郵寄地址檔或存貨檔等。

在有序性檔案中儲存的基本單位是字或位元組（Byte）。而在隨機存取檔案的基本儲存單位是一個資料錄，而錄是由 Byte 位元組合成的。如果 20 個位元組足夠我們儲存所需的資料段，那麼我們就可以把檔案用 20 個位元組的錄來組成。錄的長度完全視我們自己需要而定。我們由應用的需要來決定錄的長度，最重要的是對應用上能完全而且有效的組成檔案而能管理所需要的資料加以研究。

經常，有一群程式是被用來處理一個檔案或是一群檔案（系統檔案）——一個程式用來建檔或去掉不要的資料段，另一個用來編輯和檢查資料之正確與否，可能還有第 3 個程式用來列印一份整個檔案資料美觀的報表。

OPEN, READ, WRITE 指令應用在隨機存取檔案是對應於有序性檔案上的應用加以擴展。OPEN 帶有一個“L”值，用來設定每個資料錄位元組的長度。READ 和 WRITE 帶有一個“R”值，用來指定要讀／寫第幾個資料錄。接下來我們將更詳細的加以說明。

讓我們發展一個應用電腦的姓名—地址檔，經常在商業上或個人都會使用。這個概念是把姓名—地址存放在磁碟檔案中。接著從這個檔引伸出特定狀況下我們所需要的。姓名被一個碼來分類：用 H, W, C 來區分家族的寄信地址是丈夫、妻子或兒女。商用上用 B, S 表

示票據地址和運送地址等。

在一般商用上這些姓名可能被安排按字母或郵遞區號順序或生意大小來排列，爲了達到這個目的我們並不需要重新安排姓名檔。但是我們必須建立另外一個檔只包括所需順序錄的順序列，而我們須根據資料錄的順序號，來維護數個這種順序列，因此我們就可以寫一個程式依照這種順序，找出對應的錄順序號存取我們所需順序的資料錄。

讓我們用程式來組織信件處理的姓名—地址檔案，其中包括了好多項工作要做。程式的一部份用來從鍵盤輸入資料，另一部份把資料段寫入檔案中，另外的一部份就必須用來決定新的資料段屬於那裡。

我們必須自己組織各種資料段，以決定要包括那些資料項，每個資料項有多少個字，後來決定資料錄的長度，同時程式必須有一些碼來管理這些事項。我們須要一段程式來把資料段寫到資料檔中。可能在寫程式的工作裡決定在檔案中如何組織這些資料段是最重要的一部份了。

當我們坐下來打入第一個姓名—地址的時候，我們知道檔案是空的，到後來我們就不知道檔案到底存有多少個姓名—地址，因此我們就不曉得下個姓名—地址應放在檔案的第幾個資料段了。所以我們必須把姓名—地址也寫在紙上。整個的觀念是讓電腦來作這些工作，因此我們必須發展的程式要讓電腦保持現在作的狀況，一種方法是指定一個資料錄號包括在資料段裡，用來區別是第九個資料段，例如第一個錄編號 1，第二個錄編號 2 等。現在我們須知道下一個是多少要被存放在檔案中了，最好的方法是把這個號碼存在第 0 個錄裡，很幸運的錄的編號是從 0 開始，所以，如果一個檔案沒有放姓名—地址在裡頭的話，第 0 個錄放 1，我們就能很簡單的寫個程式來作這個啓始的工作。

後來當一個個姓名地址加入檔案中時，我們就把第 0 個錄的值每次加 1。接下來，我就要想想如何從檔案中刪除，即使是寫一個建檔的程式，因爲你可能會鍵錯資料。

要從檔案中刪除信件姓名—地址的資料有很多方法可以做到。我們可以用“DELETED”來取代姓名，或是把最近刪除的錄立刻再被用來加入新錄，我們可以把已刪除的資料錄的可用空間建立一個目錄存在檔案本身，作這些目錄我們把它視爲是資料段的一項資料就和姓名、地址和錄號一樣。所以我們把前一個刪除的資料錄號存在現在刪除的這個錄裡，而把現在刪除的資料錄號存在第 0 個錄裡頭，接著檔案最大能存放錄號的後面，這樣會把最後刪除的錄編號存放在第 0 個錄中。現在，我們有兩個數目存在第 0 個錄：檔案結束後的下一個錄

號和最近刪除的錄號。當我們建檔的時候，最近刪除的錄號是 $\emptyset$ 。

這種方法提供給我們決定資料錄是否已經被刪除。讀取資料，若資料錄區別碼和資料錄號相等，表示它是真正的資料，若不相等則表示資料段已經被刪除。而其資料錄號表示先前一個被刪除的錄。一個檔案包括一些被刪除的錄的例子。看圖 8 - 2。

$\emptyset$	9 {on the end}, 8 {last deleted entry}	
1	1 JONES JOHN . . .	
2	2 SMITH WILLIAM . . .	
3	3 HAYES MARY . . .	
4	6 {deleted entry} . . .	
5	5 BRADSHAW ELEANORE . . .	
6	$\emptyset$ {deleted entry (first one)} . . .	
7	7 HOUGH HUGH . . .	
8	4 {deleted entry} . . .	
9	{never used}	

圖 8 - 2 有用和被刪除錄的圖示

讓我們對圖 8 - 2 的有效空間加以追蹤。第 $\emptyset$ 個錄的第 2 個數目為 8，看第 8 個錄，找到 4，看第 4 個錄，找到 6，看第 6 個錄，找到 $\emptyset$ ，所以被刪除的錄是 8，4，6，若我們要重新使用第 6 個錄，程式就要把在第 $\emptyset$ 個錄現在存 8 的值存放 $\emptyset$ 。（重新使用已刪除的錄要把最後刪除的錄最先使用）。打入資料的程式必須檢查第 $\emptyset$ 個錄存放的兩個錄號來決定新打入的資料應放在檔案內的最後一個錄之後，或是加在已經被刪除的錄的位置。換言之，若刪除之錄號是 $\emptyset$ ，就加在檔案後面，否則就用已刪除的錄。

雖然我們只是建檔，也必須完全瞭解這些是非常重要的，設計程式之先，要把整個系統設計好。

我們已經能夠在控制下美好的打入或刪除資料。現在是要如何來更新資料錄呢？當每個姓名—地址已經有了區別號碼以後，我們就能很容易的根據這個號碼讀取對應的資料錄，同時顯示出來檢視而加以更正再寫回原來的地方。我們因此必須週期性的印出檔案的姓名—地址錄和區別號碼。那也是相當地簡單寫個程式把整個檔案從頭至尾把沒刪除的資料列印出來。程式也能根據有須不同種類順序的順序檔，列印出不同順序的資料。

到現在看起來，我們已經把姓名—地址檔案想成已有了4種功能：新增、刪除、更正和列印。我們必須注意到必須先在打入資料之前做起始的準備。看程式8—8。

```

90 REM * INITIALIZE MAILING LIST FILE
95 D$=CHR$(4)
100 PRINT D$;"OPEN FIRST FILE"
110 PRINT D$;"WRITE FIRST FILE"
120 PRINT 1
122 PRINT 0
130 PRINT D$;"CLOSE FIRST FILE"

```

程式8—8 起始準備姓名—地址檔

當這個程式被執行以後，則在第一個錄將含有兩個值1和0。當然我們必須確定這個程式不會再重新被執行，像這種重新組織檔案的工作，最好留給特定的人來做。

現在，讓我們設計一個資料錄的格式，看表8—1。

表8—1 信件處理姓名—地址資料錄格式

DATA ITEM	LABEL	MAXIMUM # OF CHARACTERS
Identification #	ID #	4 + 1
Code	CODE	5 + 1
Last name	LAST	20 + 1
First name	FRST	20 + 1
Address	ADDR	30 + 1
City	CITY	16 + 1
State	STAT	2 + 1
Zip	ZIP	5 + 1
Telephone	PHON	17 + 1
		119 + 9 = 128

在表8—1我們允許每個資料項後加上一個CR碼。電話號碼項保留17位，允許加上國際碼、區域碼或電話號碼之擴展等，總共變成128個字長。

如果我們小心的把上面所述考慮的事項列下來，我們就可以組成一個信件處理地名—地址檔的應用程式，每個控制程式段的副程式考慮作一個工作。如表8—3所示之程式功能。

表8—3 信件處理姓名—地址檔案程式的功能

1. 讀取資料的標記。
2. 讀取能存放資料有效空間的參數。
3. 顯示能存放資料區別號 ( ID ) 和準備讀取鍵入的資料。
4. 當鍵入資料為空串時，程式結束。
5. 準備有效空間來存放 ( OPEN 檔案 )。
6. 寫入新的資料段。
7. 在第0個檔案錄中寫入下一個有效空間的註記。
8. 回頭再作。

表 8-3 中的 8 項中，其中 6 項適合寫成副程式。這些副程式也適合在這個系統的其他程式中加以應用。第 4 項當第 3 項的副程式在輸入空串的姓名—地址資料時，送回一個訊號 (E1) 來結束程式，第 8 項只是直接的跳回第 3 項回頭重覆再作同樣的工作。

我們任意選取行號來作副程式而用在控制程式裡，很簡單的我們就能完成控制程式，如程式 8-9 a。

```

200 GOSUB 1000 : REM * READ DATA LABELS
210 GOSUB 900 : REM * READ AVAILABLE SPACE PARAMETERS
220 GOSUB 800 : REM * DISPLAY NEXT AVAILABLE ID AND
    REQUEST DATA
230 IF E1 = 1 THEN END : REM * TERMINATE ON NULL
    LAST NAME
240 GOSUB 700 : REM * PREPARE AVAILABLE SPACE
250 GOSUB 600 : REM * WRITE NEW ENTRY
260 GOSUB 500 : REM * WRITE AVAILABLE SPACE INFO BACK
    TO RECORD ZERO
270 GOTO 220 : REM * DO IT AGAIN

```

#### 程式 8-9a 信件處理姓名—地址系統的控制程式

在程式 8-9a 主程式中包括用 6 個副程式和 2 個控制指述。在第 230 行 E1 值如果操作員要結束程式執行必須被完成 1，否則第 220 行的副程式所讀取的任何資料將被寫入檔案中。第 270 行簡單的用 GOTO 指令來重覆執行以便讀取下一個鍵入的資料。接下來我們來寫一個個的副程式。

讀取資料標記的副程式從 1000 開始，如果我們多想一想如何設計一段程式從鍵盤上讀取資料，我們會有一個創新的方法。當我們可以用 8 個 INPUT 來詢問讀取資料。8 個 INPUT 每個後面再問是否鍵入資料太長，因此每次檔案的設計程式都要重寫一次，這不是一個好的想法。為何不直接顯示要輸入什麼資料，而且告訴操作員最長鍵入多少個字呢？因此最主要的改變是用 DATA 指令作個小小的修改。我們的 DATA 指述是從表 8-1 直接得來的，包括標記和長度限制。我們可以用 FOR... NEXT 把 DATA 讀入陣列中，看程式 8-9

```

b.
998 REM * READ DATA LABELS AND LIMITS
1000 READ N0
1010 FOR X9 = 1 TO N0
1020 READ LA$(X9),LE(X9)
1030 NEXT X9
1090 RETURN
1996 :
1998 REM * DATA LABEL & LIMITS
2000 DATA 9
2005 DATA ID #, 4
2010 DATA CODE, 5
2015 DATA LAST, 20
2020 DATA FRST, 20
2025 DATA ADDR, 30
2030 DATA CITY, 16
2035 DATA STAT, 2
2040 DATA "ZIP ", 5
2045 DATA PHON, 17

```

#### 程式 8-9b 信件處理姓名—地址讀取資料標記的程式



在程式 8-9b 中，NØ 是每個資料錄所含資料項之項數，而資料標記存在 LA\$ 之陣列中，資料長度限制數存在 LE 陣列中，完整的程式必須包括適當 DIM 指述。

讀取能存放資料有效空間參數的程式非常簡單，只是做啓始程式的相反動作而已，我們要讀取兩個有效空間的變數值而已。如程式 8-9c。

```

898 REM * READ AVAILABLE SPACE
900 PRINT DS; "OPEN"; FS
910 PRINT DS; "READ"; FS
920 INPUT NS
930 INPUT DS
940 PRINT DS; "CLOSE"; FS
990 RETURN

```

#### 程式 8-9c 讀取存放資料有效空間參數的程式

在程式 8-9c 中我們把新的空間變數存在 NS，而已被刪除之有效空間值在 DS。必須注意完整的程式要把 CTRL-D 存在 D\$。

現在是顯示下一個能被存放的 ID 和要求讀取資料，我們從 800 開始，這個工作直接作，但須小心的計劃，第一個工作是決定下一個真正能存放資料的有效空間，我們先選取未被存放的空間 (NS) 為新的有效空間，但是若已經有被刪除的空間 (DS)，我們先把 DS 指定為 ID。我們用 FOR-NEXT 迴路來顯示資料標記和要求鍵入資料。看程式 8-9d。

```

798 REM * PROCESS DATA ENTRY FROM KEYBOARD
800 ID = NS : IF DS < > Ø THEN ID = DS
803 PRINT
805 PRINT LA$(1); " : "; ID
810 DA$(1) = STR$(ID)
815 FOR I9 = 2 TO NØ
820 PRINT LA$(I9); "? ";
825 INPUT DA$(I9)
* 830 IF I9 = 3 THEN IF LEN (DA$(3)) = Ø THEN E1 = 1 :
      GOTO 890
835 IF LEN (DA$(I9)) < = LE(I9) THEN 845
840 PRINT "TOO LONG" : PRINT " : "; : GOTO 825
845 NEXT I9
850 E1 = Ø
890 RETURN

```

#### 程式 8-9d 處理從鍵盤輸入資料的程式

注意行 830，我們要求的 LAST 名字是長度為 Ø (空串) 的話，把 E1 定成 1。這對操作員來說很容易，只要按下 RETURN 鍵就好了。我們要包括 DA\$( ) 陣列用 DIM 指令來完成整個程式。

接下來，我們要準備有效空間來存放資料，在這裡我們須要知道資料是要被存放在新的錄中或是存在已先前刪除的錄中，如果是存放在新的錄中，我們只要把新空間變數 (NS)

加入 1 就好了。如果要存放在先前被刪除的錄中，那麼我們就當刪除發生的時候先要讀取原先被寫上的資料錄編號。維護有效空間串列是非常重要的。可記得圖 8 - 2 的這些？

現在我們必須注意到資料段相關的長度了。當我們 OPEN 一個隨機存取檔案，我們必須告訴 BASIC 檔案的錄的長度，這由 OPEN 指令的 " L " 值來訂，如

```
100 PRINT D$;"OPEN FIRST FILE,L128"
```

就會做 OPEN 的工作；同時把資料長度定成 128。我們可以用數字變數來存放資料長度，用字串變數來存放檔案名稱，如下：

```
200 PRINT D$;"OPEN";F$;" ,L";L0
```

當我們讀和寫時，對隨機存取檔案我們必須定 R 值來設定資料錄要被讀和寫的錄編號。

```
PRINT D$;"WRITE";F$;" ,R";R1
```

將會使下一個 PRINT 指令指定 F\$ 檔的 R1 個錄。

```
698 REM * IF THIS ENTRY REPLACES DELETED DATA MAKE
      PREPARATIONS
700 IF DS = 0 THEN 760
* 710 PRINT D$;"OPEN";F$;" ,L";L0
* 720 PRINT D$;"READ";F$;" ,R";DS
730 INPUT DS
740 PRINT D$;"CLOSE";F$
750 GOTO 790
760 NS = NS + 1
790 RETURN
```

#### 程式 8 - 9e 準備有效空間的程式

在程式 8 - 9e 中行 710 和 720 我們用了 " L " 和 " R " 的設定，在這個副程式中只有當新的空間或刪除空間的時候其中之一情況發生，而不會兩種情況同時發生。

接下來就是我們要正確的把資料寫入檔案中，這是一個短短的副程式，而且一點也不複雜。

```
598 REM * WRITE ENTRY
600 PRINT D$;"OPEN";F$;" ,L";L0
610 PRINT D$;"WRITE";F$;" ,R";ID
620 FOR I9 = 1 TO N0
630 PRINT DA$(I9)
640 NEXT I9
650 PRINT D$;"CLOSE";F$
690 RETURN
```

#### 程式 8 - 9f 把資料寫入檔案中的程式

毫無疑問的，最後我們要把有效空間串列的參數寫回到檔案的第 0 個錄。這和啓始程式差不多，只是除了要寫上 NS 和 D\$。如程式 8 - 9g。

```

498 REM * WRITE AVAILABLE SPACE DATA
500 PRINT D$;"OPEN";F$
510 PRINT D$;"WRITE";F$
520 PRINT NS
525 PRINT DS
530 PRINT D$;"CLOSE";F$
590 RETURN

```

程式 8-9g 把有效空間參數寫回檔案第  $\theta$  個錄的程式

最後，爲了完成整個程式，記得加上 CTRL-D 在 D\$，128 在 L $\theta$ ，檔案名稱在 F\$，和適當的 DIM 之數值的指述。

```

9 REM * ID => ENTRY IDENTIFICATION NUMBER
10 REM * NS => NEW SPACE
11 REM * DS => DELETED SPACE
30 D$ = CHR$(4)
50 L $\theta$  = 128
60 F$ = " FIRST FILE"
70 DIM LA$(9),LE(9),DA$(9)

```

程式 8-9h 程式中的一些參數

在程式 8-9h 中，行 30 把 CTRL-D 存在 D\$，使得檔案存取的指令 PRINT 看得見。很有趣的先不加入行 30，我們可以先測試程式而不須寫入檔案中。把 D\$ 當成一個空串，所有檔案的存取就轉變成鍵盤和螢光幕了。行 50 把資料錄的長度存放在變數 L $\theta$  中，在應用上使得改變資料錄的長度變得非常容易。行 60 把檔案名稱存放在 F\$ 中，同樣的，改變檔案名稱存取另外的檔案也變得簡單了。

```

9 REM * ID => ENTRY IDENTIFICATION NUMBER
10 REM * NS => NEW SPACE
11 REM * DS => DELETED SPACE
30 D$ = CHR$(4)
50 L $\theta$  = 128
60 F$ = " FIRST FILE"
70 DIM LA$(9),LE(9),DA$(9)
196 :
198 REM * ENTER NAMES AND ADDRESSES IN A MAILING LIST
FILE
200 GOSUB 1000 : REM * READ DATA LABELS
210 GOSUB 900 : REM * READ AVAILABLE SPACE PARAMETERS
220 GOSUB 800 : REM * DISPLAY NEXT AVAILABLE ID AND
REQUEST DATA
230 IF E1 = 1 THEN END : REM * TERMINATE ON NULL LAST
NAME
240 GOSUB 700 : REM * PREPARE AVAILABLE SPACE
250 GOSUB 600 : REM * WRITE NEW ENTRY
260 GOSUB 500 : REM * WRITE AVAILABLE SPACE INFO BACK
TO RECORD ZERO
270 GOTO 220 : REM * DO IT AGAIN
496 :
498 REM * WRITE AVAILABLE SPACE DATA
500 PRINT D$;"OPEN";F$
510 PRINT D$;"WRITE";F$
520 PRINT NS
525 PRINT DS
530 PRINT D$;"CLOSE";F$
590 RETURN

```

```

596 :
598 REM * WRITE ENTRY
600 PRINT D$;"OPEN";F$;"L";L0
610 PRINT D$;"WRITE";F$;"R";ID
620 FOR I9 = 1 TO N0
630 PRINT DA$(I9)
640 NEXT I9
650 PRINT D$;"CLOSE";F$
690 RETURN
696 :
698 REM * IF THIS ENTRY REPLACES DELETED DATA MAKE
PREPARATIONS
700 IF DS = 0 THEN 760
710 PRINT D$;"OPEN";F$;"L";L0
720 PRINT D$;"READ";F$;"R";DS
730 INPUT DS
740 PRINT D$;"CLOSE";F$
750 GOTO 790
760 NS = NS + 1
790 RETURN
796 :
798 REM * PROCESS DATA ENTRY FROM KEYBOARD
800 ID = NS : IF DS < > 0 THEN ID = DS
803 PRINT
805 PRINT LA$(1);": ";ID
810 DA$(1) = STR$(ID)
815 FOR I9 = 2 TO N0
820 PRINT LA$(I9);"? ";
825 INPUT DA$(I9)
830 IF I9 = 3 THEN IF LEN(DA$(3)) = 0 THEN E1 = 1 : GOTO
890
835 IF LEN(DA$(I9)) < = LE(I9) THEN 845
840 PRINT "TOO LONG" : PRINT " : "; : GOTO 825
845 NEXT I9
850 E1 = 0
890 RETURN
896 :
898 REM * READ AVAILABLE SPACE
900 PRINT D$;"OPEN";F$
910 PRINT D$;"READ";F$
920 INPUT NS
930 INPUT DS
940 PRINT D$;"CLOSE";F$
990 RETURN
996 :
998 REM * READ DATA LABELS AND LIMITS
1000 READ N0
1010 FOR X9 = 1 TO N0
1020 READ LA$(X9),LE(X9)
1030 NEXT X9
1090 RETURN
1996 :
1998 REM * DATA LABEL & LIMITS
2000 DATA 9
2005 DATA ID #, 4
2010 DATA CODE, 5
2015 DATA LAST, 20
2020 DATA FRST, 20
2025 DATA ADDR, 30
2030 DATA CITY, 16
2035 DATA STAT, 2
2040 DATA "ZIP ", 5
2045 DATA PHON, 17

```

程式 8 - 9 信件處理建檔輸入資料的完整程式

這個程式是準備作建立姓名—地址檔案而能用的例子。應用前面所敘述的討論和一些程式段，你就同樣的能發展、能夠刪除資料、更改資料、和列印資料的其他程式了。

這個程式有些地方可以改成更具彈性，我們可以把錄的長度值存放在已經存放有效空間

參數的第  $N$  個錄裡。我們可以由操作員來指定存取那一個資料檔，我們可以把 DATA 的指述去掉而另外用檔案來代替，用這些方法有很大的好處，有關檔案的資料都存在檔案本身，使得程式可以處理許多其他不同的信件檔案。我們可以處理不同數量資料錄、不同長度的資料錄、和不同的資料標記而共用一個程式。很快的我們可以發現寫程式做資料處理都離不開這些有效的資訊。由於檔案中資料的改變，只須花小小的代價，就可以改變程式的功能，使得程式能做到資料庫的管理能力。

我們可以分開另外寫刪除或更正資料的程式，或在 8-9 中加入其他新的所需的副程式，因此我們也就可以用“菜單”的方式讓使用者來選擇所需要的功能了。

#### 四、結 論

綜合以上所談，我們可以很明顯的探究出隨機存取檔案之作業方式及其優點，然資料量在何種情況下採取此等作業方式較為適宜呢？依筆者以往之作業經驗：主檔之資料量與異動檔之資料量之比例在十比一之下，以此等作業方式其效率最高否則應以順序存取較優，故隨機存取之作業方式通常皆應用於工商各界，諸如航空訂位系統、查訊作業系統、庫存管理、物料管理……等。

事實上，檔案結構尚有三種方法，因篇幅之限制未能詳盡說明，只以表列式作個扼要之介紹（如附表）。

討 論 要 點	順 序	直 接 儲 取	索 引 順 序	串 列 結 構
1. 檔案中錄的排列	順 序	任 意	順 序	任 意
2. 儲取一個錄的正常方法	從檔案的起始開始順序搜尋	經由表或公式作直接儲取	直 接 或 順 序 法	利用指標順序搜尋
3. 更新資料的方法	重新組織檔案	插 入 一 個 錄	儲存在超溢區域，視情況而做重組的工作	利 用 指 標
4. 優 點	<ul style="list-style-type: none"> <li>• 簡易</li> <li>• 可以儲存於不昂貴的磁帶上</li> <li>• 耗費少量的儲存空間</li> </ul>	<ul style="list-style-type: none"> <li>• 快速儲取任何的錄</li> <li>• 可線上完成快速而簡易的更新</li> </ul>	<ul style="list-style-type: none"> <li>• 可以順序或直接地儲取</li> </ul>	<ul style="list-style-type: none"> <li>• 即使沒有順序儲存亦可順序儲取</li> <li>• 容易完成錄的刪除與添加</li> </ul>

5. 缺點	<ul style="list-style-type: none"> <li>• 順序搜尋將花費大量的時間</li> <li>• 異動資料必須整批處理且須加以排列</li> <li>• 如果使用不同的鍵欄，檔案必須重新分類</li> </ul>	<ul style="list-style-type: none"> <li>• 不能使用磁帶</li> <li>• 比順序檔需要更多的空間</li> <li>• 並不適於須存取大部份的錄</li> </ul>	<ul style="list-style-type: none"> <li>• 經常需要重新組織</li> <li>• 錄的刪除，超溢區域與索引表會消耗儲存空間</li> <li>• 需要複雜的軟體</li> </ul>	<ul style="list-style-type: none"> <li>• 爲了保存指標，錄的長度便需大些</li> <li>• 因爲連接的關係，儲取時間要比順序的慢些</li> </ul>
-------	--	---	---	--

## 五、參考書籍

1. 電腦系統概論（下冊） 華光電腦
2. APPLE II 使用手冊 張其邦
3. APPLE 機器語言 張其邦
4. 檔案設計 儒林出版
5. APPLE II 磁碟操作系統 李國凱
6. BASIC APPLE BASIC James S. Coan
7. Disc Operating System instructional and Reference Manual James S.

期第六十九頁至七十二頁。

註26：同註25。

註27：同註13，第三十一至三十四頁。

註28：同註13，第六十九至七十頁。

註29：同註13，第一〇一及一〇二頁。

註30：同註13，第一三五至一三八頁。

參考書目：

- 一、中國大陸研究叢書編委會，「共產主義與共產策略批判選編」，台北市正中書局，民國六十五年。一版。
- 二、李金清，「中共匪黨的眞面目」，民國五十八年十一月，第二版，著者印行。
- 三、政治作戰學校，「匪黨策略及對策研究」，政治作戰學校印行，台北市民國六十八年九月再版。
- 四、易陶天，「共黨理論批判」，台北市黎明文化公司，民國六十五年十二月一版。
- 五、謝傳聖，「共匪統戰顛覆實錄」，民國六十八年聯經出版公司印行。
- 六、林邁可，「共產主義錯在那裏」，台北市民國六十五年九月一版，聯經出版公司印行。
- 七、袁樹基，「粉碎共匪統戰陰謀」，中華史記編輯社，民國六十八年十一月一版。
- 八、「匪黨理論批判」，政治作戰學校編印，民國六十八年九月再版。
- 九、司法行政部調查局，「共黨問題研究」，民國六十五年七、八月第二卷第七、八期，民國六十七年九、十、十一、十二月第四卷第九、十、十一、十二期。民國六十八年一月第五卷第一期。民國六十九年十一月第六卷第十一期。

註7.：同註4，第九頁。引自註1.第一集，第二五—二七頁。

註8.：同註4，第九—十頁。

註9.：同註4，第十頁。

註10.：同註5，第三六六頁。引自司法行政部調查局編印「共產黨底解剖」，民國五十一年，再版。

註11.：同註10，第三六七頁。

註12.：同註10，第三六七頁。

註13.：同註4，台北政治作戰學校編印，「匪黨策略及對策研究」，民國六十八年九月再版第十一—十二頁。

註14.：蔡國裕，「中共建黨理論剖析」，司法行政部調查局，「共黨問題研究」民國六十七年十二月第四卷第十二期第二十一頁。

註15.：司法行政部調查局，「共黨問題研究」民國六十五年，七、八月第二卷第七、八期，九九頁與一一〇頁，「共黨問題史料選載」。

註16.：蔡國裕，「中共黨的組織原則剖析」，司法行政部調查局，「共黨問題研究」民國六十八年一月第五卷第一期，第二十三頁。

註17.：王章陵，「論共產黨統一戰線策略」，司法行政部調查局，「共黨問題研究」民國六十七年九月第四卷第九期，第十三頁。

註18.：同註17。

註19.：沈志遠，「略論統一戰線諸問題」，司法行政部調查局，「共黨問題研究」民國六十七年十一月第四卷第十一期第一一四—一一七頁。

頁。

註20.：同註19。

註21.：周鯨文，「共產黨的魔術——統戰工作」，司法行政部調查局，「共黨問題研究」，民國六十七年十月第四卷第十期第八十四頁至九十四頁。

頁。

註22.：同註17.第十九頁。

註23.：鄭一清，「抗戰勝利後共匪之初期叛亂」，台北司法行政部調查局，「共黨問題研究」，民國六十九年十一月第六卷第十一期第五十一頁。

十一頁。

註24.：黃星，「共匪民主鬥爭策略透視」，司法行政部調查局，「共黨問題研究」，民國六十八年一月第五卷第一期第五十七頁。

註25.：陳錫鏘，「關於馬、恩武裝群眾「暴力革命」思想之研究」，司法行政部調查局，「共黨問題研究」民國六十五年八月第二卷第八