

Understanding BASIC Program BASIC 程序入門



微型電腦叢書



周 啟編著 · 萬里書店出版

微型電腦 叢書

- 專家撰寫、翻譯，文筆簡潔，插圖精美。
- 資料新。部分書向外國出版社購買中文版權，內容切合實際。
- 編排醒目，印刷質量高，遠非一般翻版本可比。

IBM PC 最佳商業軟件指南

Richard C. Dorf 著 周其節譯 24.00

微型電腦數據庫入門

招兆鏗 方錦城編著 25.00

正確使用你的微型電腦

周其節編著 22.00

如何選擇小型商用電腦

Mark Birnbaum & John Sickman 著 18.00

APPLE 磁碟操作系統入門

方錦城 招兆鏗編著 28.00

BASIC 商業子程序

招兆鏗 方錦城譯校 25.00

APPLE II 操作入門

方錦城 招兆鏗編著 30.00

APPLE II 中級技術

金韶寧編著 即出



萬里書店出版

香港鰗魚涌芬尼街2號D

前 言

關於BASIC的書已經有幾百本了，爲什麼還要寫這一本？簡單地說，因爲讀者不同了。從前只有少數有機會接觸電腦的人才能使用像BASIC這樣的程序語言。那時程序員很少。現在情況不同了。個人電腦的普及使BASIC已成爲最廣泛使用和最易接受的電腦語言。現在大多數用戶只有較少的技術基礎。他們在娛樂、教育、商業或不同的專業方面使用電腦。

這本書是爲這些新用戶而寫的。它是一本初學者的書，不要求讀者有任何技術知識。它可以幫助讀者很快入門，學會BASIC初階。

作者相信，所有電腦初學者都想學會使用BASIC寫自己的程序。他們希望有一種簡單快捷的學習方法。這本書正是採取這種方法，它盡可能使讀者學得容易些和生動些。此外，這本書的目的是在幾小時內使你學到BASIC的基本知識。在一小時內你便可以寫出自己的第一個BASIC程序。幾小時之後，你應該能夠開始編寫有用的和實際的程序了。

作爲一本自學的書，它是循序漸進的。讀者應該從頭依次讀下去。在接着閱讀下一章之前，應該先理解每一章的內容。每章結尾都附有練習，幫助讀者掌握新的技巧。請盡可能做多些練習。書末附有部份練習的答案。

如果你有電腦，請試做全部程序。爲了真正學會和記住，必須實踐和實驗。這本書將幫助你得到必要的入門知識，但請記住，沒有什麼可以代替實踐。

這本書的目的是盡快使你掌握用BASIC編寫程序。因此，它只能介紹最重要的內容，而不能包括BASIC的全部特點或概念。

第1章解釋什麼是電腦語言，並介紹一些概念：電腦、解釋程序、

程序、指令等等。

第 2 章說明你怎樣和電腦交換信息，怎樣使用鍵盤和顯示器。你將學會打入初步的BASIC程序，並執行它們。

第 3 章說明怎樣用BASIC進行計算。

第 4 章幫助你編寫可以重複使用的程序，你將知道怎樣正確地和有效地使用變量。

第 5 章說明怎樣使你的程序清晰和易讀。

第 6 章說明怎樣用邏輯和數值作出複雜的決策。

第 7 章解釋怎樣用程序循環自動做重複的工作。

第 8 章說明設計程序的正確方法，從設計算法到編寫文件——包括流程圖的設計。

第 9 章通過實例說明怎樣運用上述各種概念。

第10章幫助你進一步提高編程技巧。

附錄A 是部份練習的答案。附錄B 列出通常的保留字。

目 次

前 言	1
第 1 章 關於BASIC	7
編寫程序	7
BASIC解釋程序	9
什麼是BASIC ?	11
哪一種BASIC ?	12
電 腦	13
電腦與語法	17
第 2 章 和電腦通信息	18
鍵盤操作	18
使用 BASIC	23
一個較長的程序	29
摘 要	33
練 習	33
第 3 章 用 BASIC 計算	35
打印數字	35
科學表示	36
做算術	37
打印格式	39
應用例	41
摘 要	42
練 習	42
第 4 章 存儲數值和使用變量	44
INPUT 語句	44

	兩種變量.....	47
	給變量賦值 (LET 語句)	53
	變量計數器方法.....	58
	摘 要.....	60
	練 習.....	61
第 5 章	編寫清晰的程序.....	62
	REM 語句.....	62
	在一行上寫多條語句.....	63
	使用空白.....	64
	改善顯示.....	66
	簡便輸入語句.....	66
	選擇變量名稱.....	67
	正確地編行號.....	68
	摘 要.....	70
	練 習.....	70
第 6 章	作出決定.....	71
	IF 語句.....	71
	算術練習.....	77
	GOTO 語句.....	80
	IF 語句 (續)	82
	1 的計數.....	83
	算術練習 (續)	85
	核對輸入值.....	85
	哩數轉換.....	86
	出生日期.....	86
	摘 要.....	88
	練 習.....	88
第 7 章	自動循環.....	90
	IF/GOTO 方法.....	90
	FOR...NEXT 語句.....	93
	前 N 個整數的和.....	96

	數值表.....	97
	星號行.....	98
	高級的循環.....	99
	另一些特色.....	102
	摘 要.....	102
	練 習.....	102
第 8 章	編寫程序	104
	設計算法.....	104
	畫流程圖.....	106
	編 碼.....	115
	糾 錯.....	116
	編寫文件.....	117
	摘 要.....	118
	練 習.....	119
第 9 章	實例：公制轉換	120
	設計算法.....	120
	流程圖.....	121
	編寫程序.....	127
	試 驗.....	135
	摘 要.....	139
	練 習.....	139
第10章	下一步	140
	你可以用 BASIC做些什麼？.....	140
	改善技巧.....	140
	更多的 BASIC.....	141
	結 論.....	144
附 錄		145
	A. 部份練習答案.....	145
	B. 常用的BASIC保留字.....	150

—— 第 1 章 關於 BASIC ——

前言中說“在一小時內你便可以寫出自己的第一個BASIC程序”。爲什麼還要用這一章來介紹一些概念和定義？這豈不浪費時間嗎？不，相反，我們的目的是學會和記住知識，而真正的學會需要深刻的理解。這一章介紹的知識將幫助你更好地理解什麼叫做編程序，BASIC 程序是如何執行的，以及有關電腦的術語。

在開始編寫第一個程序之前，你應該知道一些重要的定義和概念。當你理解了這些術語之後，我們可以用簡單而準確的方式敘述有關的內容，而你也可以容易地接受它們。因此，請仔細閱讀這一章，這樣你就能真正理解你正在做些什麼事情，而不是簡單地敲擊着鍵盤。

首先，我們學習怎樣把指令送入電腦，這叫做編寫程序。然後我們說明爲什麼需要有程序語言，例如BASIC。接着我們將討論BASIC解釋程序是什麼，並介紹BASIC的歷史，它的術語和它的應用。最後我們考察電腦系統有那些部份，並學習有關這些部份的一些技術名詞。

編寫程序

電腦是用來處理信息的，包括文字的和數字的信息。例如，你可以使電腦在屏幕上顯示字和句——這叫做**文本處理**(text processing)。你也可以把安士計算的重量轉換爲以克計算的重量——這叫做**數字處理**(numerical processing)。爲了使電腦完成這種處理，必須以它能理解的格式或“語言”向它發出指令。每個電腦只能“理解”(即認識和執行)爲數不多的各種指令(例如幾百條)。

電腦能直接理解的指令叫做機器語言(machine language)指令。這些指令以二進制(binary)格式(即0和1的組合)存儲在電腦的存儲器裏。每個0或1叫做一位(bit),8位構成一個字節(byte)。

能完成某些有用功能的一系列指令叫做程序(program)。電腦依次執行程序的每條指令。不幸,用機器語言(即二進制形式)編寫電腦的指令是費時而使人厭煩的事情。

最好是,我們能夠用日常的語言(例如用英語)來向電腦發出命令,並使它執行這些命令。但這是不可能的,因為電腦不能理解任何一種日常的語言——不管是口述的或寫出的。理由很簡單:電腦嚴格而準確地執行命令。它遵守邏輯,並且是精確的。它需要明確的、毫不含糊的指令。指令的順序和格式要正規。常用的語言的問題在於它本身——語句常常是含糊的,它們的含意依話題和場合而不同,例如面部表情和姿態。這種通話方式是不能被電腦理解的。

即使謹慎地書寫的英語,對電腦來說也是不夠精確的。例如,你不可以對帶電腦的機器人說:“到廚房去煮一隻雞蛋”,就希望有結果,除非機器人已編了程序,知道到廚房去要怎樣走法。能夠在日常環境中工作之前,機器人必須受到訓練(或編程序)。甚至即使某個機器人受了訓練知道怎樣走進你的廚房,在朋友家的廚房中它就未必能勝任,因為裏面的東西可能擺佈在不同的地方。請記住,和電腦通信必須是清楚、精確和毫不含糊的。

正是由於這個緣故,人們創造了簡潔的“語言”來和電腦通信。前面講過,二進制語言(又叫做機器語言)是最容易被電腦理解的語言。不過對人們來說,這種語言是難以應用的。因此人們發明了其他的語言來幫助通信。這些語言類似英語,叫做高級語言。(High-level Languages)。

爲了有效而清楚地和電腦通信,只使用了爲數不多的英語字作爲規定的命令。此外,向電腦規定指令的句子或語句(statement)一定要遵循嚴格的文法規則,這叫做語言的語法(syntax)。有限的詞匯和語法構成編程語言(programming language)。BASIC就是這樣的一種語言。

簡言之,編程語言是一些規則(語法)和字及符號(詞匯)的集

合，你可以用它以能準確地理解的格式向電腦發出指令。一系列這些指令就叫做一個程序。

下面是一個例子。假如我們要做 2 加 2 的算術，並把結果顯示出來。使用 BASIC 寫出

```
1 R = 2 + 2
```

```
2 PRINT R
```

這裏 R 表示“結果”。

不過請等一等。上面我們說過，電腦能直接理解的唯一的語言是機器語言，而我們現在向電腦發出的指令却用接近英語的語言，這豈不矛盾嗎？

這並不矛盾。當然，電腦本身不能直接理解 BASIC 或其他高級語言（使用像英語的語句的語言）。因此，為了能被電腦理解，必須用一種特殊的程序（人們恰當地把它叫做解釋程序 interpreter）來翻譯用高級語言（例如 BASIC）寫的程序。換言之，通過解釋程序，你可用 BASIC 和電腦對話。因此，為了執行 BASIC 程序，你的電腦必須有 BASIC 解釋程序。讓我們看看解釋程序是幹什麼的。

BASIC 解釋程序

BASIC 解釋程序讀入你從鍵盤打入的每條指令，並自動把它翻譯成一系列機器語言指令，使電腦能理解和執行它。你完全不會察覺到這個過程（這發生在電腦內部）。在電腦啟動解釋程序後，就實際意義來說，你的電腦能夠講 BASIC。如果提供合適的解釋程序的話，你的電腦也可以講其他編程語言。

在你的電腦上有幾種 BASIC 解釋程序可供使用。這裏我們將說明兩類主要的：駐留的（resident）和非駐留的（non-resident）。

大多數小型電腦都提供駐留的 BASIC 解釋程序。它叫做駐留的，是因為它永久性地駐留在電腦的存儲器中。當接通電腦電源後，立即就能使用它，或者發一個例如 B 或 BASIC 這樣的命令，它就可供使用。當看到 BASIC 的符號（或提示符（>））在屏幕上出現時，你就知

道電腦已準備執行你給的BASIC指令。

駐留解釋程序一般有一個缺點：它只能提供 BASIC 的最小版本。由於駐留解釋程序是“駐在”電腦的固定存儲器中，它的容量就要盡可能小，因為電腦擁有的存儲器總數是有限的。電腦存儲器必須容納程序，包括 BASIC 解釋程序，而且還要提供足夠的空間供計算、系統管理和存放被處理的數據之用。這些對空間的要求限制了駐留程序的容量，因而限制其複雜程度。在有較少存儲器的電腦中，駐留解釋程序通常是“微型BASIC”，它限制了能做的工作範圍。

不過任何駐留 BASIC 已足以用來學習如何用BASIC編程序，至少對這本書的目的來說是夠用了。以後當學會編寫較複雜的程序後，也許你希望有更多的功能。你將希望有非駐留解釋程序。

爲了擁有更多的特點，解釋程序就要做得複雜些，因而要佔用較大的容量。因此就把它存放在大容量的存儲裝置上，例如卡式磁帶或軟磁碟，這樣就不至於長期佔用大量可用的存儲器。在大多數小型電腦上，爲了容納較大的解釋程序，通常需要購買額外的存儲器。

較大功能的解釋程序可提供叫做詳盡(full)BASIC、擴展BASIC、浮點(floating-point)BASIC或高級BASIC的版本，各種名稱依製造廠而不同。根據所用的磁性存儲媒體，又叫做卡式BASIC或磁碟BASIC。通常用駐留BASIC寫的程序不必改動就能由詳盡BASIC執行。此時，這兩種版本叫做向上兼容的(upwards compatible)。

使用卡式或磁碟 BASIC 時，首先要將解釋程序從卡式磁帶或軟磁碟傳送到電腦的存儲器中。這叫做裝入解釋程序。然後還要按規定給一個指定的命令，例如BASIC，以便啓動詳盡BASIC解釋程序。然後它就顯示一個提示符。一般它的提示符和駐留 BASIC 的不同，以免產生混淆。到此時你才可以輸入BASIC指令。

現在來說明什麼是 BASIC，人們是怎樣創造它的，以及一些慣用語。

什麼是BASIC?

創造高級語言是爲了方便用戶向電腦發出指令，亦即給它編程序。多年來已創造了上百種編程序語言。

早期的電腦主要用於科學計算，因此早期的編程序語言是爲了方便數字計算。所以語言的祖師爺FORTRAN（“FORMula TRANslator”公式翻譯器的縮寫）是專爲數字計算而設計的。不過FORTRAN有很多缺點。後來創造了很多新的語言。BASIC就是其中之一。還有COBOL、APL和Pascal等語言也變得廣爲採用了。

BASIC的創立標誌着重大的突破。BASIC是爲了簡單和易學而設計的。此外它是對話式（interactive）的。下面解釋這是什麼意思。

BASIC是英文“**B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode”的縮寫，意思是初學者通用符號指令碼。它是1964年由美國Dartmouth學院的John Kemeny和Thomas Kurtz創造的。作者的意圖是設計一種初學者容易使用的語言。他們是成功的。目前BASIC是最容易學會的編程語言之一。

由於BASIC設計爲對話式的，——事實上它是最早的對話式語言——用戶可以在終端和程序對話，而不必像用其他較早的語言那樣把大批IBM穿孔卡片送入機器。開始時是在Dartmouth學院的GE 225分時系統上執行BASIC的。在校園各處都設有終端，很多用戶可同時使用電腦。

BASIC很快就成功了。General Electric（GE）立即決定把它用於商業。1967年Kemeny和Kurtz發表了第一本關於BASIC的書。以後Hewlett Packard（HP）和Digital Equipment Corporation（DEC）決定使BASIC能在它們的大多數電腦上使用。

和FORTRAN等語言比較，BASIC有兩個主要優點：1. 對於用戶：BASIC是最容易學習的語言，尤其是對於初學者；2. 對於製造廠：BASIC是在電腦上最容易配備的。由於語言簡單，解釋程序也就簡單，只要求少量的存儲器。

還有第三個因素使BASIC獲得成功：廉價微電腦的發展。在70年代末期，當微電腦變得廣爲流行時，BASIC就成爲在這些小型電腦上

通用的編程語言。由於BASIC簡化版本的解釋程序只需用4 K (4,096字節) 的存儲器，即使最小的電腦也能容納它。(請記住，駐留BASIC是指其解釋程序存放在電腦的固定存儲器的。) 近期較大的電腦有較多的存儲單元 (64K 或更多一些——1 K 有1,024 字節)，因而可能配備功能較大的BASIC版本。

現在幾乎在所有電腦上都使用BASIC。多年來製造廠把一些擴展功能和“特點”加到BASIC上，因此目前BASIC大概是最不標準化的電腦語言，沒有兩種BASIC是完全相同的。事實上BASIC已不再是單種語言，而是一種語言系列。雖然提出過很多標準，但都失敗了。近期內大概也沒有那一種會獲得成功。這是不是說在每種電腦你都得重新學習BASIC？不一定。當你懂得BASIC所有版本的共同要點之後，很容易就學會每種版本增加的內容。所有BASIC基本上有相同的主要指令。這本書將使你學會這些指令。

哪一種BASIC？

也許你的電腦可提供幾種不同的BASIC版本。上面我們講述了BASIC的兩種主要類型：駐留BASIC (通常是微型BASIC) 和非駐留BASIC (詳盡BASIC或擴展BASIC)。現在來看看它們之間的一些差別。

“微型”或“小型”BASIC的特點較少，也不及“詳盡”或“擴展”BASIC那麼方便使用。“微型BASIC”的常見局限性是它們只能進行整數運算。就是說，它不能處理分數。這種BASIC版本又叫做“整數BASIC”。相反，也能處理分數的BASIC改善版本叫做“浮點BASIC”。如果你打算做計算工作，這個特點是十分需要的。

微型BASIC一般在卡式磁帶上記錄信息，而不在軟磁碟上。並且它一般只存儲程序，而不存儲數據——因此它不處理“文件”。通常它只能處理作為程序的一部分的信息，或由鍵盤提供的信息。相反，使用磁碟的詳盡BASIC在處理信息 (文件) 和程序方面有較大的方便和功能。

有些製造廠提供更高級的該廠家特有的BASIC版本。通常標記為“擴展BASIC”，以提醒用戶注意。這意味着能向熟練的程序員提供功能更大的資源。不過，使用這些特有的資源，通常會使BASIC程序不能與其他BASIC解釋程序兼容。

這本書用的BASIC是一種通用的微型BASIC，因此你學到的技巧對BASIC各種版本都適用。不過，我們在課文中也將指出常見的變型和擴展。

總之，你不必擔心這裏所用的BASIC版本。以後如果你想寫較複雜的程序，你可以查閱所用BASIC版本的參考手冊。此外，本書最後一章將介紹BASIC的若干高級的特點的情況。

現在，我們對編程語言特別是BASIC已了解得較多了。讓我們進一步了解電腦，以及它是如何處理信息的。

電 腦

電腦處理信息，並通過鍵盤和屏幕或者再加上打印機和你通信。鍵盤（ keyboard ）用來向電腦輸送信息。每次壓下一個鍵時，與此鍵對應的電子編碼就被送入電腦中。電腦能確認它，作出反應，或不理會它。鍵盤就是你的輸入裝置，它向電腦提供信息。

屏幕或CRT（陰極射線管的縮寫）顯示器展示出程序產生的信息。通常你在鍵盤壓下的每個字符都立即在屏幕上出現。它首先被送到電腦，然後“反射”在屏幕上。一般鍵盤與屏幕之間沒有直接的聯系。所有通信都經過電腦。下面來說明這一點。

根據系統設計的不同，可以在單獨的機殼內裝置電腦本體，或者把它和鍵盤、屏幕和／或磁碟單元裝在一起。不管怎樣裝配，電腦本體包括處理單元（中央處理器）、存儲器和若干接口（與打印機或其他器件連接的電子線路）。下面介紹這三個部份。

中央處理器（CPU）每次一條地從存儲器取出程序的指令，並執行它。CPU只有少數元件。它們叫做集成電路或“芯片”。所有微電腦使用一塊微處理器芯片作為CPU的主要元件。下圖表示典型的芯片。

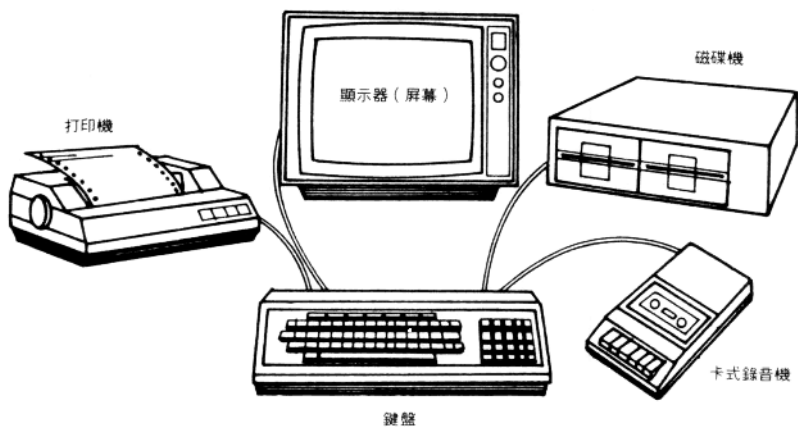
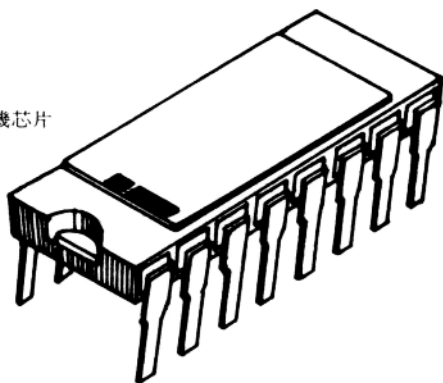


圖1-1 微型電腦系統的基本裝置



圖1-2 採用整體式設計的微型電腦系統

圖1-3 微處理機芯片



存儲器存放着程序和所有信息。這些信息是在執行程序時由程序處理、讀出或產生的。爲了執行某個程序，先要把它放在電腦的存儲器中。例如，如果程序原來是在卡式磁帶或軟磁碟上存放着的，就必須把它傳送到電腦的存儲器中，這叫做裝入程序。電腦必須有足夠多的存儲器，以便容納下最長的程序和它處理的數據。

電腦中有兩種存儲器：R O M和R A M。通常你可以用來存放程序的是R A M，即隨機存取存儲器。R A M是讀／寫存儲器，可以把信息寫入或從R A M讀出來。外表上R A M和圖示的微處理器相似，只是芯片的內容不同。不幸，目前的技術狀況只能把它做成可揮發性的：一旦切斷電源，R A M的內容就消失了。如果你想把程序保存下來，在下班時就要把它存放到非揮發性的媒體上，例如卡式磁帶或軟磁碟，原因就在這裏。還要記住，在執行BASIC程序之前，必須把BASIC的一個解釋程序存放在存儲器裏（在RAM或ROM）。

R O M是只讀存儲器的縮寫。這種存儲器的內容由製造廠用程序裝入，不能再改變它。它是非揮發性的，不會被抹掉。通常它包含一個駐留BASIC解釋程序和一個特別的叫作監控器的程序。電腦通電後，就需要監控程序和電腦通信。

如果在R O M中什麼也沒有裝入，當你在鍵盤按鍵時電腦就不知道要做什麼事情。最低限度R O M要包含一個監控器。監控程序檢查