資訊科學譯叢 ②

FORTRAN 77
featuring structured programming

Loren P. Meissner
Elliott I. Qrganick

施亮光

計算機語言 FORTRAN 77

湯耀中 博士 校訂

五南圖書出版公司 印行

TP312
9029

S

中文书库

資料科學譯叢②

FORTRAN 77

featuing structured programming

Loren P. Meissner
Elliott I. Qrganick

計算機語言 FORTRAN 77

施亮光

湯耀中 博士 校訂

五南圖書出版公司 印行

TP312/9029

* 671168*

序

自1966 年美國國家標準局製定了FORTRAN Ⅳ 以來，FORTRAN 語言就成了世界最通用的電腦語言之一，它廣為學術教育機構所接受，幾乎所有學習計算機的學生都由它著手，但學過FORTRAN語言的人都可以發現，FORTRAN Ⅳ 有一些重大的缺陷，比如缺乏 IF…THEN…ELSE 的結構，使寫出的程式不易瞭解，也過於龐大。同時FORTRAN Ⅳ的輸入和輸出指述必須格式化，使用時頗多不便。它對於檔案（file）的運用也不夠方便，因此 FORTRAN 漸漸失去原有優勢，有鑑於此，美國國家標準局於1977年製定了 FORTRAN 新版，即為FORTRAN Ⅴ，也就是 FORTRAN 77 ， FORTRAN 77針對上述缺失加以改進，使其更 powerful ，因此也漸為商業界所運用，在此書中可見到 FORTRAN 商業運用的例子。

此書參考 FORTRAN 77 FEATURING

STRUCTURED PROGRAMING（LOREN. P. MEISSNER）所編寫，為國立交通大學所採用的教材書，亦為最新的一本 FORTRAN 語言，書中除了介紹 FORTRAN 語言外，並對計算機內部操作和 FORTRAN語言的執行有詳細的描述，使讀者能知其所以然。書中誤繆，在所難免，希望各位先進，不吝指正。

編著者 施亮光

70 年雙十節

於交通大學

計算機語言 FORTRAN 77

目　次

此为试读,需要完整PDF请访问: www.ertongbook.com

第1章 概論

在本章中，我們將簡單地介紹一下計算機的演進史，計算機的基本結構；以及在正式設計FORTRAN程式前所應作的準備工作。

1-1 計算機簡介

1.1.1 計算機的種類

就計算機的設計與作用而言，大體上可以分爲三類：數位計算機（digital computer），類比計算機（analog computer）以及混合型計算機（hybrid computer）。

數位計算機精確度較高而速度略慢，類比計算機則速度較快而精確度較低，混合型計算機爲前兩型計算機之結合，是故精確度與速度均介於二者之間。

1.1.2 計算機的演進

最早是美國數學家Charles Babbage於1833年在紙上設計出數位計算機，當時稱爲解析引擎（Analytical Engine）。直到1940年，此理論爲哈佛大學之Howard

Aiken用來設計一套自動程序控制計算器。而後，在1944年由IBM公司完成。到了第二次大戰期間，美國陸軍亦着手發展電子數值積分器和計算器，於1946年完成,是爲計算機的祖先。

由此開始，計算機與人類生活的關係愈來愈密切。於1946年迄今,計算機演進大致可分爲：

1. 1958年以前：計算機係由眞空管爲其主要構成要件，資料的儲存量有限，以磁帶（magnetic tape）爲主要儲存記憶介體，計算速度相當緩慢，以千分之一秒爲單位，是爲第一代計算機。

2. 1958年至1964年：由於電晶體（transistor）之誕生，不僅使計算機體積縮小不少，並使計算機速度增快至以百萬分之一秒爲單位，利用磁帶與磁碟（magnetic disk）作儲存介體（storage medium），使資料儲存量亦大爲增加，是爲第二代計算機。

3. 1964年至1970年：由於積體電路（integral circuit）之發展成功，使計算機之體積縮小成第一代計算機之四十分之一。計算速度也加快至以十億分之一秒（nano second）爲單位。並增加了許多種輸出與輸入介體，是爲第三代計算機。

4. 1970年迄今：由於大型積體電路（ＬＳＩ）之產生，使計算機速度更快，也令計算機體積更形縮小，是爲第四代計算機。

1.1.3 計算機的結構

計算機系統大體上可分爲四個部分：中央處理單元（central processing unit 簡稱爲CPU），記憶單元（memory unit），以及輸入單元（input unit），輸出單元（output unit）,計算機主要是由前兩部分構成。其結構圖如1.1所示。

1. 中央處理單元：中央處理單元可分爲兩部門：一爲控制單元（control unit），根據我們所設計程式中的指令來控制計算機的運轉。一爲計算邏輯單元（arithemetic & logic unit）,用來執行算術與邏輯運算。

2. 記憶單元：用來儲存資料，程式以及程式執行中的暫時結果與最後的答案。

3. 輸入單元：將所設計的程式利用輸入介體輸入計算機。輸入介體有卡片（punch

card），紙帶（paper tape），磁帶（magnetic tape），磁碟（magnetic disk），終端機（terminal）……等。

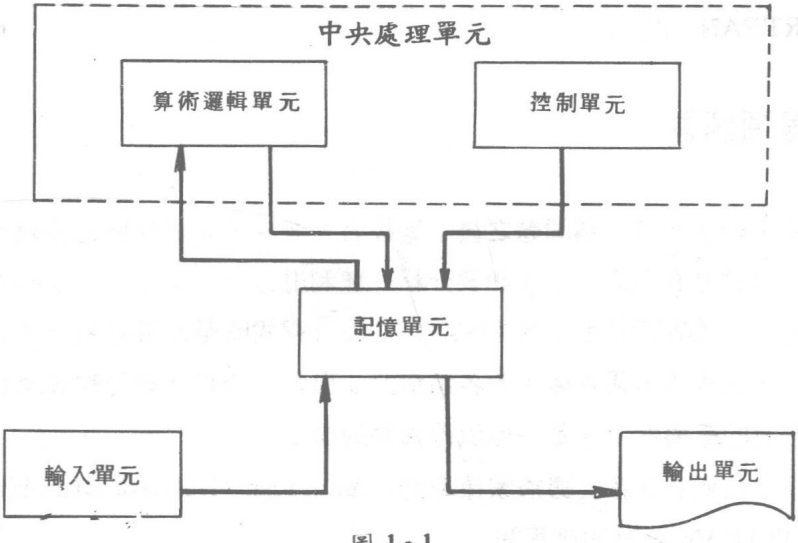


圖 1·1

在圖 1.1 中，程式計劃首先經由輸入單元傳至記憶單元中。然後控制單元會由記憶單元中取出控制指令，根據這些指令來控制記憶單元，與算術邏輯單元作用。算術邏輯單元，則根據控制單位之命令自記憶單元中取出資料來處理，同時將結果存入記憶單元中，輸出單元則等候控制單元的信號，自記憶單元中取出結果來作輸出。

在這些操作中，中央處理單元與記憶單元是電磁動作，速度極快。而輸入與輸出單元却是機械動作，速度甚緩。

1-2 FORTRAN簡介

FORTRAN一詞係 FORmula TRANslator二字的簡縮，是一種最普遍的計算機語言。主要運用在科學與工程方面，幾乎每一類型的計算機都具備了這種語言，因此大多數都以 FORTRAN 爲學生的初習語言。

早在 1957 年由 IBM公司在 IBM－704 型計算機上設計出來的，是爲 FORTRAN I。而後每在 FORTRAN 基本指令群上，作一些修改，增進一些功能，便產生了一新

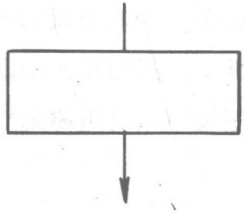
的版本。直到1966年，美國國家標準局（American National Standard Institute簡稱ANSI）製定了標準 FORTRAN IV ，廣爲一般學者接受，一直流行到今。1977年美國國家標準局又製訂了 FORTRAN新版，到1978年定義完全，是爲 FORTRAN V俗稱 FORTRAN 77 。

1-3 流程圖語言

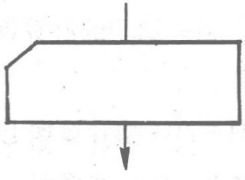
當我們用演繹法解決一個問題之後，緊接著要考慮的是如何將此演繹法轉換成計算機所能接受的語言？在這之前，我們要介紹一種利用方塊（block）與流綫(flowline)來表示的語言——流程圖語言。流程圖語言在設計程式時是非常便利而常用的工具，它可以明確地表示出程式邏輯會產生的各種情況。但是，今日一般的程式設計者卻往往忽略了流程圖語言的重要性，這是一件值得警惕的事。

在本節中，我們將介紹美國國家標準局（American National Standards Institue）所制定 FORTRAN 語言的流程圖。

1.處理盒（processing）：爲一長方形的盒子，此盒子中通常有一個或多個指定陳述（assignment statement）。用來表示程式的處理過程。

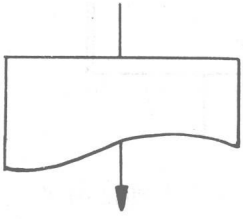


2.卡片輸入（punch card input）：用來表示讀入由卡片輸入的資料，通常將所需要的輸入變數寫在盒子中。

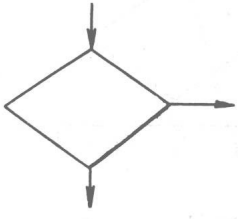


此为试读,需要完整PDF请访问: www.ertongbook.com

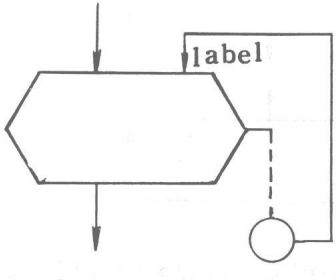
3. 文件輸出（ document output ）：用來表示資料文件的輸出，輸出介體可爲列表機（ line printer ），終端機（ display terminal ）或電傳打字機（ typewrite ），……等。



4. 決定盒（ decision ）：這種菱形的盒子用來表示狀態控制指述，通常是代表邏輯 IF 指述，有一條流綫進入，而有兩條流綫流出，代表不同的決定。

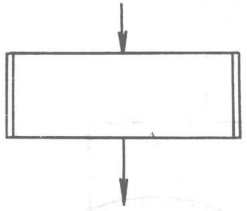


5. 準備盒（ preparation ）：通常用來表示 FORTRAN 中的 DO 指述，在盒子的右上方要記錄結束指述的標號（ labe ），在盒子中，有一指標（ index ）以及它的始起值（ inital value ），增值（ increment ），結束值（ terminal value ），若是指標未超過結束值，則執行右邊的流綫，否則則由左下方的流綫離開此 DO 指述。

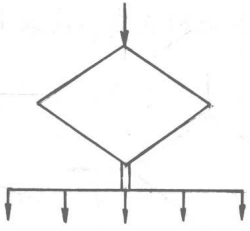


6. 副程式呼叫盒：用來作副程式呼叫（ subroutine call ）之用，副程式的名字以及參數（ parameter ）均應書寫於盒子中。至於函數程式（ function ）的呼叫，並不需

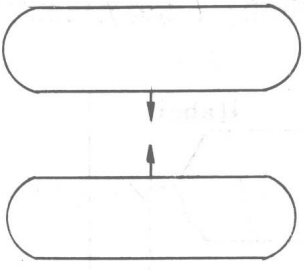
要用到此呼叫盒，只要列在處現盒中，如同指定指述一般即可。



7.多重決定盒（mutiple decision）：通常用來表示 FORTRAN 中 computed Go To 指數，可以有許多流出綫。



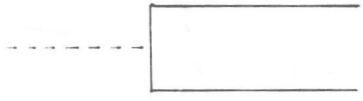
8.終端點（terminal）：終端點可分爲兩種，一是開始點，僅有一流出綫，沒有流入綫。一是結束點，僅有一流入綫，沒有流出綫。開始點可用來表示程式或副程式的開始。結束點則用來表示程式或副程式的結束，如同 END 和 RETURN 指述一般。



9.流線（flowline）：用來表示程式進行的方向與順序。

10. **連結子**（ connector ）：在連結子中間，通常放有一標號（ **label** ），用來表示 CONTINUE 指述。或是用來連結下一個要執行的指述流程。

11. **註解**（ annotation ）：通常用來表示程式中的宣稱指述（ **declaration statement** ）。



第2章

表示式和指定指述

2-1 變數和常數（Variable and Constants）

在FORTRAN指述中有兩種基本指述：指定指述（assignment statement）和READ 指述。READ指述將在第六章詳細討論；現在先討論「指定指述」。

所謂指定指述，包括以等號分開的兩部份：等號左邊爲變數而右邊爲表示式（以後簡稱爲表式）。

變數（Variable）＝表式（Expression）

在FORTRAN語言中，一個變數表示在計算機中的一個儲存單位。因此指定指述表示在儲存單位中存放一個值。而計算機對於一個指定指述的操作方式是先決定等號右邊表式的值，再將此值存入等號左邊的變數所代表的儲存單位中。

在等號右邊的表式，可以是非常簡單的形式，例如一個常數。

［例1］

```
A = 3.172                PI = 3.14159
MANY = 346021            N123 = 456
E = 2.71828              X = 4.5
K = 721                  LOVE = 2
I = 721                  THETA = 3.14159
```

上例中的常數有兩種形式：整常數（integer constant）和實常數(real constant）。在FORTRAN 語言中一個常數是不可以有小數點，而一個實常數則必須有小數點。一個表式中若只包含一個整常數則稱爲整數表式，若只包含一個實常數則稱爲實數表式。

在程式中每一個變數皆有型態（整數或實數），而在程式中的變數除非先宣稱（declaration 一見第七章）否則都依一個設定的內隱型態規則：若一變數以字母I，J，K，L，M或N 爲開頭，則爲一個整變數，而實變數則是以其他字母開頭的變數。在例1中，變數型態須和等號右邊的表式型態符合。

[例2]

```
X = Y
Y = X
THETA = PI
K = I
LOVE = MANY
HERE = THERE
```

上例中首先須注意等號兩邊的型態（整數或實數）必須相同，否則指述是錯誤的。上例中指定指述的執行方式，是將等號右邊表示所表示的儲存單位中的值放入等號左邊來，變數所表示的儲存單位中，若變數所代表的儲存單位中原來存有值，則新值將把舊值蓋掉（表示本身的值不變）。如上例中X＝Y，Y＝X，表示先將X中的值代以Y中的值，再將Y中的值代以X中的值（新值），所以這兩個指述執行完畢後，X和Y皆存有原來Y的值。

此为试读,需要完整PDF请访问: www.ertongbook.com