

大專用書

# 微電腦系統程式

林仁政 編譯

SYSTEMS  
PROGRAMMING  
FOR SMALL  
COMPUTERS

DANIEL H.  
MARCELLUS



全華科技圖書公司 印行

676511

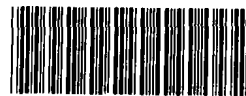
九世尊 贈送

華僑圖書館藏

大專用書

# 微電腦系統程式

林仁政 編譯



A0405652

SYSTEMS  
PROGRAMMING  
FOR SMALL  
COMPUTERS

DANIEL H.  
MARCELLUS



全華科技圖書公司 印行



全華圖書

法律顧問：陳培豪律師

## 微電腦系統程式

林仁政 編譯

出版者 全華科技圖書股份有限公司

地址 / 台北市龍江路76巷20-2號2樓

電話 / 5811300 (總機)

郵撥帳號 / 0100836-1號

發行人 陳 本 源

印刷者 華 一 彩 色 印 刷 廠

門市部 全友書局(黎明文化大樓七樓)

地址 / 台北市重慶南路一段49號7樓

電話 / 3612532 • 3612534

定 價 新臺幣 220 元

二版 / 76年12月

行政院新聞局核准登記證局版台業字第〇二二三號

版權所有 翻印必究

圖書編號 0121053

# 我們的宗旨：

**推展科技新知  
帶動工業升級**

**為學校教科書  
推陳出新**

感謝您選購全華圖書  
希望本書能滿足您求知的慾望

「圖書之可貴，在其量也在其質」，量指圖書內容充實，質指資料新穎夠水準，我們本著這個原則，竭心盡力地為國家科學中文化努力，貢獻給您這一本全是精華的“全華圖書”

為保護您的眼睛，本公司特別  
採用不反光的米色印書紙!!

# **SYSTEMS PROGRAMMING FOR SMALL COMPUTERS**

**DANIEL H. MARCELLUS**

*Watson School of Engineering  
State University of New York, Binghamton*

# 原 序

假如您認為系統程式設計的工作就是撰寫一個好的系統程式，那麼這將是一本很適合您讀的好書。的確，本書將為您說明如何設計小型電腦系統使用的編輯程式 (editors)、解譯程式 (interpreter)、編譯程式 (compiler)、作業系統等。假如您認為系統程式設計的工作是在討論難懂而抽象的機器特性及深奧語言翻譯程式 (translator) 之正規理論，那麼本書就不適合您了，因為本書並不探討這些課題。

本書的目的是提供系統程式設計的一個入門課程，在這裡我們所做的是把所有一般系統軟體之最簡單的例子收集在一起。這些例子有些是為本書而設計的，有些是取自著名的商用例子。簡單的例子有助於了解。我們試著探究它們的結構，使讀者能了解並能設計這類軟體。我們希望給讀者一種反應：“喔！我也能設計！”。

本書並不依照慣例編排，因為所考慮的軟體是針對所有微電腦或迷你電腦。我們採用這種方式的理由是為了解決您在第一次想要了解在電腦內是如何工作時，所面臨大型電腦理論的挫折感。當然，有一套共同的理論能連接所有電腦的軟體，此理論是系統程式設計的核心，但此核心如以微電腦來討論，則更容易。因此假如您的手上有一顆 CPU，您將會很有信心地去了解此 CPU 在做些什麼。

本書所提供的教材架構也就是課程的安排，是採用一種相當成功的程式語言教學方法。首先讓您了解程式設計語言中所有指令之一小部份，而其他重要的部份，在學生能夠深入了解之後，再把特殊且功能更强的特性有秩序地介紹出來。把這種方法用在系統程式設計上，其困難是在各種不同層次上，要提供什麼教材？我們如何把重要觀念的核心成功且詳盡地描述出來呢？

在認定層次特性上，我們編排的原則是以整個系統複雜度為主，系統複雜度的演進也和系統歷史發展相吻合。因此，我們的最內層相當於是最簡單且最老式的系統軟體。它們是：僅讀記憶式的監督程式作業系統、組譯程式（assembler），和輔助發展程式之巨集處理程式（macroprocessor）和鏈結程式（linker）。更上一層則包含更複雜的控制軟體，像單使用者磁碟作業系統和程式發展軟體，如編輯程式、編譯程式，和解譯程式。最外層則是多工（multi-tasking）作業系統。

對每一個不同軟體層次的特性，我們會介紹相關的硬體以配合軟體，如此才能對整個系統全盤了解。

誰適合來使用本書呢？本書適用於大學部學生第一門系統程式設計課程。它是傳統三門電腦科學課程順序上的第一門。這些課程是：

- 系統程式設計
- 編譯程式
- 作業系統

假如學生沒有時間，只能選讀其中一門課時，本書亦可用做短期課程教材，以先期了解整個作業系統，語言和公用程式。本書曾在紐約州立大學作者所執教系統程式設計課程上使用。最後，本書亦可供擁有電腦系統且想要了解比僅知道如何開關電腦及如何用高階語言撰寫程式更多的人做為自修的工具。

要有效使用本教材，是否要具備某種先決條件？讀者最好曾學習過高階程式語言和組合語言的課程，以及使用電腦解決問題的經驗。因為系統程式設計會牽涉到軟體和硬體，所以有電腦硬體課程做背景者更佳。但並不是絕對需要的。除此之外，本書亦很適合從事於微電腦實驗工作的人來閱讀。

在本書有許多地方需要以組合語言來做範例說明。所以我們選用 Intel 8080 組合語言。雖然 Intel 8080 晶片不是最好的，但它是最著名的晶片之一。對於不了解 8080 的人，我們在附錄會介紹它。

某些程度較佳的讀者可能不滿意本書對不同種類軟體的詳細解析。我們覺得要深入了解軟體就好像在學開車一樣，主要的是先把它開動，如此您才能知道每一部位如何工作。當然，我們可能了解經過汽化器時燃料混合的空氣動力學，或汽缸內

爆炸引起熱交換的熱力學，但若沒有把汽車如何動作看成是一個系統的話，是很難整體了解的。本書是把這種分析過程應用到電腦的系統上。

本書的方針是要切合實際，所以儘量包含一些對如何產生好的系統軟體有所助益的教材。要做到實際是很不容易的，但是我們所努力的是一個簡單的原則：假如您能完成它，您就能了解它。

### 感 謝

作者在此感謝許多促成本書出版的人。James Van Zee 最先使作者對小型電腦產生喜好。紐約 Polytechnic 學院 Melvin Klerer 教授提供簡化系統程式設計教學的基本觀念。Paul Davis 和 Jane Cameron 在許多技術上提供有價值的建議。Lance Leventhal 不斷給予鼓勵及對本書結構上的建議。感謝 Velda Bartek、James Van Zee 和 Juan Rodriguez-Torrent 等對錯誤的指正。最後感謝 WORDSTAR 程式的作者 Rob Barnaby。

DANIEL H. MARCELLUS



# 譯者序

系統程式主要目的是幫助使用者很容易使用此計算機系統，並有效運用系統的資源。本書提供設計微電腦系統上之監督程式、組譯程式、鏈結程式、載入程式、巨集處理程式、編輯程式、編譯程式、解譯程式、作業系統等等，由淺而深，由簡而繁，以循序漸近方式使讀者能容易接受、吸收。

本書主要是以小型電腦（微電腦）為討論對象，使您能建立整體系統的架構，以做為了解大型電腦系統的基礎。事實上許多大型電腦的軟體慢慢都可在小型電腦上使用。因為在硬體技術不斷進步，小型電腦的記憶容量不斷地增大，執行速度也加快許多。而小型電腦的使用，不僅適於教學，價錢便宜，而且功能也逐漸趕上大型電腦。因此，學習本書是了解電腦系統最直接而有效的捷徑。

本書能順利完成得感謝陳本源先生及李茂欽先生的鼓勵；同時也感謝張子建先生及黃光華先生的幫忙及校對。

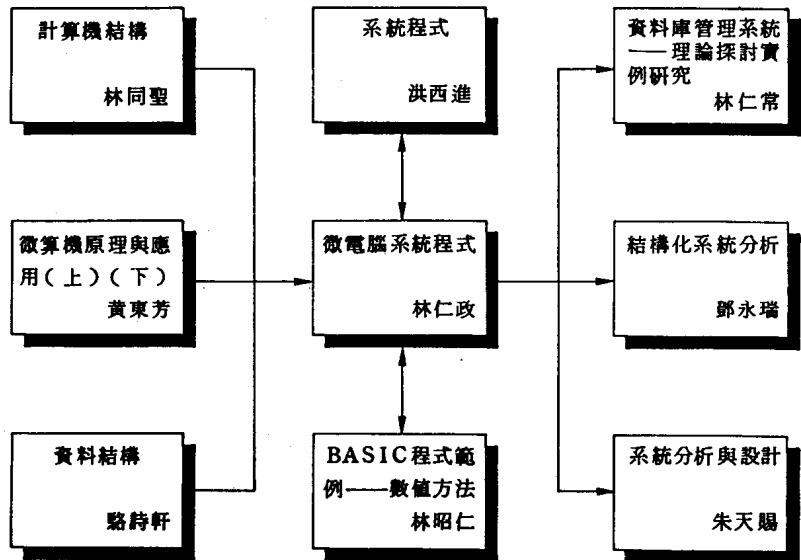
林仁政 謹識

# 編輯部序

「系統編輯」是我們的編輯方針，我們所提供給您的，絕不只是一本書，而是關於這門學問的所有知識，它們由淺入深，循序漸進。

現在我們將這本「微電腦系統程式」呈獻給您。Marcellus的“Systems Programming for Small Computers”出版後即廣受各大專院校採用為系統程式之教本，本書主要是以微電腦為討論對象，使讀者能建立整體系統的架構，以做為了解大型電腦系統的基礎，其內容的淺顯易懂，編排的結構嚴謹，讀者可建立深厚的系統程式概念，並為以後更深入的電腦課程奠立根基。

同時，為了使您能有系統且循序漸進研習微電腦方面叢書，我們以流程圖方式，列出各有關圖書的閱讀順序，以減少您研習此門學問的摸索時間，並能對這門學問有完整的知識。若您在這方面有任何問題，歡迎來函連繫，我們將竭誠為您服務。





<b>1</b>	<b>一般性的介紹</b>	<b>1</b>
1.1	何謂系統程式設計	2
1.2	系統軟體的演進	3
1.3	為什麼介紹小型電腦的系統程式設計方法？	8
1.4	目前的發展方向？	10
1.5	系統程式複雜度的測量	11
1.6	系統程式設計的學習計劃	12
	習題	13
<b>2</b>	<b>系統環境：第一部份</b>	<b>15</b>
2.1	單板計算機系統簡介	16
2.2	硬體特性及介面	17
2.2.1	介面埠	17
2.2.2	鍵盤	18
2.2.3	簡單的印表機	19
2.2.4	儲存程式用的磁帶機	21
2.3	簡單輸出入裝置的程式設計	24
	習題	28
<b>3</b>	<b>最簡單的作業系統：僅讀記憶監督程式</b>	<b>31</b>
3.1	簡介：電腦的基本控制	32
3.2	使用者介面的典型設計	33
3.3	監督程式所使用的資料結構	36
3.4	監督程式如何決定下一步驟做什麼：控制和剖析	38
3.5	任務識別後之任務執行	44
	習題	52

<b>4</b>	<b>組合語言</b>	<b>55</b>
4.1	簡介：比機器語言還高一階的語言	56
4.2	符號表之排序和查尋	60
4.3	基本的二次處理組譯程式	66
4.4	增加符號表式功能	75
4.5	可重定位的組合碼	80
	習題	82
<b>5</b>	<b>鏈結程式和載入程式</b>	<b>85</b>
5.1	簡介：大型程式的處理	86
5.2	鏈結程序	87
5.3	鏈結程式庫	92
5.4	載入程序	93
	習題	95
<b>6</b>	<b>巨集處理程式</b>	<b>97</b>
6.1	簡介：利用巨集來擴充語言的功能	98
6.2	巨集處理系統的特性	98
6.3	簡單式巨集組譯程式的設計	104
	習題	107
<b>7</b>	<b>系統環境：第二部份</b>	<b>109</b>
7.1	簡介：强有力的單使用者微電腦系統	110
7.2	硬體特性和介面	112
7.2.1	軟式磁碟	112
7.2.2	視頻顯示器	115
7.3	在强有力的硬體環境中的程式設計	118
	習題	121
<b>8</b>	<b>單使用者磁碟作業系統：CP/M</b>	<b>123</b>
8.1	簡介：為軟體創造一個環境	124
8.2	CP/M的記憶體用法	127
8.3	最低層次的輸出入程式 (BIOS)	130

8.4	實體與邏輯上的輸出入動作	133
8.5	CP/M的檔案概念	134
8.5.1	系統之檔案處理概觀	134
8.5.2	磁碟目錄的處理	136
8.5.3	動態的空間分配	138
8.5.4	實際及邏輯的磁碟磁扇	142
8.5.5	存取檔案中的記錄	143
8.6	與機器無關的輸出入 (BDOS 常式)	146
8.7	控制台命令處理程式 (CCP)	148
8.8	系統的載入與啓始狀態的設定	153
8.9	更強而有力的單使用者磁碟作業系統之特性	155
	習題	157



## 文字編輯程式

159

9.1	簡介：程式製作的捷徑	160
9.2	編輯程式與使用者之間的交談介面	162
9.3	記憶緩衝區內之檔案編輯	165
9.4	超長檔案之編輯程式	169
9.4.1	特定文字列的定址	171
9.4.2	記憶體與磁碟間的資料調換	173
9.4.3	列刪除	175
9.4.4	插入新增之列	177
9.4.5	列內資料的更動	180
9.5	更具功能之編輯程式	184
9.6	全螢幕編輯程式：最複雜的設計	185
	習題	187



## 高階計算機語言的描述與分析

189

10.1	語言系統簡介	190
10.2	非正式地定義一套語言：TINY BASIC	192
10.3	正式地定義一套語言：TINY BASIC	196
10.4	使用最終狀態結構來分析與剖析程式	204
10.5	高等剖析技巧	211
	習題	212



## 解譯程式：TINY BASIC

215

11.1	解譯系統之簡介與範例設計	216
11.2	編輯與程式之內部形式	218
11.3	反波蘭記法的執行程式	222
11.4	解譯程式概觀	225
11.5	陳述剖析	228
11.6	提取下一條指令	231
11.7	符號表的維護	231
11.8	單一程式陳述之處理常式	233
11.9	循環圈處理的控制	236
11.10	培基語言的透視方法	239
	習題	242



## 編譯程式：TINY BASIC

245

12.1	編譯之簡介與設計範例	246
12.2	陳述列之剖析	250
12.3	分支點之碼產生	252
12.4	程式陳述的碼產生	254
12.4.1	指定陳述之碼產生	254
12.4.2	條件分支之碼產生	256
12.4.3	非條件分支之碼產生	261
12.4.4	副常式呼叫之碼產生	261
12.4.5	返回陳述之碼產生	262
12.4.6	插入陳述之碼產生	262
12.4.7	輸出陳述之碼產生	263
12.4.8	標題陳述之碼產生	263
12.4.9	註解陳述之碼產生	265
12.4.10	結束陳述的碼產生	266
12.5	循環的碼產生	266
12.6	代數表示式之碼產生	271
12.7	程式編輯之範例	274
12.8	解譯器與編譯器之比較	275

## **13** 系統環境：第三部份 279

13.1 大型的微電腦系統簡介	280
13.2 硬體特性及介面	284
13.2.1 更具能力的微處理機	284
13.2.2 溫徹斯特磁碟	286
13.2.3 可擴充的記憶體位址空間	287
13.2.4 使用岔斷系統	291
13.3 由岔斷驅動輸出入的設計	293

## 習題

301

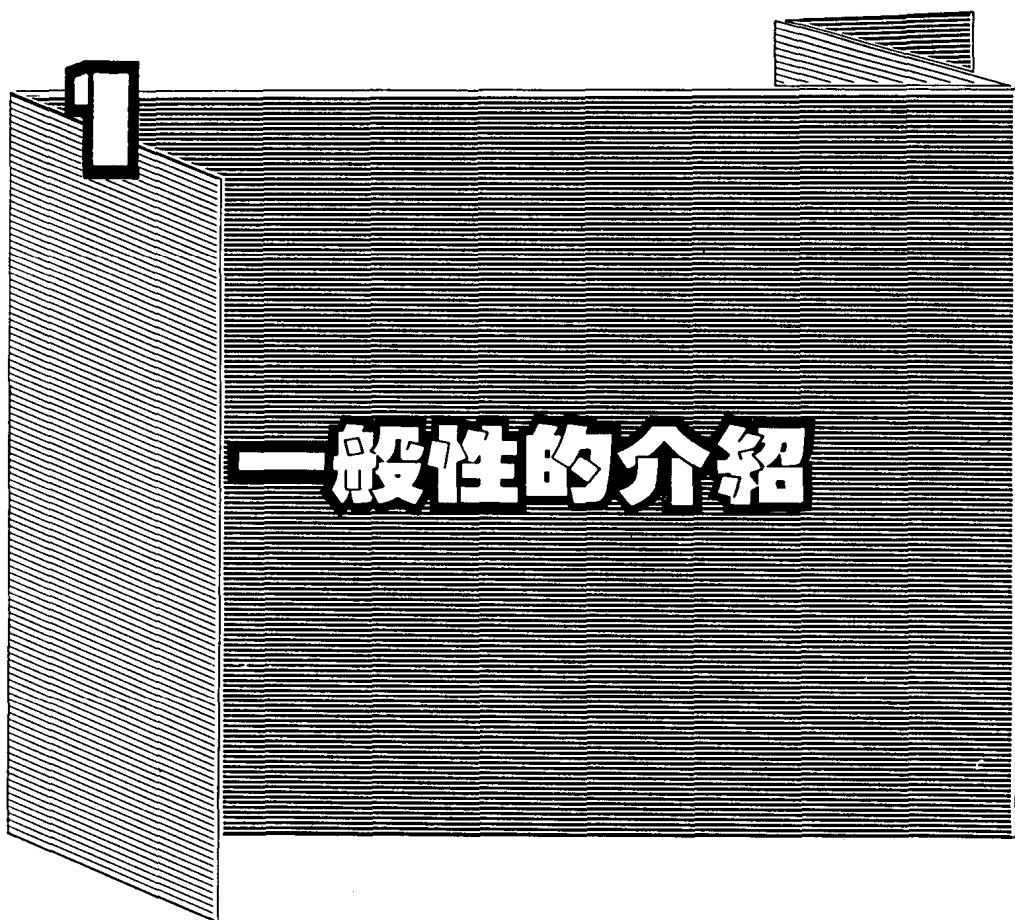
## **14** 多工作業系統 303

14.1 簡介：電腦的最佳用途	304
14.2 工作配送程式及定序程式	307
14.3 並行工作間控制的交互作用：MP/M-II	314
14.4 即時作業系統之核心——VRTX	323
14.5 樹狀檔案結構：UNIX	327
14.6 樹狀結構的處理召喚：UNIX	331
14.7 嵌入式的命令語言解譯程式：UNIX SHELL	333
習題	340

## 附錄

343

A.1 學習 Intel 8080 微處理機之程式規劃	343
A.2 數位通訊用的 ASCII 字元碼	352
圖表及資料結構的中文索引	353
索引	355



<b>1.1</b>	何謂系統程式設計？	2
<b>1.2</b>	系統軟體的演進	3
<b>1.3</b>	為什麼介紹小型電腦的系統程式設計方法？	8
<b>1.4</b>	目前的發展方向？	10
<b>1.5</b>	系統程式複雜度的測量	11
<b>1.6</b>	系統程式設計的學習計劃	12



## 1.1 何謂系統程式設計？

### (WHAT IS SYSTEMS PROGRAMMING?)

計算機系統是用來幫助人們在執行資料處理和解決計算問題上的一套工具。此工具的某一部份是由機械硬體和電子線路所組成，像計算機本身和週邊設備——終端機、印表機、磁碟機和通訊設備等。另一部份是軟體——即系統程式，這些程式使得使用者不需要了解那些複雜且對問題毫不相干的計算機系統內部運作的情形，就能解決他的問題。

系統程式有兩個目的，它們使一般人能很容易使用此計算機系統，並且能有效地利用系統的資源。當一套計算機系統賣出去的同時，系統程式也是包含在內的，通常這些程式是由製造商或賣方提供。例如：

- 輸出入次常式套裝軟體 (Input/Output subroutine packages)
- 監督程式 (monitor)

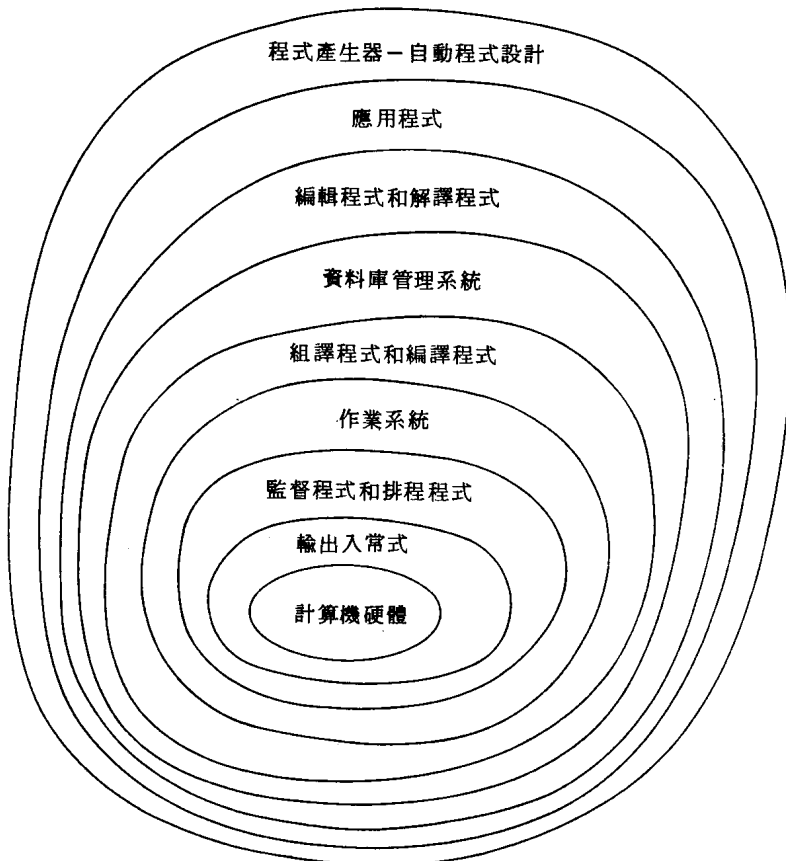


圖 1.1 系統軟體結構圖。使用者和系統軟體是層層相隔的