

# 第四代程式語言——FORTH

丁陳漢蓀博士 編著

# FORTH

87

73

永華科技圖書出版社

# 第四代程式語言——FORTH

丁(陳)美蓀博士 編著



FORTH

The word 'FORTH' is rendered in a large, bold, 3D block font. The letters are white with thick black outlines and drop shadows, giving them a three-dimensional appearance. The text is slightly curved and appears to be hanging from a thin black line that runs across the middle of the page.

版權所有 · 請勿翻印

---

---

## 第四代程式語言——FORTH

譯著者：丁陳漢蓀博士

發行人：馬超

出版者：永華科技圖書出版社

地 址：九龍北帝街60號

東寶大廈三樓

承印者：永華科技圖書出版社印刷部

地 址：九龍土瓜灣新碼頭街

幸福工廠大廈五樓

---

---

定 價：港幣H.K.\$19.00

# 序

人最高的價值在於承先啓後，丁陳漢森博士藉返回講學之便，將這套備受專家推崇的第四代程式語言大力推薦給國人，引起了陣陣迴響。

我沾了曾是同學的光，承丁陳博士之指導，我們得將這套語言改爲“中文第四代語言”，頗受年輕朋友們的歡迎，正擬廉價推廣，以發揚丁陳博士之心願。

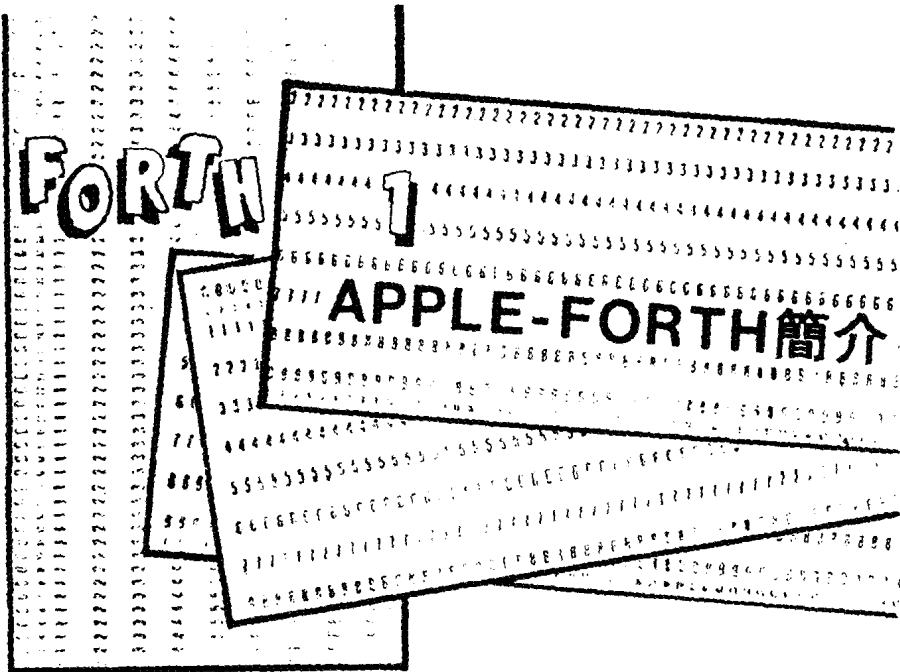
最令人欽佩的是丁陳博士的高風亮節，在舉世滔滔唯利是圖的今日，他未索取分文報酬，利用寶貴的點點滴滴的時間，完成這本鉅著交給我們來出版發行。現在，他已回到僑居地的原工作崗位上，臨行他只囑咐了一句話：希望能用最優惠的代價，將這本書的種籽，播在國內豐沃的園地上。

# 目 錄

<b>第一章 APPLE-FORTH簡介</b> .....	1
1-1 FORTH的起源.....	1
1-2 Apple FORTH.....	4
1-3 FORTH詞典.....	11
1-4 疊 層.....	12
1-5 數據疊層.....	15
1-6 返回疊層.....	18
1-7 後算符算法.....	19
1-8 FORTH名詞集解.....	22
<b>第二章 數字及算術</b> .....	27
2-1 數字的輸入及輸出.....	27
2-2 算術及比例指令.....	30
2-3 數字的轉換.....	33
2-4 實 值.....	35
<b>第三章 編 輯</b> .....	37
3-1 文字的編輯工作.....	37
3-2 FORTH程式的測試.....	45

3-3	FORTH 指令的編輯	47
3-4	列印原程式	50
<b>第四章 常數、變數與數陣</b>		52
4-1	常 數	52
4-2	變 數	53
4-3	雙常數與雙變數	55
4-4	數 陣	58
<b>第五章 結構化程式</b>		61
5-1	條件分支	61
5-2	比較指令	65
5-3	循 環	67
5-4	循環結構舉例	74
<b>第六章 詞典、詞彙與記憶</b>		81
6-1	記憶區分配	81
6-2	磁碟記憶	83
6-3	假磁碟	38
6-4	指令結構	89
<b>第七章 字串與數字列印法</b>		105
7-1	字串操作指令	105
7-2	字串的暫存區	107
7-3	字串的輸出指令	107
7-4	字串的輸入指令	110

7-5	字串轉變為數值	115
7-6	數字的列印指令	116
<b>第八章 程式編輯</b>		<b>121</b>
8-1	程式的編輯	121
8-2	行編輯指令	122
<b>第九章 組合程式</b>		<b>126</b>
9-1	6502的機器碼	126
9-2	數據疊層的使用法	127
9-3	返回疊層	128
9-4	FORTH的暫存器	129
9-5	6502的暫存器	129
9-6	備用記憶區	130
9-7	流程的控制	131
9-8	其他結束指令	133
附錄一		135
附錄二		137



## 1-1 FORTH的起源

FORTH是美國的一位程式師查爾斯·摩爾（Charles Moore）發明的一種電腦語言。摩爾是一個所謂自由工作的程式師，為不同的公司機構寫作各種電腦程式，因而他使用過各種型式的電腦，也接觸過各種電腦語言。他極不滿意電腦公司供給電腦界使用的各種語言及發展軟體的工具，因為這些軟體的工具都不能讓使用者有效地指揮電腦工作以解決他們所面對的程式問題。摩爾在他自己寫作程式時，是持守着藝術家對他作品美化的理念，對程式的品質要求十分嚴格。他深自相信，雖然一個程式問題可以有許多不同的解，但是最好的解，必定是最簡單的解。因為簡單（Simplicity）是效率、速度，及對問題的基本了解等程式品質的總括。摩爾在五十年代及六十年代所積聚的程式經驗在一九六九年結晶成為我們現在所知的FORTH語言。FORTH乃是FOURTH（第四）的縮寫，因為摩爾認為這套語言是第四代的電腦語言，超越了當時方才出現的第三代電腦語言。



## 2 第四代程式語言 FORTH

摩爾在一九六九年受聘於美國佛吉尼亞州的國立無線電天文台。他的任務是設計全套的中型電腦的程式系統，來控制巨大的無線電天文望遠鏡的自動運轉，由收集、記錄並加以處理這架望遠鏡所收錄的天文資料。在很短的時間內，摩爾成功地將 FORTH 語言裝置到天文台中各式的電腦上做各種不同的工作。FORTH 語言也隨着無線電天文望遠鏡而傳佈到美國、歐洲、南美洲及澳洲各地，成爲天文界廣用的電腦語言。國際天文學會也在一九七六年決定採用 FORTH 作爲天文台自動化的標準語言。

摩爾在一九七三年離開美國國立無線電天文台，他邀集幾位同僚創設了 FORTH, Inc. 公司，專門銷售 FORTH 的軟體系統並提供各種特殊的軟體設計與服務。他們陸續地發展並推出爲各種中型電腦用的 mini-FORTH，爲微電腦用的 micro-FORTH。在一九七九年他們推出一個多用途，多人共用（multi-programming）的系統，稱爲 Poly-FORTH。這是目前在 FORTH 語言市場上功能最高的 FORTH 系統。FORTH, Inc. 公司雖然樂意銷售他們的 FORTH 軟體系統，但是爲了保護他們的權益，却不希望使用者完全瞭解 FORTH 語言。因此他們並不熱心於推廣這種語言。又因爲他們的產品價格相當貴，例如 poly-FORTH 是在美金五千元左右，所以傳佈的幅度有限。與同時發明的 PASCAL 相比，步調就落後了許多。

在歐洲由於天文台界的引進，許多熱心於 FORTH 語言的程式師組織了一個歐洲 FORTH 使用者協會（European FORTH User's Group）簡稱 FUG。他們經常集會研究 FORTH 語言，並且交換 FORTH 程式以及實用的經驗。爲了便利使用者間交換程式及傳佈 FORTH 資料。他們在一九七七年制定了一套 FORTH 的語言標準，稱爲 FORTH-77 標準。這是一套基本的 FORTH 指令集，用其中的指令寫作的程式，可以很方便地在不同的電腦上操作。

一九七五年以後，微電腦逐漸普及，各型微電腦的使用者都在各地區成立微電腦俱樂部，在舊金山南郊的矽谷中最早成立的組織叫做 Homebrew Com-puter Club，經常在史坦福大學集會，一些使用 FORTH 的程式師在這個俱樂部裏經常碰面而決定自行成立 FORTH 的討論會。他們在一九七八年正式成立了 FORTH 協會（FORTH Interest Group），這是一個非營利的組織，它的目的乃是宣揚及教導這種語言，並且是針對微電腦使用者的需要而以定期的討論會及各種出版物來進行推廣的工作。

FORTH 協會有兩項重要的成就。其一是他們在成立之際就組織了一個 FORTH 工作團（FORTH Implementation Team），根據 William Ragsdale 在 Apple 電腦上發展出來的一套 FORTH 程式作為模型，建成了六套 FORTH 的程式，分別用在 6502、8080、9900、6800、PDP-11 及 PACE 等六個不同的微電腦上，這就是所謂的 FIG-FORTH 模式。這六個 FORTH 系統的原始程式都在一九七九年初發表，FORTH 協會將這些程式的版權公開。每份原程式僅售美金十元，因為這低廉的價格及系統的品質，使得 FORTH 在美國能迅速的傳佈開來。同時 FORTH 協會並與 Byte 雜誌合作，在一九八〇年八月份的 Byte 中刊出了 FORTH 專集，使得玩微電腦的人，差不多都知道了 FORTH 的存在。

FORTH 協會的另一項成就是 FORTH 語言的標準化。它在一九七八年及一九七九年兩次邀集了出版 FORTH 系統的公司及歐美有經驗的 FORTH 程式師在加州洛山磯附近的 Catalina 島上集會討論 FORTH 語言標準化的問題，這兩次會議的結果是 FORTH-78 及 FORTH-79 兩套標準。FORTH-78 的語言標準規定及考慮不甚周詳，在 FORTH-79 標準中加以改進而成為一套基本而定義詳密的指令集。因為主要的 FORTH 公司及使用者都同意以此套標準作為他們以後發展的基礎，使

#### 4 第四代程式語言 FORTH

得各種 FORTH 語言分歧的現象得以改善。這對 FORTH 語言的推展，程式的交換及資料的出版都有很大的助益。

我是在一九八一年九月回到台灣，來前收集了許多 FORTH 的系統程式及資料，以期提供國人這種發展軟體的犀利工具。在這半年多的時間裏，也激發了不少朋友對這種語言的興趣。大家合力在不少微電腦系統上發展出來許多 FORTH 的系統，我所知道的目前能使用的系統有 CP/M、Apple、PDP-11、ZDS、MDS、AIM-65、68000、PA-800 等。北部好些大專及機構都開始在利用 FORTH 做一些具體的工作。每個月第四個星期六的下午在台大電機館也固定地舉行 FORTH 討論會，交換研究的心得。最近 Apple 逐漸普及，我深感若能有一個燒錄在 ROM 中的 FORTH 系統，不必依賴磁碟機貯存程式的話，極有助於大家學習使用 FORTH 語言。這一套 Apple - FORTH 系統，具有 Editor 及 Assembler 功能而且符合 FORTH-79 標準。使用 Apple 的朋友們，可以利用它來熟悉 FORTH 並能做相當大的專業工作。

### 1-2 Apple-FORTH

Apple-FORTH 是一片 ROM Card，可以插在 Apple 電腦主機板左後任意一個插孔中，這片 ROM 板上的四片 2716 EPROM 記憶 IC，錄有一個 FORTH 語言系統。開機後這個 FORTH 系統即可開始操作。因為整個系統已燒錄在 ROM 中的，所以可以很可靠地工作，如果有問題時，按 RESET 鍵或重行開機操作也是很方便的事。藉着這一片 ROM 板，可以將 Apple 電腦變成一個可靠而方便的 FORTH 電腦，供給有興趣研究 FORTH 的朋友一個很方便的工具。

這個 Apple - FORTH 的程式，是根據 FIG - FORTH 的 6502 程式建成的。它具有譯碼 ( Interpreter )、編譯 ( Compiler )、編輯 ( Editor ) 及組合 ( Assembler ) 的功能，可以用 FORTH 的高階語

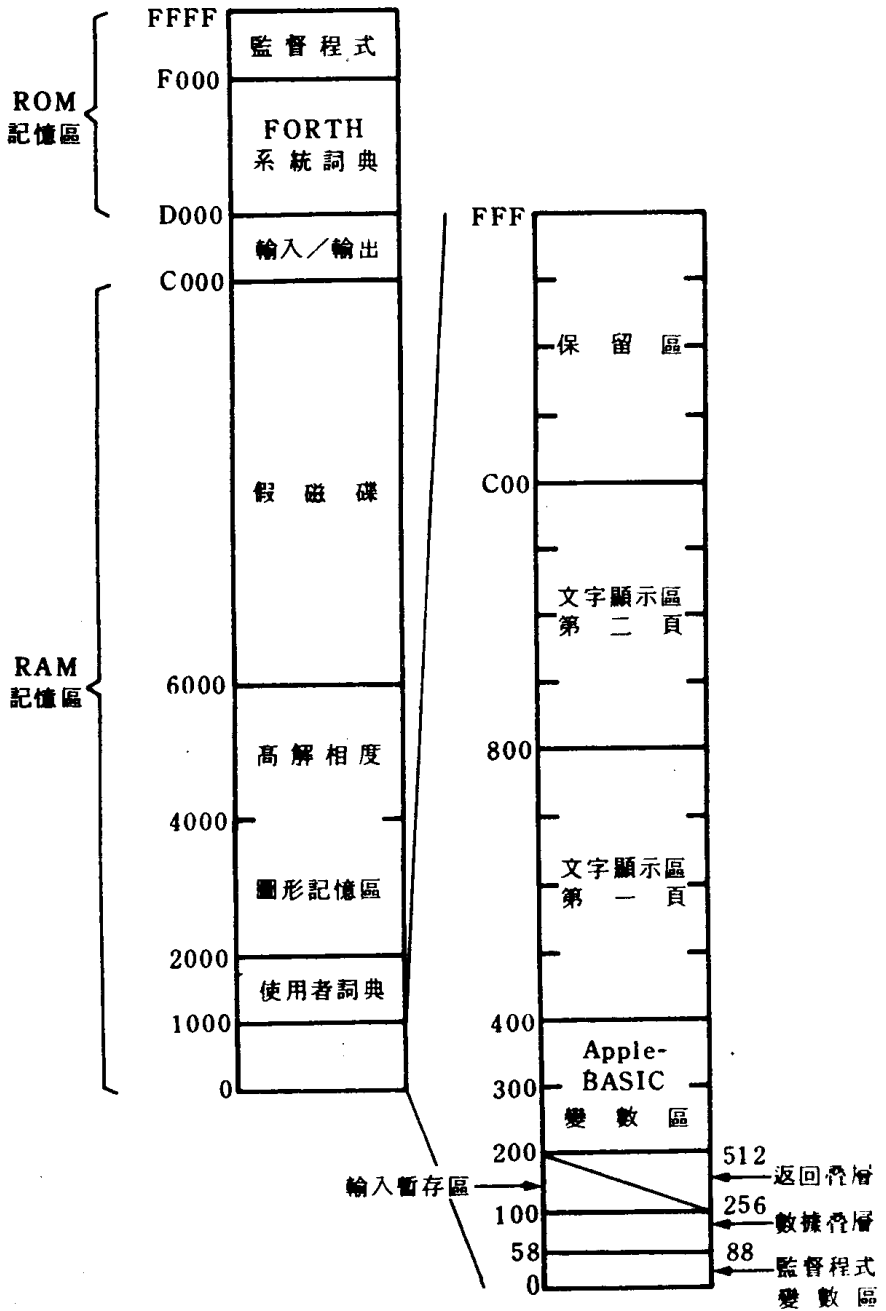


圖 1-1 Apple-FORTH 記憶區分圖

## 6 第四代程式語言 FORTH

言來遵循結構化程式法 ( Structured Programming ) 寫作程式。它需要一個有 48K RAM 的 Apple 電腦作為主機，並且使用一般用的錄音機作為貯存程式的媒介。因為 Apple 的磁碟機必須由國外進口，價格昂貴也不普及，所以我們這個 Apple - FORTH 不包括使用磁碟的指令，而設法盡量用磁帶的形式來存取大量的資料或程式。其目的乃是提供給使用者一個合理價格的 FORTH 系統，以便 FORTH 語言的推廣。

圖 1-1 是記憶區分圖中顯示這個 FORTH 系統中各部份 ROM 及 RAM 記憶的位置及其功能。Apple - FORTH 實際上取代了 BASIC 在記憶中的地位。圖 1-2 是 FORTH 詞典中指令按其功能區分的示意圖。

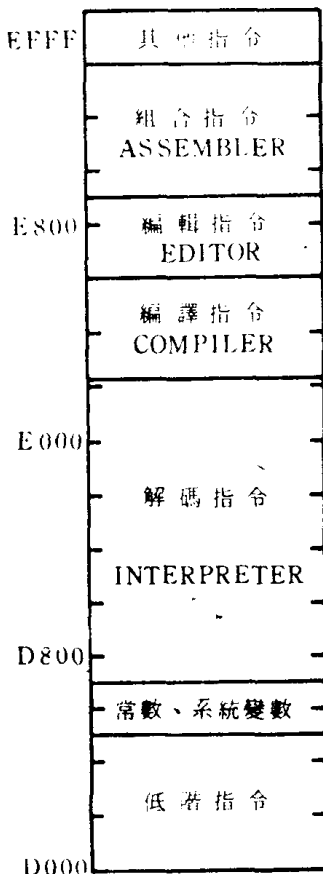


圖 1-2 Apple-FORTH 系統  
詞典及其功能區分

## 開機程序

將Apple主機電源切斷後，把Apple-FORTH ROM Card 插入主機板後部任意一個插孔中。打開CRT 螢幕機，再打開Apple主機電源，立時可以聽到“嗶”的一聲，同時在螢幕上會顯示下列字樣：

```
APPLE I
79-FORTH 2.1
OK
```

表示已經跳入FORTH系統，等待使用者按入FORTH指令了。

連續按入Return鍵，螢幕上就顯示一連串的OK，表示FORTH系統正常的操作。按入VLIST指令後再按Return鍵即可。VLIST指令是最常用以確定FORTH系統是在正常情況下的指令。

按入MON指令再按Return，則跳Monitor系統；

```
OK
OK
MON
*
```

此時出現星號可以執行任何Apple Monitor的指令。FORTH與Monitor共存在此Apple中，可以隨意呼叫交換使用。在Monitor中我們最需要的功能是将資料存入磁帶及將資料自磁帶讀入虛磁碟記憶區。若磁帶上錄有FORTH程式，而我們希望將其讀入時，可在跳入Monitor，再鍵入以下Monitor指令，將磁帶上的FORTH程式讀回Apple系統中。如：

```
*6000·BFFFR
```

## 8 第四代程式語言 FORTH

先調整適當音量並按下磁帶機 PLAY 鍵後再在 Apple 主機上按 Return 鍵，即可開始讀入資料。至星號重新顯現時，資料即正確讀入，否則有 ERR 字樣出現，表示讀取資料錯誤。

將虛磁碟記憶區的 FORTH 程式或資料存入磁帶中的指令是

```
* 6000·BFFF W
```

有關磁帶機操作的詳細步驟，請參考 Apple II Reference Manual 第 46 頁及其後的說明。

由 Monitor 跳回 FORTH 時，有好幾個跳入點，地址是開機之後建立 FORTH 系統的跳入點，如果 FORTH 系統已經在使用中（即 CLOAD START），許多新指令已建入詞典，則必須用下述指令進行 Warm Start 的動作，以免將建好的詞典清洗掉：

```
* D004 G
```

```
OK
```

如此回到 FORTH 系統而詞典的內容還是保留，可繼續原來在 FORTH 中的工作。跳入 D000 時：

```
* D000 G
```

```
79-FORTH 2.1
```

```
OK
```

Cold Start 的動作，會將系統以外的指令清除，剩下的詞典內容與開機後的相同。同時也是清除詞典的一個方法。在 FORTH 系統中操作時，指令 COLD 也做這樣相同的動作。

假如 Return 及 VLIST 等指令動作無誤時，FORTH 系統是可以正常地操作。如果系統有任何問題，VLIST 通常不能操作。在這種情

形之下，使用者應該檢查Apple本機是否能正常操作。關掉主機電源，將Apple-FORTH ROM Card拔出來，再開Apple看BASIC是否能正常操作。一般拼裝Apple最常見的問題是RAM晶片故障，在Apple BASIC中可用

```
] PRINT FRE(0)
```

指令列印RAM記憶最高值，若此值不在-16000附近，則必定有RAM記憶失效，將之換新可能即可正常操作。

其他問題則會可能是ROM Card插入不適當，接觸不良，此外問題多半會影響到BASIC的操作。可參照在BASIC中偵錯的方法去進行分析改正。

若主機板後方沒有ROM Card的插孔座時，使用者可以將主機板上CPU前面的四片BASIC的2716 IC拔出來，將ROM Card上的ROM拔出依序插入主機上2716的插座，FORTH的ROM # 0, 1, 2, 3分別插入D0、D8、E0、E8等位置，F8位置的ROM必需保留，這樣Apple即變成一架真正的FORTH電腦了。這是我希望見到的事。因為BASIC語言只是很好的學習電腦的工具，真正高品質的程式，不宜用BASIC來發展。

學習FORTH語言，請參考STARTING一書，並且按其中例題實習，可按步就班地熟悉各個FORTH指令，並以之為基礎發展應用程式。

### 錯誤信號 (Error Messages)

FORTH系統在執行時若遇到錯誤的地方，在錯誤發生的地方即終止運算而返回操作系統，並會印出一列文字：

```
MSG # 4
```



此數之值即指示錯誤發生的原因。通常發生的錯誤原因及其號數如下表：

MSG #	原因
0	無此指令
1	Data Stack
2	詞典空間填滿
4	指令名稱重覆
17	指令不能用在定義以外
18	指令不能用在定義之內
19	結構錯誤
20	定義未能完成
21	指令在系統中不能刪除
22	指令只能在編譯磁碟區時用
24	詞彙錯誤

FORTH 標準的信號文字輸出法是將信號文字儲存在磁碟記憶的第 4、5 兩塊 (BLOCK) 中，然後用 MESSAGE 的指令將某一行文字印出來。這些文字可以是錯誤信號，或是使用者自己選定的印出文字。使用這種方法輸出文字時，除了必須將這些文字資料存入磁碟記憶區之外，並要將變數 WARNING 設定為 1。之後，所有的錯誤信息即不再以數目字表示而印出適當的文字。錯誤信息的內容及排列方式，請參閱 FIG-FORTH Installation Manual 第 20 頁。其使用法請查閱 MESSAGE, ERROR, ? ERROR 及 WARNING 等指令之詳細定義及其使用法。

因為此套 Apple-FORTH 不使用磁碟，所以使用者必須利用 Editor 在第 4、5 塊中加入適當文字以後，再以下列指令