

apple II

APPLE II FORTH-79

操作手册



北方电脑公司信息资料部

50989

APPLE II FORTH-79 操作手冊

(1983年修訂版)

鄒靖寧 譯

北方电脑公司信息资料部

譯 序

FORTH-79 這本書是專門配合 Micro Motion FORTH-79 磁碟片的使用手冊。在寫完 FORTH 入門一書之後，有很多讀者來信詢問，為什麼在 FORTH 入門上的程式有很多都不能執行或執行結果不一樣？。這是由於 FORTH 入門上的指令很多都是在 poly-FORTH 上的指令，而 poly-FORTH 在國內可能只有一、兩套系統，故一般的電腦系統若不修改基本指令，就無法與其匹配。但就 FORTH 的整個系統而言，仍然是了解 FORTH 最佳的教材。而 FORTH-79 這本手冊則恰可彌補這個缺點，書上的例子，程式，都是可以執行的，同時由它來發展出其他的系統，目前已經發展出：餐飲業的管理系統，庫存系統，很多的自動化控制程式，及中文 LOGO …等。手冊與磁碟片配合起來，就是一套能做事的系統。現在 Micro Motion 又推出 FORTH-79 第二版，及一張擴增磁碟片（上而有浮點運算，及繪高解像圖形的程式），這更是方便使用者，能直接用來解決問題，而不必再花心思來設計基本指令。對想了解 FORTH 操作系統的人更是有利，在祖磁碟片上有 60 幕，在擴增磁碟片上有 74 幕，的原始程式（Source Program），可供您參考。這是深入研究 FORTH 的最佳途徑。

本書雖然經過多次編校，但我相信仍然有很多不理想之處，請讀者來信指正。

FORTH 協會
鄒靖寧

目 錄

第一章 緒 論	1
1.1 FORTH-79 標準	1
1.2 其他資料來源	2
第二章 疊層與數字	3
2.1 疊層	4
2.1.1 列印指令	4
2.2 算術運算指令	5
2.2.1 加法	5
2.2.2 1+ 和 2+	6
2.2.3 減法	7
2.2.4 1- 和 2-	7
2.2.5 乘法	8
2.2.6 除法和餘數	8
2.2.7 2* 和 2/	9
2.2.8 混合運算	9
2.2.9 NEGATE	9
2.2.10 ABS	10
2.2.11 MAX	10
2.2.12 MIN	10
2.2.13 R 和 CR	12

2.3 疊層操作指令.....	12
2.3.1 DUP	13
2.3.2 SWAP	13
2.3.3 OVER	14
2.3.4 ROT	14
2.3.5 DROP	15
2.3.6 PICK與ROLL	15
2.4 FORTH 表示法.....	16
第三章 常數、變數、和陣列.....	19
3.1 常數.....	19
3.2 變數.....	20
3.2.1 檢查變數.....	20
3.2.2 改變變數.....	21
3.3 改變常數.....	23
3.4 陣列.....	24
3.4.1 字元陣列.....	27
3.4.2 陣列的初值設定.....	28
3.5 表格；陣列的補註.....	30
第四章 指令與詞典	35
4.1 印出訊息.....	36
4.2 定義指令.....	36
4.3 VLIST	37
4.4 FORGET	38
4.5 其他定義.....	39
4.6 詞典.....	42

4.7 保存詞典.....	44
第五章 流程控制.....	47
5.1 處理決策.....	47
5.1.1 邏輯運算指令.....	48
5.1.2 條件結構.....	51
5.2 迴路.....	56
5.2.1 有限迴路：DO LOOP	56
5.2.2 不定迴路.....	60
5.2.3 BEGIN...UNTIL	61
5.3 另一種 DUP 指令.....	64
5.4 返回疊層.....	66
5.5 提早結束.....	66
5.5.1 離開DO...LOOP	66
5.5.2 跳出一個指令.....	67
5.6 CASE 敘述.....	67
5.6.1 號碼選擇 CASE	68
5.6.2 位置判斷 CASE	69
第六章 大量量存及編輯程式.....	73
6.1 編輯程式.....	75
6.2 列出幕.....	75
6.3 使用一個幕.....	76
6.4 使用連續的幕.....	76
6.5 暫存區管理.....	77
6.6 暫存區指令.....	78
6.7 塊.....	79

6.8 跨越幾個塊的資料.....	80
第七章 數 字	83
7.1 二進位法.....	83
7.2 十六進位數字.....	84
7.3 字元與指令.....	86
7.4 單整數.....	86
7.5 比例算術.....	88
7.6 雙整數.....	89
7.7 正整數.....	90
7.8 混合數字運算指令.....	91
第八章 字 串	95
8.1 字串變數與數字實值.....	95
8.2 副字串.....	97
8.2.1 選擇副字串.....	99
8.2.2 改變副字串.....	100
8.3 字串的比較.....	101
8.3.1 相等字串.....	101
8.3.2 在字串內找尋副字串.....	102
8.4 字串的連結.....	102
8.5 字母及字串的轉換.....	103
8.6 字串陣列.....	104
第九章 使用者界面	107
9.1 單字元輸入與輸出.....	107
9.2 輸入字串和數字.....	110

9.3	文句列印格式	112
9.4	數字列印格式	113
第十章	深入探討	117
10.1	特殊情況	117
10.2	使用者變數	118
10.3	CREATE 及 DOES >	119
10.4	執行與編譯	121
10.5	使用機器碼	122
附錄 A	啓動 APPLE II	125
A.1	建 FORTH 磁碟	127
A.2	FORTH 擴充程式與應用程式	129
A.3	將 FORTH-79 第 1.2 版本轉換成第 2.0 版本	131
附錄 B	補充定義	133
附錄 C	詞彙	141
C.1	詞彙表示法	141
C.1.1	順序	141
C.1.2	疊層表示法	141
C.1.3	特質	142
C.1.4	大寫字母	142
C.1.5	讀音	142
C.1.6	疊層變數	142
C.1.7	輸入文句	144
C.2	FORTH 指令集	144

C. 3	雙整數指令集	189
C. 4	組合語言指令集	191
附錄 D	FORTH-79 必備之指令集	193
附錄 E	指令的功能分類	195
附錄 F	編輯程式	199
F. 1	編輯程式的使用	199
F. 2	編輯指令	199
F. 2.1	游標移動指令	200
F. 2.2	文句操作指令	202
F. 2.3	其他指令	204
F. 3	改變編輯指令	205
附錄 G	APPLE II 的公用程式	207
G. 1	從RESET 還原	207
G. 2	文句窗操作	208
G. 3	繪製圖形	209
G. 3.1	低解像度繪圖	209
G. 3.2	高解像度繪圖	210
G. 4	使用遊戲輸入 / 輸出	211
G. 5	呼叫機器語言副程式	211
附錄 H	6502組合語言定義	213
H. 1	簡單的組合語言定義	213
H. 2	開始一個組合語言定義	214

H. 3	指令格式	215
H. 3. 1	運算元及標示	215
H. 3. 2	定址法	216
H. 3. 3	運算碼	217
H. 4	結束一個組合語言定義	217
H. 5	疊層、暫存器，及第 0 頁	218
H. 5. 1	參數疊層	218
H. 5. 2	返回疊層	219
H. 5. 3	6502 CPU 暫存器	220
H. 5. 4	FORTH 假暫存器	221
H. 5. 5	第 0 頁的使用	221
H. 5. 6	疊層傳送副程式	222
H. 6	組合語言流程控制	223
H. 6. 1	條件分支	223
H. 6. 2	結構化的條件指令	224
H. 6. 3	迴路	226
H. 6. 4	套裝迴路	227
附錄 I	APPLE II 記憶圖	231
附錄 J	磁碟公用程式	235
J. 1	暫存區的重新分配	235
J. 2	幕的轉移及備存	236
J. 3	索引及分類	237
J. 4	TURNKEY 及詞典的保護	238
附錄 K	FORTH-79 內部結構	241

附錄 L	錯誤訊息	245
附錄 M	可移轉性	247
M. 1	改進的VLIST	249
M. 2	速度與可移轉性的關係	250
M. 3	測試可移轉性	251
M. 4	輸入 / 輸出向量	253
附錄 N	啓動擴增擴片	255
附錄 O	浮點算術	257
附錄 P	高解擴度烏龜繪圖	273

第 1 章

緒 論

這本手冊可以做爲你學習 FORTH 的指引。在你開始閱讀第二章之前，必須先找一套有 FORTH 語言的電腦。當你一章章地讀下去時，應該要按照書本所示的例子試著在終端機上做做看。本書中電腦回答的部份會以橫線劃出來。

如果你已經從 Micro Motion 購買了 FORTH 語言的軟體，你就必須先翻到附錄 A，看看如何開始用這種特別的電腦。

1.1 FORTH-79 標準

本書中所使用的 FORTH 是依據國際 FORTH-79 標準（1980 年 10 月）。同時亦備註了早期的 FORTH（FIG）模式（當它和 FORTH-79 有顯著差異時）。

除了 FORTH-79 中所提到的指令集之外，本書中也敘述了另外許多有用的 FORTH 指令。這些指令的定義見附錄 B。在你用這些指令以前必須先把它們的定義加入你的 FORTH 語言中。如果你用的是 MicroMotion FORTH，那麼這些定義早就已經被加進去了。如果不是，你就應該請一位熟悉你所用的 FORTH 系

統的人來替你將這些定義編進去。

附錄 C 是一個很完整而且按字母順序排列的小詞典，包含了本書所使用的全部 FORTH 指令。FORTH-79 必要的指令集在附錄 D 中。附錄 E 是依功用區分的指令集，如疊層操作指令，算術運算指令等。

1.2 其他資料來源

當地的大專院校通常都資助電腦社團。你也許能發現這些社團中有對 FORTH 感興趣的人。由於 FORTH 通常被用來當做控制天文望遠鏡的語言，你也可以查詢當地的天文學社團。你也可以打電話給電腦公司，查詢附近地區可能有的電腦社團。

FORTH 愛用者俱樂部 (FORTH Interest Group) 發行了一種雜誌，名為 FORTH DIMENSIONS，介紹新社團、研討會，及其他 FORTH 活動。FORTH DIMENSIONS 中有所有 FORTH 程式師感興趣的文章。FORTH 愛用者俱樂部 (FIG) 的會費及一年期 (6 本) FORTH DIMENSIONS 的費用共為美金 12 元。你可以將支票或現金寄到：

FORTH Interest Group
P.O. Box 1105
San Carlos, Ca. 94070

或直接打電話到 415-962-8653。

第 2 章

疊層與數字

FORTH 中所有的運算都是由指令來執行的。有些 FORTH 指令看起來並不像字——如 `+` 或 `*`——但是電腦卻認為是。就 FORTH 而言，任何一個字母或是一群字母，其前後兩邊都用一個空格和另一個字母或另一群字母分開，這就叫做一個指令。例如：

```
HELLO $10,000 >>> BYE
```

以上都是有效的 FORTH 指令，所以你現在就應該要習慣在每個指令之後打空白鍵，這樣 FORTH 才會認得你的指令。

數字也是 FORTH 指令。數字中沒有任何標點，通常也不能超過 32767，也不能小於 -32768。（較大的數字將在後面的章節中再作介紹。）負數前面有一個減號（負號）。

```
1 0 -1 32000 -64
```

以上也都是有效的 FORTH 數字。數字通常是被其它 FORTH 指令所使用，並且被保存在一個特別的位置「疊層」上。

* 譯註：因為 FORTH 指令，可用 124 個 ASCII 字符來定義，有些字符在文句中，容易產生誤解，諸如 (,) , (:) , (;) 等，因此在本手冊中，容易混淆的指令在文句中，用「反相」方式表示。

2.1 疊層

疊層是數字的「暫存檔」，暫存指令運算所需的數字，任何打在螢幕上的數字都會輸入疊層上等待後來之用。這些數字儲存的順序是非常重要的；你必須要瞭解疊層如何作用，以及如何做最佳的操作。

我們可以用一種很熟悉的方法來想像FORTH疊層，假想為放在一個彈性架上的自助餐盤，如果你將三個餐盤分別標示為1，2和3，並依序放在疊層頂上，當你再取下它們時，順序就會反過來了：你第一次取下的是3號，第二次取下的是2號，第3次取下的是1號。你沒有辦法先拿第1號的盤子，除非先取下第3號和第2號的盤子。

# 1 → — # 1 — (頂值)	把 # 1 堆上疊層。
# 2 → — # 2 — (頂值)	把 # 2 堆上疊層。
# 1 (第2個值)	
# 3 → — # 3 — (頂值)	把 # 3 堆上疊層。
# 2 (第2個值)	
# 1 (第3個值)	
← — # 2 — (頂值)	把 # 3 取出疊層。
# (第2個值)	

2.1.1 列印指令

以下的指令

. (dot)

會將疊層頂端的值取出並顯示在螢幕上。你應該注意的是這個數字再也不在疊層上了，如果你另有用途，就必須用其他運算來保留它。如果你打一連串的点 (dots)，一個點印出一個數字，FORTH 可以將疊層上全部的數字印出來 (依前述的順序印出)，現在來試試看，先打一串數字：1 2 3 4，要確實加上足夠的点：

1 2 3 4

現在打 RETURN 鍵，讓 FORTH 認出該指令，然後它就會回答你：

4 3 2 1 OK

如果你打的点比疊層上的數字多時，FORTH 就會告訴你疊層空了，並顯示出一個錯誤訊息，如 EMPTY STACK，再打一点用來檢查：

. 0 ? EMPTY STACK OK

當你使用 FORTH 時，必須先掌握疊層的觀念，因為你會常常被要求記住你的資料是如何儲存在那兒的。更進一步的，你還要學習操作疊層上資料順序的方法，有個簡便的方法可以幫助你記住這個觀念，就是 LIFO — 後進先出。

2.2 算術運算指令

2.2.1 加 法

FORTH 中最不尋常的是算術問題中元素——在電腦術語中稱為「自變數」(“arguments”) ——輸入的順序。通常你可

能將一個加法問題寫成下面這樣：

$$2 + 3$$

在許多電腦語言和計算機中這種寫法是正確的。然而，在FORTH中，同上述的運算應寫成：

$$2\ 3\ +$$

為什麼呢？主要是為了便於FORTH處理。同時，你以後就會發現這個系統在較複雜的代數運算中可以減少使用括弧。

現在回到我們簡單的加法問題上，當你打

$$2\ 3\ +$$

並打RETURN，FORTH就會將2和3存在疊層上（以前述後進先出的順序），把它們加起來，然後用和來代替此兩數。現在疊層上有什麼呢？只有答案5。當你再打一個點，疊層內容就會顯示出來：

$$2\ 3\ +\ .\ \underline{5\ OK}$$

加一長串的數字也很容易，但是你必須要記住打入足夠的「+」運算元。FORTH一次只能加兩個數字，它會先取疊層頂端的兩個數字加起來，並留下兩數的和，然後它會取疊層的下一個數字，把它們相加，並留下和，等等。你只需在最先的兩個數字之後放一個+，並在其後的每一個數字之後放一個+：

$$9\ 17\ +\ 31\ +\ 47\ +\ 998\ +\ .\ \underline{1102\ OK}$$

或者同一問題可以寫成如下的形式：

$$9\ 17\ 31\ 47\ 998\ +\ +\ +\ +\ .\ \underline{1102\ OK}$$

2.2.2 **1+**和**2+**

對某些經常執行的運算，FORTH將運算中的元素結合在一起，以節省你電腦的時間與空間，例如：如果你想要在14上加