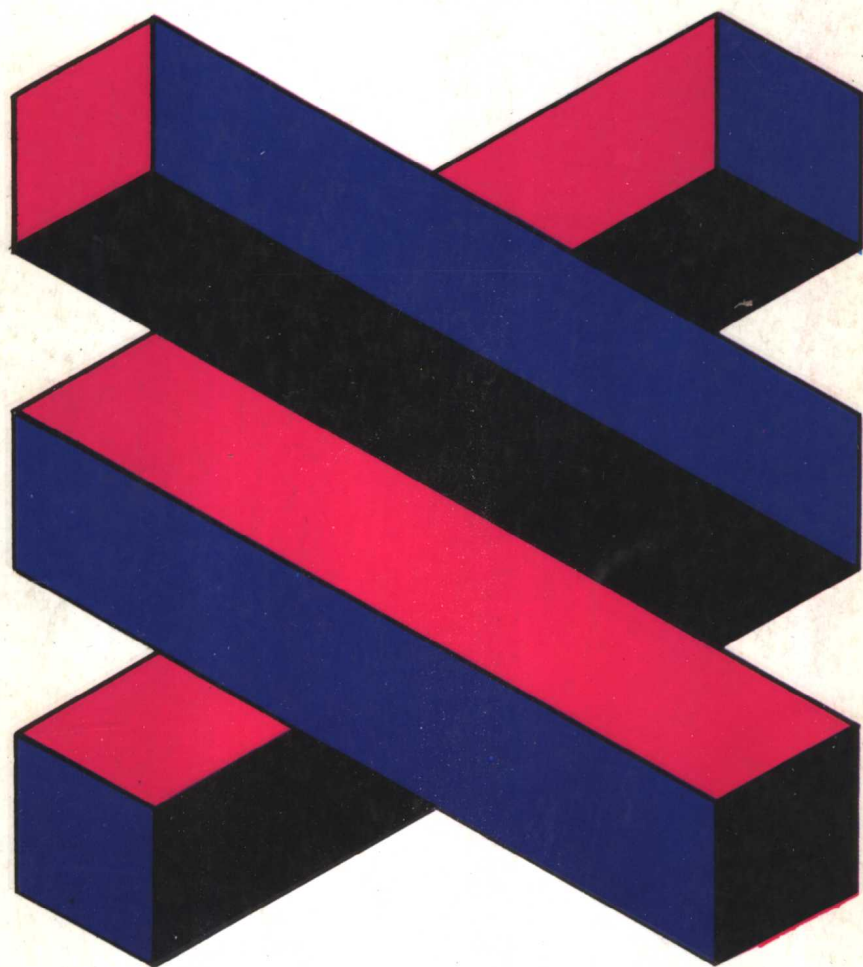


FORTH 概論

第四代電腦語言新形象

王學智 ■ 譯編

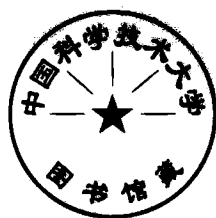


協群科技出版社

FORTH 概 論

第四代電腦語言新形象

編譯・王 學 智



協 群 科 技 出 版 社

FORTH 概論

編譯者：王 學 智

出 版：協 群 科 技 出 版 社

發 行：協 群 科 技 出 版 社

香港中環卑利街684號二樓

印 刷 者：廣 源 印 務 局

青山道875號工廠大廈

定價：7 H.K.\$

編輯大意

電腦自從一九六四年迎接了第三代以來，可以引用一日千里來形容其發展的神速，其適用範疇與適用方法也日漸拓展，今天已經面臨第四代的電腦時代。MIS 的高度進展、資訊化時代、資訊產業、資訊工業與數年來我國的電子處理也都隨之快速地發展。

本編輯部為推動資訊教育，有效培植資訊科技人才，促進資訊工業研究與發展，數年來遴聘國內外專家學者，網羅英、日、德文等有關資訊名著，逐譯成中文，供各級學校學生及社會大眾閱讀。

所編輯之各種書籍，不僅圖文並茂，逐譯工作嚴謹，且力求文字簡明、內容深入淺出，以期每位讀者有一卷在手，智珠在握之感，至於印刷之清晰精美，猶為餘事。望有志於資訊科學者，多加利用，則為本編輯部榮幸之至。

本編輯部同仁為求盡善盡美而不遺餘力，以充裕的時間編審。唯魯亥之處仍恐難免，尚祈博雅君子，不吝賜教，俾於更正，不勝感激之至。

前 言

在經常不斷地變化的程式計劃語言的世界，最近登場的 FORTH，是開始為世所矚目的程式語言之一。顆粒雖小但看來蠻新鮮的，亦即是適合於微電腦設計程式的語言。話雖如此，FORTH 可不是只有微電腦才能使用，FORTH 的利用價值，在更「大型」的電腦也許仍然不會變。總而言之，對於要使用 FORTH 的人來說，和電腦的規模或型號全然沒有關連。成為問題的，只是 FORTH 有關的活動幾乎全部是在微電腦的領域進行。

FORTH 的登場不是只在現有程式語言名單中多加了一個新的項目而已，FORTH 是將所謂的函數性程式計劃 (functional programming)，把重點放在生產性與可靠性以及執行效率的目前最新的語言。和 FORTH 有連帶關係的概念，有結構化程式計劃或上下 (top-down) 設計法，或也有虛擬記憶等。不過，FORTH 不僅是止於這些已往大家所熟悉的觀念而已，它代表了現代程式計劃技巧的新趨勢。

FORTH 語言本身的結構，是根據逆波蘭記法 (reverse Polish notation) — 有些計算機學者稱為後置記法 (postfix notation) — 為依據。和此基本想法在效果上設計、定義的語言結構相輔助，對於當作語言要求高度的彈性，與從特定應用領域環境之各種要求，乃產生了能適應語言之能力。

本書將以 FORTH 來招待你。主要是以使用這種語言進程式計劃者，程式設計根據 FORTH 的系統或應用者，以及和最近的電腦技

2 FORTH概論

術之進步同一步調的人士等為對象。本書所說明的內容，或多或少都含有背景性的知識，除了這以外，盡可能直接地邁進FORTH語言的階段。這便是本書的主要目的。

本書由下面的 10 章編輯而成：

第 1 章：FORTH的觀念—在本章，舉述 FORTH 的基本概念。

第 2 章：電腦的基礎—使你獲得有關微電腦的較基本的概念與知識。

第 3 章：軟體技術—對程式計劃、軟體系統以及有關電腦的應用，補充了基礎知識。

第 4 章：逆波蘭記法—說明逆波蘭記法、疊式儲存器概念以及解釋性的處理。

第 5 章：基本計算與堆疊操作—簡述算術演算子、基數、疊式儲存器操作、算術函數以及補數演算。

第 6 章：常數、變數與儲存器操作—說明 FORTH 的常數以及定義與使用方法，值之讀出 / 存放之方法。

第 7 章：定義與輸出入—詳細說明 FORTH 演算之定義法，以及對於顯示輸出操作。

第 8 章：控制結構—解釋適合於結構化程式計劃的控制結構，與 FORTH 的其它表現法。

第 9 章：倍精度演算—解釋 FORTH 的倍精度數的處理法，以及相關連的各種機能。

第 10 章：資訊管理—說明 FORTH 語言的記憶區域之構成，對磁碟的輸出入、程式之管理，字元之處理，鍵盤之處理，輸出時的格式控制或變換等，有詳細的說明。

本書的目的不用說就是要幫助各位了解 FORTH 概念。筆者是明確地意識這個目的，有關各項目是在淺顯的文章上提出例子來表示。

有關於電腦本身或程式計劃的預備知識，在爲了要完全了解本書或要學 FORTH，這並不是特別需要的。只是，各位如果以總體性去了解電腦究竟是什麼的話，也許可以幫助各位認識 FORTH 語言的功能與彈性。此外，本書所包含的各種話題，即使對不能使用 FORTH 語言的人士，也可學會像 FORTH 那樣的思考方式。爲了要幫助複習而附加了專有名詞表，同時，也按各章準備了習題以及其解答。

筆者自信本書是有助於從事有關計算機工作者、技術者或商業分析者以及致力於微電腦的人士邁向 FORTH 語言的綜合性的入門書。在這裏因爲有非常清楚的理由，沒有說明和 FORTH 系統本身有關的一些事項— FORTH 語言的處理系統、編排常式、執行時環境以及語言本身的擴大性等。凡對於這些事項有興趣的讀者，可以洽詢 FORTH Interest Group, P.O. Box 1105, San Carlos, California 94070。⁽¹⁾ 也許可以獲得各種資訊。

直到現在爲止，FORTH 所具有的能力，尚未完全地受到開發，而語言本身也在不斷地進化。因此，本書只不過是引導你有豐碩的將來而已。此外，在實際的程式製作時，也許需要有關 FORTH 的各種使用者指南。

本書的製作在儘可能的範圍使用了利用微電腦的文件製作裝置。這是筆者爲對 FORTH 有興趣的人士，不遺餘力地提供了適時資訊的緣故。對筆者發行人來說，要編輯本書，其實真是一種很愉快的工作，在這裏的願望正如我們感覺到將本書公諸於世的這份喜悅一樣地，也確實盼望各位讀者能高高興興地學習 FORTH 語言。但願你從事於一種愉快的程式計劃。

編輯本書時，獲得了多方面寶貴的協助與建議。首先，對於發行人 O. R. Petrocelli 氏的有益的教言以及把 FORTH 語言以嶄新式的題材編輯成書推出於世這種先見之明與勇敢果斷表示佩服，同時

4 FORTH概論

由衷感謝 Bruce Tucker 氏，提供了適時適地的資訊，有如錦上添花。吾妻 Margaret 擔任了語句處理的部分，爲了要製作立即照相版面，過了既長又無聊的時間，在寫本書的全部過程，做我的一個好伴侶，在這裏一併致謝。

〔譯註〕 由FORTH使用者所構成的 FORTH Interest Group(FIG)，正在進行發行會員雜誌等各種活動。

Harry Katzan, Jr. Stillhouse Road
Freehold, N. J. 07728

1981年1月

FORTH概論目錄

第1章 FORTH 的觀念

1・1	桌上電子計算器與 FORTH	11
1・2	數式的記法	12
1・3	使用時的環境	13
1・4	疊式儲存器	14
1・5	FORTH 的演算	14
1・6	執行模式	15
1・7	定義模式	16
1・8	FORTH 的能力	17
1・8	專有名詞	18

習題

第2章 電腦的基礎

2・1	計算機設計的哲學	22
2・2	電腦的記憶單位	23
2・3	硬體、軟體以及韌體	26
2・4	微電腦系統的結構	27
2・5	微處理機的構成與動作	29
2・6	疊式儲存器的動作	33
2・7	磁碟儲存器的技術	37
2・8	專有名詞	39

習題

6 FORTH概論

第3章 軟體技術

3·1	為什麼使用軟體？	45
3·2	軟體的分類	46
3·3	什麼是演算法？	47
3·4	什麼是程式？	48
3·5	組合語言	49
3·6	程式計劃語言	50
3·7	程式的結構	53
3·8	語言的處理系統	54
3·9	組合語言程式	55
3·10	編譯程式	55
3·11	翻譯程式	57
3·12	監督程式與操作系統	58
3·13	實用系統	60
3·14	開發系統	60
3·15	專有名詞	62

習題

第4章 逆波蘭記法

4·1	數式	66
4·2	式的結構	67
4·3	字腰記法與後置記法的互相變換	73
4·4	字腰記法的解釋性的執行	75
4·5	專有名詞	78

習題

第 5 章 基本計算與堆疊操作

5.1	FORTH 的語句.....	83
5.2	關於句點.....	85
5.3	疊式儲存器的表現方法.....	85
5.4	比較基本的算術演算.....	86
5.5	數值的表現方法.....	90
5.6	疊式儲存器操作語句.....	91
5.7	算術函數.....	96
5.8	補數演算.....	99
5.9	專有名詞.....	103

習題**第 6 章 常數、變數與儲存器操作**

6.1	常數.....	108
6.2	變數.....	109
6.3	從儲存器讀出.....	110
6.4	向儲存器寫入.....	111
6.5	向儲存器的加法.....	112
6.6	辭典.....	113
6.7	專有名詞.....	116

習題**第 7 章 定義與輸出**

7.1	冒號定義.....	119
7.2	註解.....	121
7.3	點輸出.....	122

8 FORTH概論

7.4	點-R輸出.....	123
7.5	輸送架轉回.....	124
7.6	字元輸出.....	124
7.7	畫面操作.....	125
7.8	空白輸出.....	126
7.9	無正負號的輸出.....	127
7.10	儲存器內容的顯示.....	128
7.11	專有名詞.....	128

習題

第8章 控制結構

8.1	邏輯值.....	131
8.2	比較演算.....	132
8.3	邏輯演算.....	135
8.4	DO迴路.....	138
8.5	IF敘述(條件敘述).....	144
8.6	EXIT、LEAVE操作.....	147
8.7	條件迴路.....	149
8.8	專有名詞.....	155

習題

第9章 倍精度演算

9.1	倍精度表現.....	159
9.2	算術演算.....	161
9.3	疊式儲存器操作.....	163
9.4	算術函數.....	166

9 · 5	比較演算.....	168
9 · 6	混合演算.....	171
9 · 7	輸出操作.....	173
9 · 8	常數與變數.....	174
9 · 9	記憶體操作.....	175
9 · 10	專有名詞.....	177

習題

第10章 資訊管理

10 · 1	記憶體結構.....	181
10 · 2	分配.....	182
10 · 3	磁碟輸出入.....	182
10 · 4	程式管理.....	183
10 · 5	鍵盤操作.....	185
10 · 6	字元轉移.....	187
10 · 7	輸出格式指定與變換.....	190
10 · 8	專有名詞.....	192

習題

參考文獻

習題解答

第 1 章

FORTH的觀念

FORTH 雖然小巧玲瓏，却是一種很巧妙而且容易學習的語言。它能把複雜而麻煩的作業的電腦程式計劃，非常有效率地變換為明快的程式計劃。FORTH的效率良好。亦即，用 FORTH 所寫的程式能在電腦上高速地執行。更進一步，FORTH 容易讓使用者接近它。一旦學會了語言的基本概念，就不會讓使用者感到煩惱，毋寧說可幫助使用者設計程式。使用 FORTH 就宛如使用了電子計算機般容易。但是，藉 FORTH 所寫的程式，能夠使用最不容易學的語言，這一點能表現複雜的演算法。

1.1

桌上電子計算器與FORTH

我們日常所看到的所謂的桌上電子計算器，能輕輕鬆鬆地進行各種計算。既簡潔又便以攜帶，是一種用法簡便的工具，適合於某種間

12 FORTH概論

題。與此相對，所謂的電腦所適用的問題之領域便不相同，因此，其特徵也和桌上電子計算器有很大的差異。

電腦雖然有很大的自由度，同時通用性雖然也高，但是，其處理却非常複雜。實際上，以擴大人與機器的分界面，利用電腦所有的適用性與高速性為目的，乃開發了許多的程式計劃語言。FORTH程式計劃語言，是把電腦所有的功能與自由度，能夠以桌上電子計算器所有的簡便程度來利用，把這兩個概念結合起來的。

1.2 數式的記法

桌上電子計算器的算術演算的數學的表記法通常有下面所述兩種。

- 算術輸入記法 (algebraic entry notation)
- 逆波蘭記法

算術輸入記法是每輸入一次算術演算的指令，就執行一次算術演算。譬如說，對於

$$7 + 12 =$$

進行這種鍵輸入，則桌上電子計算機會顯示出結果 19。因此，譬如說，對於

$$2 + 3 \times 4 =$$

這種鍵輸入，結果可以獲得 20。也許各位讀者看了結果會嚇了一跳，但切莫忘記是加法那邊先被輸入，因而才先被執行這件事。計算在“累積器”進行，累積器又保持了應予顯示之值。

採用算術輸入記法的桌上電子計算機也容許使用括弧，譬如說，要求值 $(6 - 3) \times (10 - 5)$ 時，要按照下面所述鍵入。

$$(6 - 3) \times (10 - 5) =$$

進行這種鍵輸入，則結果變成 15。算術輸入記法之特徵，是算術演算子之記號被放置於數與數之間——譬如說， $2 + 2$ 。

表記算術式的另一種方法，有的是使用逆波蘭記法（RPN）。在這種情況下，在 2 個數值為對象的演算，是繼這 2 個數後面，有算術演算子。譬如說，

3 ENTER 2 +

是表記了 $3 + 2$ 的。RPN 的長處之一，就是進行複雜的計算時，按鍵的次數可以少一點。如果希望求 $(6 - 3) \times (10 - 5)$ ，則在 RPN 是按照下面所述鍵入的。

6 ENTER 3 - 10 ENTER 5 - ×

在逆波蘭記法，即類似一部桌上電子計算器所進行的計算步驟，而且也酷似許多加法機器⁽¹⁾（adding machine）。FORTH 的系統也採用逆波蘭記法，關於這一點將會在第 4 章解釋。了解逆波蘭記法的規則，可以不用看第 4 章。

1.3

使用時的環境

FORTH 是對話式的語言。說起來，FORTH 系統出現於畫面之同時，會立即和系統開始作交易。可是，FORTH 有下面所述兩個執行時的模式。

- 執行模式
- 定義模式

在執行模式，每一次輸入一連串的演算操作或參考程序，則每一次會從系統反應過來。在定義模式，輸入的一連串指令，是爲了往後的參考，執行時需要而被儲存起來。於是，FORTH 無論對單純的計算或者是複雜的程式，均容易使用。