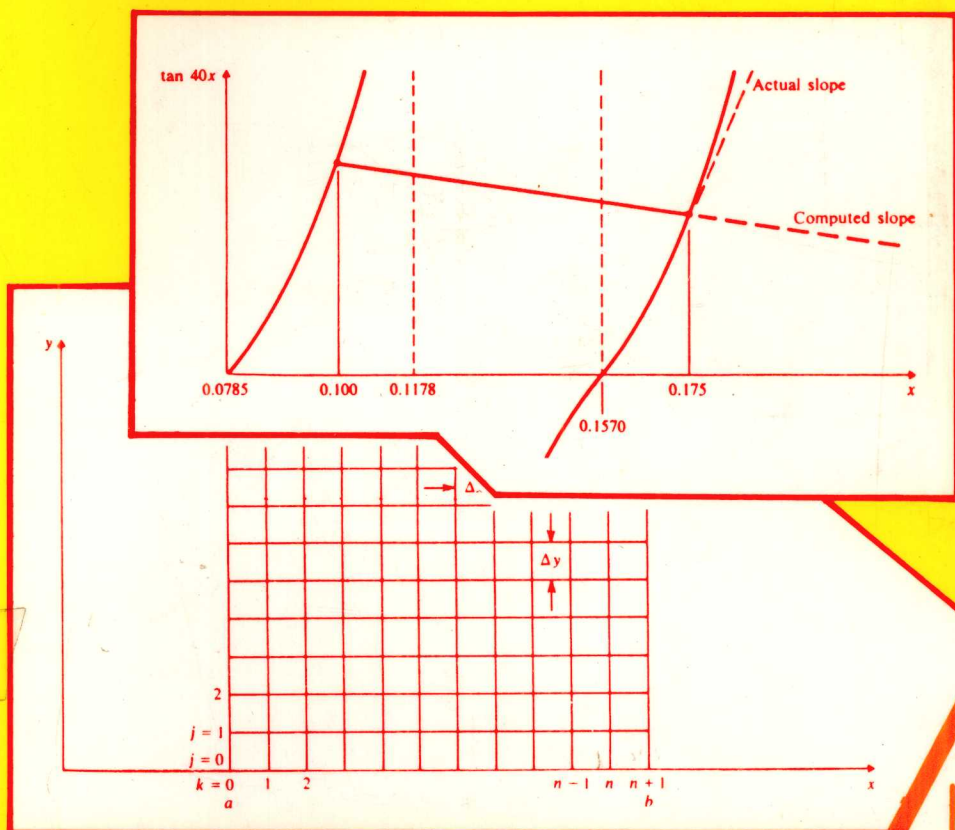


數值方法

附 BASIC 程式



姚修慎 譯

版權所有
翻印必究

詳論C程式與8086組合語言程式呼叫

編譯者：蔡 明 峯

發行人：楊 鏡 秋

出版者：儒林圖書有限公司

地 址：台北市重慶南路一段111號

電 話：3118971-3 3144000

郵政劃撥：0106792-1號

吉豐印刷廠有限公司承印

板橋市三民路二段居仁巷一弄五十三號

行政院新聞局局版台業字第 1492 號

中華民國七十七年五月初版

定價新台幣 220 元正

序 言

我們在撰寫本書之初就是希望能達到下列三個目標。首先，把如何加快C程式碼執行速度的方法、如何妥善地運用C編譯程式以及C常式可能變慢或受到限制之處告訴讀者。我們將利用一些具體而微的範例，提供較為複雜的技巧來分析以及增加硬體的執行效率。其二，我們希望能夠把MS-DOS和C語言對8086組合語言界面的所有相關資料收集在本書中。對於IBM PC，儘管我們很早就曉得它的奧妙是在於ROM BIOS程式、與週邊設備有關控制元件的規格及顯示器界面卡的線路圖，但欲從其中得到的資料轉為可使用的程式仍是相當困難。我們第三個目標就是要把這些經驗告訴給對增進C程式語言能力有興趣的人，讓他可以很容易地進入情況。希望我們的努力有成果，更希望IBM PC程式設計員讀完本書後，能從簡單的程式設計一躍進入日漸流行的C語言高等程式設計的領域中。

Mitchell Waite

Harry Chesley

前言

做為 IBM PC 應用軟體的撰寫者，我一星期至少得為顧客花上 30 小時的時間來寫 C 程式。常常我必須把我的程式修改以符合不同機器的使用，或者是尋找一些能夠使得程式執行速度變快的方法。但是在某些地方，C 語言無法符合需要，必須借重於組合語言和 MS-DOS 的內部環境。

我一直期望能有一份指導我修改 C 程式最終碼的單一資料來源。現在我發現，在讀過由 Harry Chesley 和 Waite Group 一起撰寫的“以組合語言強化 C 語言能力”(Supercharging C with Assembly Language)一書的原稿後，已經找到我所需要的資料了。

作者所謂的“強化工作”就是完整的“最佳化”(optimization)。藉著 Microsoft C 的程式範例和穿插著片段以 MASM 組合語言所寫的程式，作者教導我們如何從各個角度來使程式碼最佳化。作者 Chesley 在書中談到如何把組合語言程式加入 C 程式碼的方法以及如何辨別使用組合語言程式的時機。另外，他還提供一些作為惡作劇使用的組合語言程式加入的位置以及告訴您何時可使程式運作出錯或變慢。在學過如何使組合語言程式最佳化之後，Chesley 接著列出很多直接使用 ROM BIOS 來改善程式的範例。

本書有許多程式範例告訴您如何使用 PC 顯示幕、鍵盤和串列埠潛在的能力。其內包括有使用 EAG 界面卡的完整資料、產生衍生圖形的程式以及模擬串列式終端機的程式。

本書的最後一節討論到直接使用 PC 的硬體，雖然這一點在軟體可攜性的問題上常常為人所詬病，但它的確有必要存在於程式內。舉例說，“視窗程式”若藉著 ROM BIOS 與“位元對應”(bit mapped)的 EGA 交談，在效率上將比不上直接存取 EGA。這本 Chesley 書在探討硬體及程式的效率上給我很

深刻的印象。他在本書裏討論了所有主要的晶片和觀念，包括直接存取顯示幕，高速的動畫、中斷式的串列輸出輸入以及聲音的直接控制。

在本書的最後一章將談到他所謂的“IP Histogram”程式。利用它可以找出佔用最多執行時間的程式段。在您使用 IP Histogram 找出程式執行較慢之處後，您可以再利用書本前幾章所提到的技巧重新撰寫這段程式碼，以增加執行的速度。

這本書另外相當有價值的部份是 Harry 為顯示幕繪圖所提供的 C 基本指令集。我發現這些函數撰寫的相當好而且易於使用，以至於我捨去了商業用的繪圖指令集；從現在起，在我的程式中將開始使用這些函數。

Lloyd Zusman
Master Byte Software

目 錄

第一章 導論.....	1
閱讀本書前應該有的認識.....	4
本書的組織架構.....	4
第二章 Supercharging的觀念.....	7
縱觀 IBM PC 的架構.....	9
強化工作的種類.....	11
改善處理器的執行速度.....	12
改善磁碟輸入／輸出的速度.....	13
跳過MS - DOS.....	13
跳過ROM BIOS的一般化常式.....	13
使用PC 內藏的功能.....	14
強化工作的得失.....	14
模組化的程式設計.....	17
區別模組的功能及模組的內部細節.....	18
寫成小型的模組.....	18
減少模組資料的傳遞.....	19
避免內隱式模組間的資料傳遞.....	21
度量程式的執行狀況.....	22
摘 要.....	24

練習.....	24
第一部份 強化執行速度	25
第三章 如何由C語言呼叫組合語言	27
簡述組合語言程式的呼叫.....	29
分段與分段群.....	31
編譯程式的記憶體模式.....	35
C語言與組合語言間的介面.....	37
編譯程式分段與分段群的用法.....	38
呼叫協定.....	41
使用組合語言的捷徑.....	45
摘要.....	46
練習.....	47
第四章 處理器執行時間的最佳化：RAMSort	53
RAMSort.....	55
RAMSort C語言程式.....	56
把C語言轉換為組合語言.....	60
測量執行效益.....	60
錯誤的改善方式.....	61
正確的改善方式.....	63
最後的調整.....	63
摘要.....	64
練習.....	64
第五章 改善處理器執行的速度：井字遊戲	81
井字遊戲.....	83
搜尋棋步的策略.....	85

井字遊戲程式.....	87
改善井字遊戲程式.....	90
摘 要.....	92
練 習.....	92
第六章 輸入／輸出速度的改善：Encrypt	103
編密碼.....	105
Encrypt 程式.....	108
作業系統與 C 語言檔案的輸入／輸出.....	108
改善程式的效率.....	112
改善 CPU 執行效益.....	114
其它的磁碟驅動器.....	115
摘 要.....	115
練 習.....	115
第二部份 使用ROM BIOS	121
第七章 如何呼叫ROM BIOS	123
中斷.....	125
中斷的程序.....	127
ROM BIOS.....	129
呼叫ROM BIOS.....	134
int86 函數.....	139
摘 要.....	140
練 習.....	140
第八章 螢幕控制：ShowFile	155
字元輸入／輸出.....	157
鍵盤字元輸入.....	158

ShowFile 程式	159
顯示文字檔	162
VANILLA	163
ANSI	163
ROM BIOS	163
直接存取顯示區	164
組合語言常式	164
執行效益	164
摘 要	165
練 習	166
第九章 顯示與鍵盤輸入／輸出：Border	175
顏色顯示與顯示屬性	177
ROM BIOS 鍵盤函數	178
Attrib 程式	180
摘 要	180
練 習	181
第十章 CGA和EGA與ROM BIOS繪圖功能：Fractal	187
部份衍生圖	189
ROM BIOS 圖形顯示常式	191
Fract 程式	194
執行 Fract 程式	197
摘 要	199
練 習	199
第十一章 序列埠—鍵盤／顯示輸出入：Term	207
序列輸入／輸出	209
Term 程式	214

BIOS 的序列埠函數.....	215
Term Print	217
印表機輸出埠 BIOS 函數.....	218
摘 要.....	220
練 習.....	220
第三部份 直接控制硬體.....	231
第十二章 如何直接控制硬體.....	233
綜觀 IBM PC 硬體架構.....	236
使用程式位址空間外的記憶體.....	242
使用 IN 和 OUT 指令.....	247
使用中斷處理.....	250
摘 要.....	254
練 習.....	255
第十三章 螢幕顯示的直接控制：ShowFile II.....	257
顯示記憶體.....	259
在主記憶體低位址的顯示資料.....	262
新的 scrPuts 函數.....	263
重回 ShowFile 程式.....	265
摘 要.....	265
練 習.....	266
第十四章 高速的物體動畫製作：Pong.....	269
顯示記憶體.....	272
rect 函數.....	273
rect 函數的內部處理.....	277
Pong 程式.....	283

摘 要.....	284
練 習.....	284
第十五章 中斷驅動的序列輸入／輸出：Term II.....	295
序列埠輸入／輸出.....	297
序列埠中斷.....	300
循環式緩衝區.....	301
中斷驅動的序列介面.....	302
Term 2 程式.....	304
摘 要.....	304
練 習.....	305
第十六章 直接控制音效：NoiseMaker	315
音效硬體.....	317
NoiseMaker 程式.....	319
處理器的執行速度.....	320
直接控制揚聲器.....	322
摘 要.....	323
練 習.....	323
第十七章 強化工作的工具：IP分析表	329
IP 分析表.....	331
計時器中斷.....	334
IP 分析表模組.....	335
使用 IP 分析表模組.....	336
連結表.....	337
分析 RamSort	338
摘 要.....	339
練 習.....	339

第四部份 附錄	349
附錄A 對C程式設計者的組合語言介紹	351
簡易的範例	353
組譯程式的語法	354
資料結構與定址模態	355
暫存器	356
記憶體 & 分段	357
指令的執行	359
定址模態	359
推疊	361
旗標	362
處理器指令	364
算術運算	364
資料移動	366
執行的流程	367
輸入／輸出	369
字串的處理	370
擬似指令	372
資料儲存區	372
分段、分段群和 ASSUME 擬似指令	372
程序	374
摘要	374
附錄B 編譯程式、組譯程式和連結程式的用法	379
編譯與組譯	381
連結過程	381
使用編譯程式、組譯程式和連結程式	382

附錄C ROM BIOS中斷和暫存器的用法.....	387
附錄D 鍵盤碼	395
名詞解釋	399
索 引	413

第一章

導 論

對“強化工作”一詞的定義和範例

閱讀本書前應該有的認識

本書的架構

“強化工作”與改善並增強程式能力是同樣的意思。本書將會告訴您如何來“強化”您的C程式。這本書將從處理器速度和磁片存取的最佳化來改善您程式的速度。它也將告訴您如何從C程式中呼叫組合語言以及如何來寫這類的組合語言常式。最後，書中也提到如何呼叫ROM BIOS直接使用先前您的程式無法控制的設備，及減少MS-DOS的使用時間以加快程式的執行速度。它也說明了如何直接存取或控制硬體。硬體的直接使用就如同ROM BIOS的直接使用，允許您能做更多的工作或做得更有效率。沿用這樣的方式，您將學到很多使用在後面將討論到的技巧的程式。這些程式當然是有用的，您可以以它們作基礎來開發更為複雜的應用。而且為了使日常生活更有趣些，您也可寫一些好玩的遊戲，同時也能把本書的精神表現於其中。

學習有關程式設計方面的新題材，最好的方法是了解其原理、學習範例、再將之編譯然後執行它，最後是加入其它的功能以增進此程式的能力。為此，我們的範例程式是完整的、隨時可以在機器上執行的，而且是具備有用的功能的獨立應用程式。再者，我們於每一章的最後加上一些能夠以某種方式來擴充範例程式功能的練習。本書中還包括了下列這些完整且可執行的程式：

- ShowFile 顯示文字檔並允許使用者利用控制游標鍵來顯示文字檔的不同部份。
- RAMSort 在隨機存取記憶體中，對不同的文字檔排序（類似於MS-DOS的SORT過濾程式，不同的是此處有原始程式碼）。
- Encrypt 編密碼及解密碼。
- TicTac 漂亮的井字遊戲。
- Term 在序列埠上模擬簡單的終端機。
- Fract 顯示同一類的部份衍生圖樣。
- Pong 顯示幕上的乒乓球遊戲。
- Noise 製作音效。
- IP Histogram 測量出程式最花時間的部份。

閱讀本書前應該有的認識

這本書主要是為熟悉在 IBM PC 上使用 C 語言的程式設計員所寫的。它被設計來幫助您進行更深入的探討：將組合語言程式加入您的 C 程式中，直接使用 ROM BIOS 和機器的硬體介面以及儘可能使程式的執行速度加快。

本書對 C 語言初學者也相當有用。雖然本書不是為初學者所寫的，但書中有很多的範例可幫助他們了解 C 語言，進而寫 C 語言的程式，特別是對那些熟悉 IBM PC 組合語言的人。

這本書所有的內容均適用於 IBM PC、IBM PC/XT 和 IBM PC/AT。其所列舉的範例皆已在這些機型上執行過。而大部份的範例也可在 IBM PCjr 和 IBM PC 衍生型的機器上執行。使用到 ROM BIOS 的程式可能無法在與此 BIOS 不相容的機器上執行。但若能與 IBM PC 完全相容者，所有書中談到的觀念皆可運用於其上。另外，由於本書濾過與作業系統相關的內容，因此書中所討論的與作業系統的版本無關。在此先聲明，書中提到的作業系統是指“MS-DOS”，但所有的程式同樣地可以在 PC-DOS 上執行。

此外，在本書中均使用 Microsoft C (version 3.0 或 version 4.0 和 Microsoft macro assembler MASM。而且書中所述的理論以及大部份的特殊程式碼皆可應用於其它的編譯程式與組譯程式之組合。每當有編譯程式或組譯程式的特別資料，書本將會指出，並且在那一節內的要點將說明得更加詳盡。因此使用其它編譯程式或組譯程式的讀者也將發覺本書相當有用，但是當應用特別的資料或是編譯範例時須多加小心。

本書的組織架構

第一章與第二章是介紹與解釋“強化工作”的觀念。其餘章節，也就是本書的主要部份，分為三個部份。

- 第一部份 執行速度的最佳化。
- 第二部份 直接使用 ROM BIOS。