

微 電 腦 程 式 語 言

蔡文能 譯
薛榮桃

LOREN P. MEISSNER
ELLIOTT I. ORGANICK
FEATURING STRUCTURED PROGRAMMING

FORTRAN

672107

微電腦程式語言



蔡文能 譯
薛榮桃



人民财产，加意爱护，
遵守纪律，按期归还。

南京工學院

分类号

73.87
79

登录号

672107



無線電出版社印行
九龍大南街三十五號

中東印刷廠承印
九龍天仙道十八號

香港・南洋各地書局均有代售

Y5 13/013 821

前 言

是什麼魅力使得福傳語言 (FORTRAN LANGUAGE) 在歷經二十多年之後仍然具有那麼堅強的實力呢？首先我們考慮到的就是福傳程式的可攜帶性 (Portability, 即易於改寫到另一個不同的計算機上)。福傳是世界性的語言, 不論在歐美或在任何一個國家, 不管是大型或迷你型的計算機上, 都有它的蹤跡。如果福傳不是具有一些特別的基本特性, 想要搶得語言之領導地位而且又持續了二十多年恐怕是辦不到的；尤其是福傳處理器 (FORTRAN Processor) 能夠產生很有效率的執行程式碼。

福傳這個名稱在許多國家中已經被沿用了二十多年, 但是主要的變化則是在 1960 年代初期福傳四版的引入。例如邏輯假若指述 (LOGICAL IF)、標名共用區 (Labeled Common) 就在當時被引入的。1966 年第一版的標準福傳四 (ANSI FORTRAN IV) 公佈之後, 一些不甚普遍的差異均被除去, 期使福傳語言標準化。然而, 福傳語言的標準是允許適當延伸的, 所以到了 1970 年代中期, 各種延伸的福傳語言琳琅滿目, 許多增加的特性已足可構成新的標準福傳。這種更新的標準福傳, 通稱 FORTRAN 77, 在 1977 年公佈草案, 由美國國家標準學會 (ANSI) 於 1978 年正式定名。儘管 ANSI 才公佈了福傳新版 ANSI X 3.9 — 1978, 仍然有許多集團在致力於更進一步的延伸和修正, 期使福傳語言能更上一層樓。

整個福傳發展的過程正好與本書的緣起不謀而合。早在 1963 年, “福傳初步” (A FORTRAN PRIMER) 的出版便是本書的前身

，隨後於1966年出版了“福傳四初階”，1974年福傳四第二版書應市。因此本書是一本延續性的書籍，以廣泛普遍的方式來探討福傳語言，不偏重於學術性。利用各種例子來說明福傳的基本規則以及結構化程式的概念。福傳初步（1963年版）一書中早已將這個觀念引進，使得程式設計邁入“結構化”的新紀元，這是第一本強調程式結構化之重要性的書籍，主要是針對演算法（ALGORITHM）的處理過程，透過清楚而結構化的流程形式對應到福傳程式。

本書承襲過去的傳統是兼顧工程人員及非工程人員的完美書籍。本書涵蓋了FORTRAN 77，具有字元型態（Character TYPE）資料、IF-THEN-ELSE結構、零次DO迴路（Zero-Trip Do Loop），自由格式（Free-Format）輸入、輸出以及其他許多特性，穿插交織，引導讀者深入淺出的去理解福傳特性。另外，本書還介紹讀者如何利用福傳77的特性與控制結構來改善福傳的程式結構。

本書第一章是專為沒有任何程式經驗的學生而寫的入門介紹。如果讀者已熟悉其它電腦語言，如BASIC、PASCAL、PL/I 或其他類似的語言，不妨跳到第二章開始看。

因為要對福傳提供一個廣泛綜合性的探討，所以本書包括了一個學期的程式設計課程。為了顧及整部書的完備性，所以不能介紹過於艱澀、偏僻的技巧。各章、節之間的編排均由簡而繁，有些太複雜的章節可以略去，不必花費太多的時間致延誤了後面的學習。例如，2.6節和2.7節（即雙倍精準數和複數資料型態）都可以暫時跳過。

還有，本書前後均偏重於福傳語言的介紹，至於其他語言的程式技巧則只作簡略的說明，因此，第4章和第5章第1節可以併在一起介紹。至於有格式的輸入輸出（FORMATTED INPUT/OUTPUT）可以延到有了基礎之後再作介紹，可以先用福傳77中的直接列表輸入輸出（自由格式的，List-Directed INPUT/OUTPUT）暫時取而代之，例如READ及PRINT。至於第九章則要比較深入的課程才有研讀的必要。（第九章討論檔案的輸入輸出處

理)

大部份的課程都可以利用本書從頭到尾的連貫結構來組成。不過第六章(基本輸入及輸出)和第八章(副程式)幾乎是兩個完全獨立的部份。所以,在介紹模式化的程式設計時(Modular Programming)可以略去第六章。

如果遇到不解之處,作者建議你不妨多參考幾本較為專精而深入的書籍,當然,這樣你以後也就更容易去學習後面更深的教材。

第叁部份(第10章,福傳的應用)包含了六個不同的實例,每個實例都有完整的程式流程圖、數據以及計算機執行後的結果。這些完整的例子可以做為前面各章的參考資料;也可以依照題後的建議加以延伸改進。

另外,本書還附了標準福傳ANSI X 3.9 — 1978 的摘要,編在附錄C。當然,除了ANSI標準FORTRAN外,在附錄裡還涵蓋了語言的一些基本指述(例如PAUSE,指定GOTO,INTRINSIC,SAVE,CLOSE以及INQUIRE等在這本書籍沒有提到的指述)。對於一般計算機系統以及流程圖的慣用法等也都在附錄中做了介紹。

雖然,我們對FORTRAN 77的用法已經有了許多很好的例子,但是,作者知道,在使用FORTRAN 77時仍將會碰到一些困難。因為對於在77內的新特性,我們仍然沒有很多真正具體而有意義的例子。也正因如此,我們需要使用本書的教師先生們的建議,俾能讓我們了解那些例子及習題你認為較為有用,以及那些方面的應用應該包含進去。成千的老師以及上萬的學生所累積的經驗將併入本書中,希望讀者諸君能繼續給本書指教,以充實本書的內容,增加本書的價值。

在此特別要感謝Bruce Martin對本書初稿的指正以及給予本書許多有價值的建議,同時也感謝其助手的幫忙。

作者 L.P.M. 謹識於
E.I.O.

Berkeley, California
Salt Lake City, Utah
(1979年9月)

譯 序

工欲善其事，必先利其器，電腦語言是人與計算機溝通的工具。要想學好電腦語言，一本好書是不可或缺的。坊間目前電腦方面的書的確很多，尤其是福傳語言的書，更是多如牛毛，可惜大都汗牛充棟，濫芋充數，不是資料不夠完備，就是實例說明不清，不適合於初學者研習。

這本FORTRAN 77是L. P. M. 和E. I. O.兩位先生的精心傑作，從有了FORTRAN之後不久就有了這本書，隨著福傳的演進，本書也不斷的修改，以配合潮流。全書一氣呵成，由淺而深，舉例詳盡，說明完整，實為學習福傳的一本好書。然而對大多數的國人而言，原文書並非良好的學習工具，在學習一個專業知識的同時還要分心去思索英文句型將降低學習興趣，學習效果也將事倍而功半。有鑑及此，所以譯者乃決定將之介紹給大眾，期能在這一大片“電腦化”的喊聲中能對於那些有興趣電腦的讀者有些許的幫助。

譯者謹識於新竹

1981年7月

目 錄

前言

譯序

第一篇	計算機與計算學 (Computers and Computing)	1
第一章	計算簡介 (An Introduction to Computing)	2
1.1	了解計算機	2
1.2	福傳語言與福傳處理器	13
1.3	用福傳設計程式	19
1.4	精簡化	29
第二篇	福傳語言 (The FORTRAN Language)	33
第二章	運算式和指定指述 (Expressions and Assignment statement)	34
2.1	變數和常數	34
2.2	運算式	42
2.3	函數	57
2.4	指定指述的使用	65
2.5	列陣及列陣的元素	70
2.6	倍準數	75
2.7	複數	77
第三章	選擇性程式結構 (program structures for Selection)	80
3.1	選擇性指述羣	80

3.2	邏輯運算式	108
3.3	程式之終止：STOP 和 END 指述	113
3.4	程式執行次序之外顯控制	113
第四章	重複執行結構 (Program Structures for Repetition)	123
4.1	計次之重複執行	123
4.2	巢狀迴路	145
4.3	複作迴路之終結指述	153
4.4	條件式之複式迴路	156
第五章	字元型資訊的使用法 (Using Information of Character Type)	164
5.1	字元運算與指定	164
5.2	字元列陣，子字串及運算式	170
5.3	有關字元處理的預定函數	181
第六章	基本之輸入與輸出 (Basic Input and Output)	187
6.1	輸入與輸出的觀念	188
6.2	格式之描述	198
第七章	宣告指述 (Declaration Statements)	234
7.1	型態宣告與列陣宣告	234
7.2	數據和參數宣告	245
7.3	等效宣告	250
7.4	程式指述的安排位置	258
第八章	子程式的定義及使用 (Defining and using Subprograms)	259
8.1	概述	259
8.2	次常規或外函數的定義	269
8.3	實引數與虛引數的聯繫	292
8.4	子程式的各個不同的進入點	311

8.5	共同區宣告	313
8.6	指述型函數	325
第九章	更進一階的輸入與輸出 (Advanced Input and Output)	329
9.1	控制名列	329
9.2	定格式的內部數據傳送	334
9.3	其他格式碼	336
9.4	檔案	340
9.5	檔的置位	342
第三篇	福傳應用 (FORTRAN Applications)	347
第十章	福傳之應用 (Fortran Applications)	348
10.1	模擬	349
10.2	表格排列，間接排列及合併	359
10.3	薪餉表之計算	372
10.4	利用印字機繪圖	392
10.5	日曆之製作	400
10.6	樹狀結構與圖形	408
附錄 A	一般計算機系統概觀	428
A.1	計算機組件	428
A.2	計算機系統組織	428
A.3	位置——位址之貯存裝置	430
A.4	計算機的基本運作及使用	431
A.5	計算機指令的性質	431
A.6	小型計算機的設計	432
A.7	參考書目	443

附錄B	流程圖的慣用法	444
附錄C	美國標準福傳 77 摘要	447
C.1	簡介	447
C.2	福傳語言	448
C.3	輸入與輸出	481
C.4	主程式與子程式	501
C.5	標準福傳 77 子集差異摘要	509
C.6	FORTARN 77 以及老標準福傳ANS × 3.9-1966間 的主要不同	510
部份	習題解答	514
索引	(英漢名詞對照)	521



第一篇 計算機與計算學
(COMPUTERS AND COMPUTING)

第一章 計算簡介

(AN INTRODUCTION TO COMPUTING)

1.1 了解計算機 (Understanding Computers)

如何學習使用計算機 (Computer) ?

儘管計算機是非常複雜的系統，但是却很容易被我們了解，為什麼呢？因為我們可以在內心中建立一個計算機的模式 (Model)。

舉例來說，汽車也是個很複雜的系統，我們可以藉一般汽車模式來了解各種汽車，易言之，了解了一般規則可以幫我們很快的學會駕駛任何一種車子。

這種如何開汽車的模式，從我們小時候就已經建立起來，所以我們已經很熟悉，無論是迷你型的或是巨無霸，更何況我們已經搭過那麼多汽車，玩過那麼多的玩具車。同樣的道理，只要我們熟悉一個好的計算機模式，我們就能輕而易舉的去了解它以及使用它。

在尋找一個好的計算機模式時，最重要的是要考慮到它能否代表各種不同的計算機。因為無論就表面結構或就內部構造來看，計算機間的差異都比汽車來得大。如果模式太偏重了某種特別型態的計算機，那就無法代表日後我們想了解的其他計算機了。更糟的是，很可能由於模式的過份詳細，導致沒有必要的複雜性，反而把本來我們想了的特性掩蓋住了。所以，我們最主要的目的是要建立一個理想化 (idealized) 的模式，而不是一個道道地地的仿製品 (faithful imitation)。

本書所採用的模式是非常理想化的。事實上，我們可以稱它為理

想計算機，雖然有點誇張，但也差不遠了。首先，我們要知道這理想計算機是什麼？它又如何工作？然後就可以很容易地解釋人如何將指令 (Instruction) 與資料 (Data) 傳送給計算機，而計算機又如何把有用的結果 (Result) 傳送回來。

福傳 (FORTRAN) 本身就是建立於理想計算機上的。本章的目的就是要藉此模式使大家對福傳語言有個基本的了解。以後各章再深入的探討福傳的優點，俾在各種有福傳語言的系統下，諸位能將福傳應用自如。

1.1.1 演算法 (Algorithms)

所有的計算機都是聽命行事的機器，我們的理想計算機也不例外。ALGORITHM這個字在前一世紀就已被廣泛的用來當作描述如何工作的專有名詞，尤其常用在描述計算機的工作上，我們稱它作演算法則或稱方法。首先，我們將先討論什麼是演算法，以及如何來表達工作。

演算法或方法可以說就是一連串的命令，用來一步一步地執行工作程序。幾乎任何一個複雜的程序，例如製造一個餅乾，配製一份藥方，或者編輯一本書的索引，都可以被析分為更簡單的一些基本小步驟 (Primitive Steps)。

工作描述須要詳細到什麼程度呢？那就要看執行這項工作的人之技術與能力而定。一個資深的麪包師、藥劑師或索引編輯人員只須要一份簡短的指示，然而生手或初學者却須要更明細煩長的說明。每一步驟的明細程度 (degree of detail) 則端視執行者對所謂的基本步驟之觀點而定。一個好的演算法應該是由難易適中的基本步驟組成，如此，使用它的人才能順利地將工作執行下去。

計算機演算法也是根據此一原則來細分工作。計算工作可以細分為指令，每一指令均基本到計算機能夠了解的程度。(這裏所謂的“了解”是指如果將某種指令告訴計算機後，計算機便能回給一個適當

4 程式語言 FORTRAN 77

的結果。) 在本書中，我們假設這部理想計算機所能“了解”的就是福傳程式語言的指述 (statements)。

舉例來說，以下便是一行典型的福傳指述，它可以用來描述演算法則。

$$AVG = (X + Y) / 2.0$$

這條指述是從一個程式中摘出來的，這個程式目的在求 X 與 Y 的平均值 AVG 。像這樣的一個指述代表演算法的一個基本步驟，在福傳程式中便是一個指定指述 (Assignment statement)；此為福傳語言五種指述中的一種。它含有一個運算式子 (Expression，或稱表示式，陳式)，在上例中即為

$$(X + Y) / 2.0$$

該式的值 (Value) 將被計算出來並被指定 (Assign) 為等號左邊變數 (Variable，例如 AVG) 的值。

1.1.2 資訊的貯存與處理 (Information Storage and Information Processing)

在所有計算機 (包括我們的理想計算機) 的特性中，最重要的就是貯存與處理資訊的能力。處理一項資訊的基本運作將會影響到貯存資訊的方式。我們這部理想計算機所能處理和貯存的資訊有兩種形式：數值資訊 (Numeric information) 和字元資訊 (Character information)。這兩種資訊的處理方式不同，當然貯存方式也不同。數值資訊的基本單位定數目 (Number)，其基本運作則為數學運算 (Arithmetic operation) 的基本運作或貯存單位則是字元 (Character，例如字母 letter，數字 digit，或特殊符號等)，這些不能拿來做數學運算的。

本章中對福傳的討論將只及於數值資訊的處理。雖然文字資訊的

處理比數值處理來得有趣，但是傳統的福傳設計在過去都只注重數值的計算，至於文字資訊則只是被拿來與數值資料一起印在結果上，以便增加其可讀性而已。不過在未來的福傳應用上將會逐漸地加強文字的處理。

貯存格 (Storage cells)

福傳程式所處理的數都貯存於同樣大小的格 (cell) 內。所謂的格，我們可將其視為就是我們理想計算機內的一個小盒子 (box)，內有一張寫著數的紙條。程式要處理的格都有名稱，反之，這個名稱就代表著這個格內的數。格的名稱就貼在盒上的標籤上，每個盒子有個蓋子，當要改變盒內之值時，蓋子便自動啓開。每個盒子還有個小窗，所以從小窗便能讀到裏面的數值而不必打開盒蓋。

真正計算機內部的貯存格當然不是真的像我們前面所描述的有窗小盒子，而是由微電子裝置組合而成，且具有高速度的運作功能。在此我們不準備把我們的理想計算機描述得太過於詳細；我們的目的在為讀者建立一個具有真正計算機特性的模型，但只是概念性的而且較容易理解。把每個貯存單位 (格) 看成一個帶有標籤、蓋子及小窗的盒子，則我們就很容易來了解這個模式的運作方式。如此一來，我們就不會被真正計算機的微電子學以及電子系統的行爲搞得心煩意亂了。

$(X + Y) / 2.0$

名稱 (在標籤上)

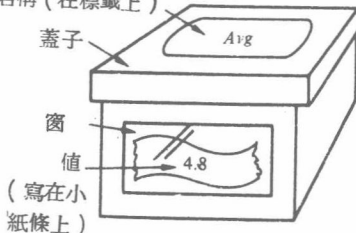


圖 1.1 貯存格可以被視為一個帶有小窗及蓋子的小盒子。盒上的標籤，標明了格式的名稱。值則寫在一張由窗戶看到的小紙條上。

1.1.3 變數與常數 (Variables and Constants)

6 程式語言FORTRAN 77

對於每一個貯存格，我們可以給它一個名稱 (Name)，叫做變數 (Variable)，在整個程式執行當中，格的名稱都不會改變，而其內容則隨時在變。其內容 (即格內的數目) 也就是變數的值 (Value)。貯存一個數到格內則等於指定一個值給變數。

當指定一個新值給變數之後，其原來的值便消失，再也沒有任何辦法能找回原來的內容。在這部理想計算機來說就是等於把蓋子打開放入新紙條時把舊紙條取出並且銷毀掉。

我們應該要了解貯存到格內以及由格內取出數值之間究竟有什麼差別。當貯存一個數時，蓋子先被打開，然後舊的值便被銷毀掉。另一方面，在需要的時候，存於格內的數，可以由盒上的窗子看到；盒蓋不打開而且其內之值並不改變。當一個值存到格內後，只要需要就可以隨時去讀取而不破壞原值。格內的內容將維持到新值被存入的時候。當一個新值存入後，老的數便消失而且格的內容也隨著改變。這種情形就像錄音機一樣，原先錄在錄音帶上的訊號可以保存著，而且也可以不斷的重複播放；只有當新的訊號錄入之後舊訊號才消失。計算機的格可以拿來跟錄音帶的部分相比較：在任何時間只有最後一次錄入的訊號能被找回來。

範例：請看這個福傳指述 (Fortran statement)

$$AVG = (X + Y) / 2.0$$

指述中用了三個變數：*Avg*，*X*，*Y*。執行這個指述將牽涉到三個貯存格，也就是如圖 1.2 中的三個有窗小盒子。三張標明名稱的標籤已經黏在盒子上，所以在執行的過程中這三個名稱是不會變的。另外，在*X*和*Y*兩盒中分別塞了一張寫有特定數值的小紙條。執行上述這條指述包括了到*X*，*Y*去取得其值，然後再計算

$$(X + Y) / 2.0$$