

組合語言 第四版

ASSEMBLY LANGUAGE FOR
INTEL®-BASED COMPUTERS
FOURTH EDITION

KIP R. IRVINE 原著

王國華、白能勝、李文猶、蕭文俊 編譯

本書適用於對作業系統、計算機結構、以及微處理機課程中的程式設計有興趣的學生和教授。Kip Irvine極盡所能讓本書第四版的内容更趨完備，他努力結合Windows/MS-DOS作業系統，並且完整說明以Intel為基礎之電腦的32位元組合語言的應用。

本書著重於如何以機器層級的程式語言，以Intel為基礎之電腦的組合語言，來進行程式設計。

本書的特色如下：

- 詳細解說數字系統與資料儲存的基本觀念。
- 所有的程式都經過微軟的MASM 6.15TM組譯器進行測試無誤。
- 內嵌組合語言程式碼，並且在真實與保護模式下，將組合語言與C/C++程式語言連結。
- 豐富的指令集參考資料，包含指令格式與CPU旗標的使用方法。
- 中斷向量和裝置I/O。
- 隨書光碟包含微軟MASM 6.15TM組合語言開發系統的評估版、可供程式設計師使用的編輯器、巨集函式庫，以及本書的原始碼。

第四版新增的特色：

- Win32程式設計，包含文字介面的API和圖形應用程式。
- 額外說明程序、遞迴、堆疊參數、結構和聯集等主題。
- 布林運算式、真值表與流程圖。
- 基本的字串操作、排序和搜尋演算法。
- 真實模式和保護模式下的位元映射圖形。
- IEEE浮點二元表示法。
- 虛擬機器結構；IA-32保護模式分段與分頁。
- 簡介指令執行週期、記憶體I/O、多工、管線，以及超純量架構。
- 磁碟的基礎知識，包含磁碟的組織結構、FAT32和NTFS檔案系統。

專案編輯：王妍蓉 / 凌鈺城 / 陳泳錫 / 鄭孟璿

PEARSON

Education
Taiwan

培生教育出版集團

<http://www.pearsoned.com.tw>

ISBN 957-21-4327-1



9 789572 143278

NT / 680



0 0 6 8 0

RT
312.933
4310
154980

Education
Taiwan

05303007

Free Microsoft® MASM 6.15™
Assembler Included

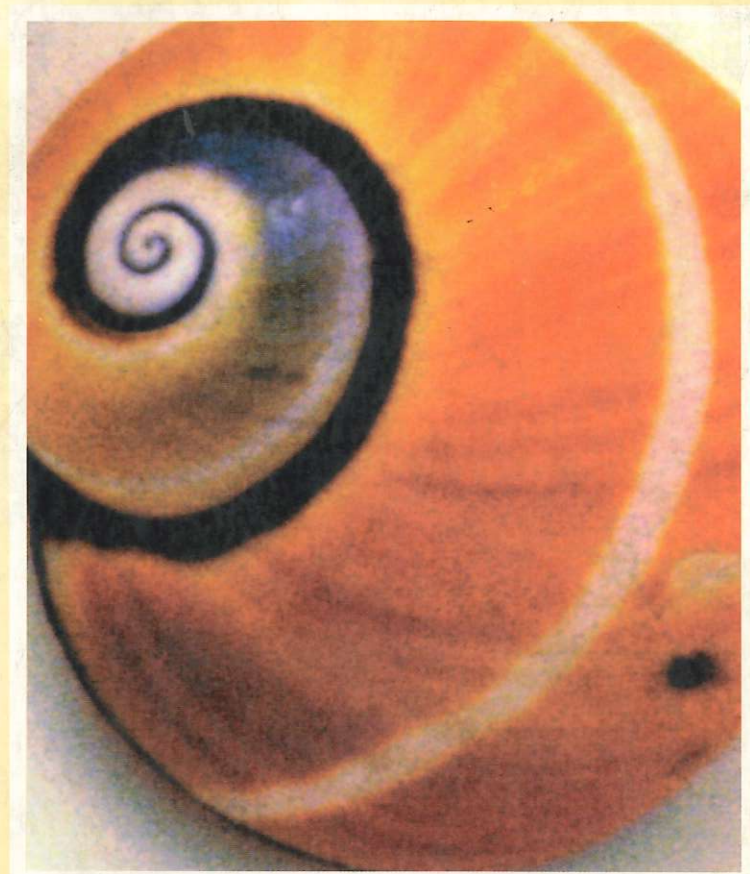
組合語言 第四版

國立勤益技術學院圖書館



154980

SEMBLY
UAGE FOR
INTEL®-BASED
COMPUTERS
FOURTH EDITION



KIP R. IRVINE 原著

王國華、白能勝、李文猶、蕭文俊 編譯

PEARSON
Prentice
Hall

全華科技圖書股份有限公司 印行

RT
312.933
4310
154980

組合語言

ASSEMBLY LANGUAGE FOR INTEL®-BASED COMPUTERS

KIP R. IRVINE 原著

王國華、白能勝、李文猶、蕭文俊 編譯



國立勤益技術學院圖書館



154980



全華科技圖書股份有限公司

PEARSON
Education
Taiwan

台灣培生教育出版股份有限公司

Pearson Education Taiwan Ltd.

序

本書是基於 Intel IA-32 處理器架構，並以程式設計者的觀點所撰寫而成。它適合用來當做計算機科學相關科系中，以下幾種課程的教科書：

- ◇ 組合語言程式設計(Assembly Language Programming)。
- ◇ 計算機系統基礎(Fundamentals of Computer Systems)。
- ◇ 計算機架構基礎(Fundamentals of Computer Architecture)。

雖然本書原來是設計用來作為社區大學程式設計課程學生用的教科書，但它已漸漸地更深入的發展。近年來，許多大學使用本書用來當做計算機架構(國內多譯為計算機結構)課程的先修。舉例而言，在佛羅里達國際大學裡，本書被用來當做課程名為計算機系統基礎(Fundamentals of Computer Systems)的教材，引領學生進入更廣泛的電腦結構課程。

先前的版本包含了一些主題，讓您可以在很自然的情況下進入了往後的課程，像是計算機架構、作業系統及編譯器的撰寫：

- ◇ 虛擬機器概念(Virtual machine concept)。
- ◇ 基礎布林邏輯運算(Elementary boolean operations)。
- ◇ 指令執行週期(Instruction execution cycle)。
- ◇ 記憶體存取時脈週期(Memory access using clock cycles)。
- ◇ 中斷與輪詢(Interrupt and polling)。
- ◇ 多階管線(Multi-stage pipeline)。
- ◇ 超階結構(Superscalar architecture)。
- ◇ 多工處理(Multitasking)。
- ◇ 載入與執行程式>Loading and execution programs)。
- ◇ 二進位浮點表示法(Floating-point binary representation)。

其他的與 IA-32 架構相關的特定主題，擷取自它們的手冊：

- ◇ IA-32 保護記憶體的定址與分頁(IA-32 Protected Memory addressing and paging)。
- ◇ 在實體位址模式下的記憶體分段(Memory segmentation in Real-address mode)。
- ◇ 中斷處理(Interrupts handling)。
- ◇ 直接硬體 I/O(Direct hardware I/O)。
- ◇ 指令編碼(Instruction encoding)。



商標聲明

在書本中所使用的商標名稱，因為編輯的原因，沒有特別加上註冊商標符號，我們並沒有任何冒犯商標的意圖，在此聲明尊重該商標擁有者的所有權利。

CHWA
TECHNOLOGY

出現在本書中的某些範例將會與您往後的計算機科學(computer science)課程做連結：

- ◇ 搜尋與排序演算法(Searching and sorting algorithms)。
- ◇ 高階語言結構(High-level language structures)。
- ◇ 有限狀態機(Finite-state machines)。
- ◇ 編碼最佳化範例(code optimization examples)。

在此版本之中，有一些和程式寫作相關的新特色：

- ◇ 更深入且更具邏輯性的資料定義說明。
- ◇ 更仔細的位址模式說明。
- ◇ 簡易的連結函數庫，全部的函數只需要極少的輸入參數即可使用。有幾個新的程序，例如可用來傾印 CPU 的暫存器及記憶體區段，和可作為延遲的計時器等等的函數。
- ◇ 對由上而下的程式設計方法的說明及示範展示。
- ◇ 使用流程圖當做是編碼產生的工具。
- ◇ 更全面地涵蓋到組合語的指引(directives, 假指令)、巨集(macros)及運算子(operators)等。例如，PROC、PROTO 及 INVOKE 指引將都被完全地說明及示範展示。
- ◇ 更完整地涵蓋到結構部份，包含巢狀結構及陣列結構。
- ◇ 塊狀結構形式的 IF、WHILE 及 REPEAT 敘述(一個 MASM 的進階特點)。
- ◇ 同時使用 BIOS 及直接記憶體對映技術，介紹視訊繪圖。
- ◇ 滑鼠程式。
- ◇ 使用呼叫 Kernel32 視窗函數庫撰寫 Win32 主控台程式。
- ◇ 更多陣列應用範例。

□ 仍舊是一本程式設計的書 注意到本書仍舊鎖定在它的原始任務：教導學生如何在機器層次撰寫程式及為程式除錯，這是最為重要的。整本書不會完全地變成計算機架構的教材，但它確實在讓學生透過瞭解電腦實際運作情形上，來獲得第一手撰寫軟體程式的經驗。這樣子的價值不能被低估，因為透過對機器直接的接觸，才能得能到大量的理論與實際知識。在工程相關課程之中，學生建立模型；在軟體課程之中，學生撰寫程式。在二者之中，他們都將擁有一個難忘的經驗能給予他們信心，使得未來將能夠在任何作業系統/機器為導向的環境下作業。

□ 真實模式及保護模式 許多教授指出希望使用 Intel 的保護記憶體模組來將程式轉變為 32-bit 的程式設計。這個版本主要著重在 32-bit 保護模式，但仍將仍然有三個章節專門用來講解真實模式的程式設計。例如，有一個完整的章節講到和鍵盤、視訊顯示(包含繪圖)及滑鼠相關的 BIOS 程式設計。還有另一個章節專門用來講使用中斷(函數)呼叫的 MS-DOS 程式設計。讓學生有一些直接和軟體及硬體相關的程式設計經驗是非常有益處的。

在本書的第一個部份之中出現的範例，幾乎全部都是在保護模式下執行的 32-bit 文字導向應用程式，使用平坦記憶體模組。它非常地直接、明確。學生們不用再處理區段偏移定址。有一些特別標示的圖示及方塊圖，可以提示學生在保護模式下及實體位址模式下程式設計的微小差異。大部份的差異被隱藏在書中的二個連結函數庫之中。

□ 連結函數庫 在本書中，有二個版本的連結函數庫可供學生用來作基本的輸出、輸入。32-bit 的版本 (*Irvine32.lib*) 可在任何版本的 MS-Windows 的 Win32 主控台模式下使用。16-bit 的版本 (*Irvine16.lib*) 可在 MS-DOS、MS-Windows 及 Linux 的 DOS 模擬器下使用。在之後的章節之中，在這二個函數庫中的所有函數都會出現，而讀者可以按自己的希望去更改它們。了解到函數庫的使用只是為了方便起見是重要的，而不是不讓學生去學習如何自己撰寫輸入、輸出的程式。

□ 包含軟體及範例 所有的範例程式都已經使用 Microsoft Macro Assembler 6.15 版測試過了。這些程式大部份也都可以使用 Borland TASM 4.0 及 5.0 版來組譯，但有一些功能是 Borland 沒有完全支援的。

□ 網站資訊 本書的更新及修正可以在本書的網站之中找到，包含了提供給指導教授用作章節完成後可指派給學生的額外的專案作業：

<http://www.nuvisionmiami.com/books/asm>

如果你無法連上到這個網站，你可以在<http://www.prenhall.com>，搜尋本書的標題或作者全名「Kip Irvine」，以找到本書的資訊及原來網站上的資料。作者的電子信箱是kip@nuvisionmiami.com。

(譯註：real-address mode, 常被譯為真實模式、真實位址模式、真實定址模式及實體位址模式等)。

本書目標

以下的每一個本書的目標，都是被設計用來拓展學生在組合語言相關主題之中的興趣及知識：

- ◇ Intel IA-32 處理器架構及程式設計。
- ◇ 組合語言指引、巨集、運算子及程式結構。
- ◇ 程式撰寫方法，如何使用組合語言來設計系統層次軟體工具及應用程式。
- ◇ 電腦硬體控制運用。
- ◇ 組合語言程式、作業系統及其他應用程式之間的互動。

我其中的一個目標是幫助學生應付機器層概念的程式設計問題。將 CPU 想成是一個互動的工具是重要的，並且學著儘可能地去注意它的每一個動作。除錯器是一個程式設計者最好的朋友，不只是幫助我們抓到錯誤而已，而是個教學的工具，教導我們 CPU 及作業系統相關的知識。

除了一些簡短的範例之外，本書還包含超過 115 個可執行的程式檔案，用來展示本書中出現的指令或想法。像是簡介 MS-DOS 中斷及記憶指令的參考資料，可以在本書的最後面找到。有一個可以幫助學生們撰寫他們第一支程式、內容廣泛的連結函數庫，可以讓程式的使用者界面更易接受。本書中所包含的巨集函數庫，也能鼓勵教授及學生做更進一步的發展。

□ **需要的背景** 讀者們應該已經有能力及信心使用至少一種以上的程式語言來撰寫程式，最好是 Pascal、Java、C 或 C++。本書其中的一章有對 C++ 的介面做某種深度的討論，因此，手邊有其編譯器是非常有幫助的。我已經同時在主修為計算機科學及資訊管理的課程中使用過本書，並且它也已經被使用在其他工程相關的課程上。我是使用 Microsoft Visual C++ 6.0 及 Borland C++ 5.0 做為範例，以處理高階語言的介面。

特色

□ **完整程式列表** 書中所附的 CD-ROM 包含了本書中所有範例的原始碼。額外的程式列表可在作者的網站上找到。本書也附上了一個內容廣泛的連結資料庫，包含超過三十個程序，可用來簡化使用者輸出輸入介面、數值處理、磁碟及檔案處理及字串處理。在這個課程一開始的階段，學生可以使用這個連結函數庫來加強他們的程式。之後，他們可以建立自己的程序並將它們加入到連結函數庫中。學生可以得到完整的 16-bit 及 32-bit 連結函數庫的原始程式碼。

□ **程式邏輯** 有二個章節著重在布林邏輯及位元層次的運用。我們有意試著將高階語言的邏輯關係對應到機器上低階的細節，如此能幫助學生更有效地完成程式設計，並且更了解語言的編譯器如何產生物件編碼。

□ **硬體及作業系統觀念** 最前二個章節介紹基本的硬體及資料表示的概念，包含二進位數值、CPU 架構、狀態旗標及記憶體映射。審視電腦硬體及 Intel 處理器家族歷史，有助於學生更了解他們所要面對的電腦系統。

□ **結構化程式撰寫方法** 由第 5 章開始，程序及模組分解會被特別強調。學生會被交付更複雜的程式問題，那需要有更小心的結構能力，來處理他們的程式碼及問題的複雜性。

□ **磁碟儲存觀念** 學生會學到在 PC 上磁碟儲存系統時的基本原則，同時由硬體及軟體的觀點來看這些觀念。

□ **建立連結函數庫** 學生可以自由地加入他們自己的程序到本書的連結函數庫之中，並可以建立自己的連結函數庫。他們會學到使用一個叫工具箱的方法來撰寫程式及編寫程式碼，在面對一個以上的程式時，這是很好用的方法。

□ **巨集及結構** 有一章專門講解建立結構、聯合及巨集，這些對組合語言及其他高階語言都同等重要。進階的運算子條件式巨集會使得巨集更為專業。

□ **高階語言界面** 有一章節專門用來講解組合語言和 C 及 C++ 之間的溝通介面。這對可能想找高階語言相關的程式設計工作的學生而言，是一個很重要的工作技巧。他們可以學著最佳化他們的程式碼，並且可以看到 C++ 編譯器如何最佳化程式碼的正確實例。

□ **教學輔助** 所有的程式碼都可在書中的光碟及網站上找到。我們還提供指導教授一個考試題庫、所有自我評量的答案、程式練習的解答及各章節的 Microsoft PowerPoint 投影片資料。

講解順序

1 到 8 章的內容為組合語言的基礎，應該按照順序講解。本書試著盡最大的努力，平順地帶過這些章節。

1. 基本概念：組合語言的應用、基本概念、機器語言及資料表示法。
2. IA-32 微處理器架構：基本微處理器設計、指令執行週期、IA-32 處理器架構、IA-32 記憶體管理、微電腦組成元件及輸出入系統。
3. 組合語言基礎：介紹組合語言連結、除錯及定義常數與變數。
4. 資料傳輸、定址及算數：簡單的資料傳輸算數指令，組譯連結及執行週期、運算子、假指令、符號、JMP 及 LOOP 指令以及直接定址。
5. 程序：連結到外部函數庫、本書的連結函數庫說明、堆疊運算子、定義及使用程序、流程圖及由上而下的結構設計。
6. 條件處理：布林及比對指令、條件式跳躍及迴圈，高階語言邏輯結構及有限狀態機。
7. 整數算數運算：有用的平移及旋轉指令應用、乘法及除法、延伸加法及減法以及 ASCII 和縮緊十進制運算。
8. 進階程序：堆疊框、區域變數、參數宣告、遞迴及進階參數傳遞。

9 到 16 章所涵蓋的範圍並不按照特定順序，給予教授有機會去選擇適合他們課程的主題教授。

9. 字串及陣列：字串原理、字元及整數陣列、雙向陣列、排序及搜尋。
10. 結構及巨集：結構、巨集、條件組合語言假指令及定義重複區塊。
11. 32 位元視窗程式設計：保護模式記憶體管理及使用 Microsoft Windows API 在主控台中顯示文字及顏色。
12. 高階語言介面：參數傳遞轉換、內嵌組合語言編碼及連結組合語言模組到 C/C++ 程式。

13. 十六位元 MS-DOS 程式設計：在主控模式下及檔案輸入輸出下呼叫 MS-DOS 中斷處理。
14. 磁碟基礎：磁碟儲存系統、磁區、叢集、目錄、檔案配置表、MS-DOS 錯誤碼應用及磁碟機和目錄應用。
15. BIOS 層程式設計：鍵盤中斷、視訊文字、繪圖及滑鼠程式設計。
16. 進階 MS-DOS 程式設計：自訂區段、執行時期程式結構及中斷處理。
17. 進階主題：使用 I/O 埠進行硬體控制、指令編碼、浮點二進位表示法及浮點運算。

附錄 A：安裝及使用組譯器。

附錄 B：Intel 指令集。

附錄 C：BIOS 及 MS-DOS 中斷。

附錄 D：MASM 參考資料。

參考資料

在我自己的組合語言課程之中，我完全依賴教學用的教材，像是自學手冊、自我評量、電子幻燈片及工具書。在那樣子的心情之下，我已經試著提供教授持續不斷的支援。如果您發現某些重要的地方被遺漏了，請聯絡在下而我也許可以提供它們。以下的參考資料不是附在書後，便是在書中所附的光碟之中，或者在我的網站上可以找得到。

□ **組合語言工具書** 在光碟中所附的互動式工具書中，包了像是數值轉換、定址模式、暫存器使用方法、程式除錯及浮點二進位數值等的重要主題。內容頁面為 HTML 的文件，使得它可以讓學生或教授輕鬆地加上他們自己特定的內容。這個工具書也同時在我的網站上可以取得。

□ **除錯工具** 使用 Microsoft CodeView、Microsoft Visual Studio 及 Microsoft Windows Debugger (WinDbg) 的自學手冊。

□ **BIOS 及 MS-DOS 中斷** 附錄 C 中包含了簡單的列表，列出了最常使用的 INT 10h (視訊)、INT 16h (鍵盤) 及 INT 21h (MS-DOS) 中斷函數。

□ **指令集** 附錄 B 中列出了 IA-32 處理器家族中最不具特定權限的指令。對每一個指令，我們說明它的效果、語法及說明那個旗標被它所影響。

□ **PowerPoint 投影片** 取自我自己的課程講解用，完整的一套 Microsoft PowerPoint 投影片可以在指導教授網站上找到。

□ **自我評量的答案** 所有奇數題的自我評量解答可以在本書的網站上找到，偶數題的解答可以透過指導教授的網站中取得。

感 謝

特別感謝 Petra Recter, Prentice Hall 的資深計算機科學編輯，在選寫本書第四版之際，提供了不少有益的指引；出版編輯 Irwin Zucker 了不起的不斷地追蹤各個微小的細節；Bob Englehardt 在準備本書的光碟時提供了非常大的幫助；Camille Trentacoste 則是本書的主任編輯。

我要對以下三位教授，提出我特別的感謝與感激，他們鼓勵我的士氣，並且給予我在教學上的指示，及不知疲倦地檢查整本書：

- ◇ **Gerald Cahill**：來自 Antelope Valley 學院，提供了無數的建議及修正。
- ◇ **James Brink**：來自 Pacific Lutheran 大學，給予我許多非常好的建議。他的 32-bit 連結函數庫啟發了我來建立一個本書的函數庫。
- ◇ **Maria Kolatis**：來自 Morris County 學院，對我的章節提供尖銳的、深度的指教，讓我重新思考許多主題的表現方式。

另外，有三個人貢獻了他們大量的時間在對我的書做校閱，或啟發我一些範例：

- ◇ **Tom Joyce**：Premier Heart, LLC 的首席工程師。
- ◇ **Jeff Wothke**：來自 Purdue Calumet 大學。
- ◇ **Tim Downey**：來自 Florida International 大學。

幾個在佛羅里達國際大學我的頂尖學生讀了我的手稿，並且做出了寶貴的意見：Sylvia Miner、Eric Kobrin、Jose Gonzalez、Ian Merkel、Pablo Maurin 及 Hien Nguyen。Andres Altamirano 則為每個程式練習寫了非常棒的解答。

□ **校閱** 很感謝以下的幾位對每個單獨的章節做校閱的工作。除了有特別提及，所有的人都是從事教職工作：

- ◇ Courtney Amor，在 UCLA 的一個數學系學生。
- ◇ Ronald Davis，Kennedy-King 學院。
- ◇ Ata Elahi，Southern Connecticut State 大學。
- ◇ Leroy Highsmith，Southern Connecticut State 大學。
- ◇ Sajid Iqbal，Faran Institute of Technology。
- ◇ Charles Jones，Maryville 大學。
- ◇ Vincent Kayes，Mount St. Mary 大學，Newburgh，New York。
- ◇ Barry Meaker，設計工程師，波音公司。
- ◇ M. Nawaz，OPSTEC 大學資訊科學學系。
- ◇ Kam Ng，香港中國大學。
- ◇ Ernie Philipp，Northern Virginia Community 學院。
- ◇ Boyd Stephens，UGMO Research, LLC。
- ◇ Zachary Taylor，哥倫比亞大學。
- ◇ Virginia Welsh，Community College of Baltimore County。
- ◇ Robert Workman，Southern Connecticut State 大學。
- ◇ Tianzheng Wu，Mount Mercy 學院。
- ◇ Matthew Zukoski，Lehigh 大學。

微軟很慷慨地提供了它的巨集組譯器軟體附在本書之中。Helios Software Solutions Inc. 允許我加入評估版的 TextPad 編輯器。

最有希望的成功者，
不一定有很大的才幹，
卻是最能善用每一時機去開拓的人。

目錄

第 1 章 基本概念	
1.1 歡迎來到組合語言的世界	1-2
1.1.1 一些很好的問題	1-3
1.1.2 組合語言的各種應用	1-9
1.1.3 自我評量	1-11
1.2 虛擬機器的概念	1-12
1.2.1 個人電腦組譯器的發展史	1-15
1.2.2 自我評量	1-15
1.3 資料表示法	1-16
1.3.1 二進位數	1-17
1.3.2 二進位加法	1-20
1.3.3 整數儲存空間的大小	1-21
1.3.4 十六進位整數	1-22
1.3.5 有號整數	1-24
1.3.6 字元的儲存空間	1-28
1.3.7 自我評量	1-30
1.4 布林運算	1-33
1.4.1 布林函數的真值表	1-35
1.4.2 自我評量	1-37
1.5 本章總結	1-38
第 2 章 IA-32 微處理器架構	
2.1 基本概念	2-2
2.1.1 微電腦基本設計	2-3
2.1.2 指令執行週期	2-4
2.1.3 讀取記憶體	2-9
2.1.4 程式如何執行	2-11
2.1.5 自我評量	2-13

2.2 IA-32 處理器架構	2-14
2.2.1 運作模式	2-14
2.2.2 基本執行環境	2-15
2.2.3 浮點運算單元	2-19
2.2.4 Intel 微處理器沿革	2-20
2.2.5 自我評量	2-23
2.3 IA-32 記憶體管理	2-24
2.3.1 實體位址模式	2-25
2.3.2 保護模式	2-27
2.3.3 自我評量	2-30
2.4 IA-32 微電腦元件	2-31
2.4.1 主機板	2-31
2.4.2 視訊輸出	2-33
2.4.3 記憶體	2-33
2.4.4 輸出入埠	2-34
2.4.5 自我評量	2-36
2.5 輸出入系統	2-37
2.5.1 運作原理	2-37
2.5.2 自我評量	2-40
2.6 本章總結	2-41
第 3 章 組合語言基礎	
3.1 組合語言基本元件	3-2
3.1.1 整數常數	3-3
3.1.2 整數運算式	3-4
3.1.3 實數常數	3-5
3.1.4 字元常數	3-5
3.1.5 字串常數	3-6
3.1.6 保留字	3-6
3.1.7 識別碼	3-7
3.1.8 指引	3-7
3.1.9 指令	3-8
3.1.10 自我評量	3-12

3.2 範例：三個整數加法	3-13
3.2.1 程式碼	3-13
3.2.2 程式輸出	3-13
3.2.3 程式解說	3-14
3.2.4 程式範本	3-17
3.2.5 自我評量	3-18
3.3 組譯、連結及執行程式	3-19
3.3.1 組譯-連結-執行循環	3-19
3.3.2 自我評量	3-23
3.4 定義資料	3-24
3.4.1 內建資料型態	3-24
3.4.2 資料定義敘述	3-25
3.4.3 定義 BYTE 及 SBYTE 資料	3-25
3.4.4 定義 WORD 及 SWORD 資料	3-28
3.4.5 定義 DWORD 及 SDWORD 資料	3-29
3.4.6 定義 QWORD 資料	3-29
3.4.7 定義 TBYTE 資料	3-30
3.4.8 定義實數資料	3-30
3.4.9 小印地安排序	3-31
3.4.10 加入變數到 AddSub 程式	3-32
3.4.11 宣告未初始資料	3-33
3.4.12 自我評量	3-34
3.5 符號常數	3-35
3.5.1 等號指引	3-35
3.5.2 計算陣列及字串的大小	3-37
3.5.3 EQU 指引	3-38
3.5.4 TEXT EQU 指引	3-39
3.5.5 自我評量	3-40
3.6 實體位址模式程式設計(選讀)	3-41
3.7 本章總結	3-43
3.8 程式設計練習	3-45
第 4 章 資料轉移、定址與算術	
4.1 資料轉移指令	4-3

4.1.1	導論	4-3
4.1.2	運算元型態	4-4
4.1.3	直接記憶體運算元	4-5
4.1.4	MOV 指令	4-6
4.1.5	整數的補零/符號擴展	4-7
4.1.6	LAHF 及 SAHF 指令	4-10
4.1.7	XCHG 指令	4-10
4.1.8	直接位移運算元	4-11
4.1.9	範例程式(移動)	4-12
4.1.10	自我評量	4-13
4.2	加法與減法	4-14
4.2.1	INC 及 DEC 指令	4-14
4.2.2	ADD 指令	4-14
4.2.3	SUB 指令	4-15
4.2.4	NEG 指令	4-16
4.2.5	建立算術運算式	4-16
4.2.6	算術影響的旗標	4-17
4.2.7	範例程式(AddSub3)	4-20
4.2.8	自我評量	4-22
4.3	資料相關運算子及指引	4-23
4.3.1	OFFSET 運算子	4-24
4.3.2	ALIGN 指引	4-25
4.3.3	PTR 運算子	4-25
4.3.4	TYPE 運算子	4-27
4.3.5	LENGTHOF 運算子	4-28
4.3.6	SIZEOF 運算子	4-29
4.3.7	LABEL 指引	4-29
4.3.8	自我評量	4-30
4.4	間接定址	4-31
4.4.1	間接運算元	4-31
4.4.2	陣列	4-33
4.4.3	索引運算元	4-34
4.4.4	指標	4-36
4.4.5	自我評量	4-38

4.5	JMP 及 LOOP 指令	4-39
4.5.1	JMP 指令	4-39
4.5.2	LOOP 指令	4-40
4.5.3	整數陣列總合	4-42
4.5.4	字串複製	4-43
4.5.5	自我評量	4-44
4.6	本章總結	4-45
4.7	程式設計練習	4-47

第 5 章 程序

5.1	導論	5-3
5.2	連結外部函數庫	5-4
5.2.1	背景知識	5-5
5.2.2	自我評量	5-7
5.3	本書所用的連結函數庫	5-8
5.3.1	概觀	5-8
5.3.2	個別程序的描述	5-10
5.3.2.1	Irvine32.inc 標頭檔	5-18
5.3.3	函數庫測試程式	5-20
5.3.4	自我評量	5-24
5.4	堆疊運算	5-25
5.4.1	執行時期堆疊	5-26
5.4.1.1	PUSH 的運作	5-27
5.4.1.2	POP 的運作	5-28
5.4.1.3	堆疊的應用	5-28
5.4.2	PUSH 及 POP 指令	5-29
5.4.2.1	PUSH 指令	5-29
5.4.2.2	POP 指令	5-29
5.4.2.3	PUSHFD 及 POPFD 指令	5-30
5.4.2.4	PUSHAD、PUSHA、POPAD 及 POPA	5-31
5.4.2.5	範例：反轉一個字串	5-32
5.4.3	自我評量	5-33
5.5	定義及使用程序	5-34
5.5.1	PROC 指引	5-34

5.5.1.1	定義程序	5-34
5.5.1.2	範例：三個整數的和	5-35
5.5.1.3	程序的註解說明	5-36
5.5.2	CALL 及 RET 指令	5-37
5.5.2.1	呼叫及回傳的使用範例	5-37
5.5.2.2	巢狀程序呼叫	5-38
5.5.2.3	區域標籤及全域標籤	5-41
5.5.2.4	傳遞暫存器引數到程序中	5-42
5.5.3	範例：計算整數陣列的和	5-43
5.5.4	流程圖	5-44
5.5.5	儲存及回覆暫存器	5-46
5.5.5.1	USES 運算子	5-46
5.5.6	自我評量	5-48
5.6	使用程序設計程式	5-49
5.6.1	設計整數加法程式	5-50
5.6.1.1	整數加法程式實作	5-53
5.6.2	自我評量	4-56
5.7	本章總結	5-57
5.8	程式設計練習	5-59
第 6 章 條件處理		
6.1	導論	6-3
6.2	布林和比較指令	6-4
6.2.1	CPU 旗標	6-4
6.2.2	AND 指令	6-5
6.2.3	OR 指令	6-7
6.2.4	XOR 指令	6-9
6.2.5	NOT 指令	6-10
6.2.6	TEST 指令	6-11
6.2.7	CMP 指令	6-11
6.2.8	清除或設定個別的 CPU 旗標	6-13
6.2.9	自我評量	6-14
6.3	條件跳越	6-15
6.3.1	條件結構	6-15

6.3.2	Jcond 指令	6-16
6.3.3	條件跳越指令的類型	6-17
6.3.4	條件跳越的應用	6-20
6.3.5	位元測試指令(選讀)	6-26
6.3.6	自我評量	6-28
6.4	條件迴圈指令	6-29
6.4.1	LOOPZ 和 LOOPE 指令	6-29
6.4.2	LOOPNZ 和 LOOPNE 指令	6-30
6.4.3	自我評量	6-31
6.5	條件結構	6-32
6.5.1	區塊結構 IF 敘述	6-32
6.5.2	複合運算式	6-35
6.5.3	WHILE 迴圈	6-37
6.5.4	表格驅動選擇	6-40
6.5.5	自我評量	6-43
6.6	應用：有限狀態機器	6-44
6.6.1	驗證一輸入字串	6-44
6.6.2	驗證一有號整數	6-46
6.6.3	自我評量	6-50
6.7	使用 .IF 指引(選讀)	6-51
6.7.1	有號及無號的比較	6-53
6.7.2	複合運算式	6-54
6.7.3	.REPEAT 及 .WHILE 指引	6-57
6.8	本章總結	6-59
6.9	程式設計練習	6-61
第 7 章 整數算術運算		
7.1	導論	7-3
7.2	移位與迴旋指令	7-4
7.2.1	邏輯移位 V.S.算術移位	7-5
7.2.2	SHL 指令	7-5
7.2.3	SHR 指令	7-7
7.2.4	SAL 與 SAR 指令	7-8
7.2.5	ROL 指令	7-9

7.2.6 ROR 指令	7-10
7.2.7 RCL 與 RCR 指令	7-10
7.2.8 SHLD/SHRD 旗標	7-11
7.2.9 自我評量	7-13
7.3 移位與迴旋的應用	7-15
7.3.1 移位多個雙字組	7-15
7.3.2 二進位乘法	7-16
7.3.3 顯示二進位位元	7-17
7.3.4 分隔一個位元串	7-18
7.3.5 自我評量	7-19
7.4 乘法與除法指令	7-20
7.4.1 MUL 指令	7-20
7.4.2 IMUL 指令	7-22
7.4.3 DIV 指令	7-23
7.4.4 有號整數除法	7-25
7.4.5 實作算術運算式	7-28
7.4.6 自我評量	7-30
7.5 延伸加法與減法	7-31
7.5.1 ADC 指令	7-31
7.5.2 延伸加法範例	7-32
7.5.3 SBB 指令	7-34
7.5.4 自我評量	7-34
7.6 ASCII 與緊縮十進制(選讀)	7-36
7.6.1 AAA 指令	7-37
7.6.2 AAS 指令	7-38
7.6.3 AAM 指令	7-38
7.6.4 AAD 指令	7-39
7.6.5 緊縮十進制整數	7-39
7.7 本章總結	7-41
7.8 程式設計練習	7-43
第 8 章 進階程序	
8.1 導論	8-2
8.2 區域變數	8-3

8.2.1 LOCAL 指引	8-4
8.2.2 自我評量	8-6
8.3 堆疊參數	8-7
8.3.1 INVOKE 指引	8-8
8.3.2 PROC 指引	8-10
8.3.3 PROTO 指引	8-12
8.3.4 傳值或傳參考	8-14
8.3.5 參數分類	8-16
8.3.6 範例：交換兩個整數	8-16
8.3.7 疑難排解訣竅	8-18
8.3.8 自我評量	8-20
8.4 堆疊框	8-21
8.4.1 記憶體模式	8-21
8.4.2 語言指定符	8-23
8.4.3 直接存取堆疊參數	8-25
8.4.4 以傳參考方式傳遞引數	8-28
8.4.5 建立區域變數	8-30
8.4.6 ENTER 及 LEAVE 指令(選讀)	8-31
8.4.7 自我評量	8-33
8.5 遞迴	8-34
8.5.1 使用遞迴計算和	8-35
8.5.2 計算階乘	8-36
8.5.3 自我評量	8-39
8.6 建立多模組程式	8-40
8.6.1 範例：ArraySum 程式	8-40
8.6.2 自我評量	8-46
8.7 本章總結	8-47
8.8 程式設計練習	8-49
第 9 章 字串及陣列	
9.1 導論	9-2
9.2 字串基本指令	9-3
9.2.1 MOVSB、MOBSW 及 MOVSD	9-5
9.2.2 CMPSB、CMPSW 及 CMPSD	9-6

9.2.3	SCASB、SCASW 及 SCASD	9-10
9.2.4	STOSB、STOSW 及 STOSD	9-11
9.2.5	LODSB、LODSW 及 LODSD	9-12
9.2.6	自我評量	9-13
9.3	特選的字串程序	9-14
9.3.1	Str_compare 程序	9-15
9.3.2	Str_length 程序	9-16
9.3.3	Str_copy 程序	9-17
9.3.4	Str_trim 程序	9-18
9.3.5	Str_ucase 程序	9-20
9.3.6	自我評量	9-21
9.4	二維陣列	9-22
9.4.1	基底-索引運算元	9-22
9.4.2	基底-索引位移量	9-24
9.4.3	自我評量	9-26
9.5	整數陣列的搜尋和排序	9-27
9.5.1	氣泡排序	9-27
9.5.2	二元搜尋	9-30
9.5.3	自我評量	9-38
9.6	本章總結	9-39
9.7	程式設計練習	9-41
第 10 章 結構及巨集		
10.1	結構	10-2
10.1.1	定義結構	10-3
10.1.2	宣告結構變數	10-4
10.1.3	對映結構變數	10-5
10.1.4	範例：顯示系統時間	10-7
10.1.5	巢狀結構	10-10
10.1.6	範例：醉漢漫步	10-11
10.1.7	宣告及使用 union	10-15
10.1.8	自我評量	10-17
10.2	巨集	10-19
10.2.1	概觀	10-19

10.2.2	定義巨集	10-20
10.2.3	引用巨集	10-22
10.2.4	巨集範例	10-23
10.2.5	巢狀巨集	10-29
10.2.6	範例程式：Wrapper(包裝)	10-30
10.2.7	自我評量	10-32
10.3	條件組譯指引	10-33
10.3.1	檢查遺漏的引數	10-34
10.3.2	預設的引數之初始設定式	10-35
10.3.3	布林運算式	10-36
10.3.4	IF、ELSE 及 ENDIF 指引	10-36
10.3.5	IFIDN 及 IFIDNI 指引	10-38
10.3.6	特殊運算子	10-39
10.3.7	巨集函數	10-44
10.3.8	自我評量	10-47
10.4	定義重複區塊	10-49
10.4.1	WHILE 指引	10-49
10.4.2	REPEAT 指引	10-50
10.4.3	FOR 指引	10-51
10.4.4	FORC 指引	10-52
10.4.5	範例：鏈結串列	10-53
10.4.6	自我評量	10-56
10.5	本章總結	10-57
10.6	程式設計練習	10-59

第 11 章 32 位元視窗程設計

11.1	Win32 主控台程式設計	11-2
11.1.1	背景知識	11-3
11.1.2	Win32 主控台函數	11-7
11.1.3	主控台輸入	11-9
11.1.4	主控台輸出	11-13
11.1.5	讀取和寫入檔案	11-16
11.1.6	主控台視窗操作	11-23
11.1.7	控制游標	11-27

11.1.8	控制文字顏色	11-28
11.1.9	時間和日期的函數	11-30
11.1.10	自我評量	11-36
11.2	寫個圖形的視窗應用程式	11-37
11.2.1	重要的結構	11-38
11.2.2	MessageBox 對話方塊函數	11-40
11.2.3	WinMain 程序	11-40
11.2.4	WinProc 程序	11-41
11.2.5	ErrorHandler 程序	11-42
11.2.6	程式列表	11-42
11.2.7	自我評量	11-47
11.3	IA-32 記憶體管理	11-48
11.3.1	線性位址	11-49
11.3.2	分頁轉換	11-53
11.3.3	自我評量	11-55
11.4	本章總結	11-56
11.5	程式設計練習	11-58
第 12 章 高階語言界面		
12.1	導論	12-2
12.1.1	通用協定	12-2
12.1.2	自我評量	12-4
12.2	inline 組合語言碼	12-5
12.2.1	微軟 Visual C++ 的 __asm 指引	12-5
12.2.2	檔案加密範例	12-9
12.2.3	自我評量	12-13
12.3	連結到 C++ 程式	12-14
12.3.1	連結到 Borland C++	12-15
12.3.2	ReadSector 範例	12-17
12.3.3	範例：大的隨機整數	12-23
12.3.4	使用組合語言最佳化 C++ 程式碼	12-25
12.3.5	自我評量	12-33
12.4	本章總結	12-34
12.5	程式設計練習	12-35

第 13 章 十六位元 MS-DOS 程式設計		
13.1	MS-DOS 與 IBM-PC	13-2
13.1.1	記憶體組織	13-3
13.1.2	輸入輸出重新導向	13-5
13.1.3	軟體中斷	13-6
13.1.4	INT 指令	13-6
13.1.5	自我評量	13-8
13.2	MS-DOS 功能呼叫 (INT 21h)	13-9
13.2.1	選出的輸出函數	13-11
13.2.2	Hello World 程式範例	13-14
13.2.3	選出的輸入函數	13-15
13.2.4	日期/時間函數	13-20
13.2.5	自我評量	13-25
13.3	標準 MS-DOS 檔案 I/O 服務	13-26
13.3.1	關閉檔案控制碼 (3Eh)	13-30
13.3.2	移動檔案指標 (42h)	13-30
13.3.3	選出的函數庫程序	13-32
13.3.4	範例：讀取並複製一個文字檔	13-33
13.3.5	讀取 MS-DOS 命令結尾	13-36
13.3.6	範例：建立一個二進位檔案	13-38
13.3.7	自我評量	13-42
13.4	本章總結	13-43
13.5	程式設計練習	13-45
第 14 章 磁碟基礎		
14.1	磁碟儲存系統	14-2
14.1.1	磁軌、磁柱及磁區	14-2
14.1.2	磁碟分割區 (Volumes)	14-5
14.1.3	自我評量	14-7
14.2	檔案系統	14-8
14.2.1	FAT12	14-10
14.2.2	FAT16	14-10
14.2.3	FAT32	14-11
14.2.4	NTFS	14-11

14.2.5	主要磁碟區	14-12
14.2.6	自我評量	14-14
14.3	磁碟目錄	14-15
14.3.1	MS-DOS 磁碟目錄結構	14-16
14.3.2	MS-Windows 中的長檔名	14-19
14.3.3	檔案配置表(FAT)	14-22
14.3.4	自我評量	14-23
14.4	讀取及寫入磁區(7305h)	14-24
14.4.1	顯示磁區內容程式	14-26
14.4.2	自我評量	14-31
14.5	系統層檔案函數	14-32
14.5.1	取得磁碟閒置空間(7303h)	14-33
14.5.2	建立子目錄(39h)	14-36
14.5.3	移除子目錄(3Ah)	14-37
14.5.4	設定目前工作目錄(3Bh)	14-37
14.5.5	取得目前工作目錄(47h)	14-38
14.5.6	自我評量	14-38
14.6	本章總結	14-39
14.7	程式設計練習	14-41
第 15 章 BIOS 層程式設計		
15.1	導論	15-2
15.1.1	BIOS 資料區	15-2
15.2	鍵盤輸入 - 使用 INT 16h	15-4
15.2.1	鍵盤如何運作	15-5
15.2.2	INT 16h 功能	15-6
15.2.3	自我評量	15-13
15.3	視訊程式設計 - 使用 INT 10h	15-14
15.3.1	基本背景	15-14
15.3.2	控制色彩	15-16
15.3.3	INT 10h 視訊功能	15-19
15.3.4	函數庫程序範例	15-32
15.3.5	自我評量	15-33
15.4	使用 INT 10h 繪圖	15-34

15.4.1	INT 10h 像素相關功能	15-35
15.4.2	DrawLine 程式	15-37
15.4.3	迪卡兒座標程式	15-39
15.4.4	將迪卡兒座標程式轉換成螢幕座標	15-42
15.4.5	自我評量	15-43
15.5	記憶體映射圖形	15-44
15.5.1	模式 13h : 320X200, 256 色	15-44
15.5.2	記憶體映射圖形程式	15-46
15.5.3	自我評量	15-49
15.6	滑鼠程式設計	15-50
15.6.1	滑鼠 INT 33h 功能	15-50
15.6.2	滑鼠追蹤程式	15-58
15.6.3	自我評量	15-64
15.7	本章總結	15-65
15.8	程式設計練習	15-67

第 16 章 進階 MS-DOS 程式設計

16.1	導論	16-2
16.2	定義區段	16-3
16.2.1	簡化的區段指引	16-3
16.2.2	完整的區段定義	16-6
16.2.3	區段覆寫	16-11
16.2.4	結合各區段	16-12
16.2.5	自我評量	16-14
16.3	執行時間的程式結構	16-15
16.3.1	COM 程式	16-16
16.3.2	EXE 程式	16-18
16.3.3	自我評量	16-21
16.4	中斷處理	16-22
16.4.1	硬體中斷	16-24
16.4.2	中斷控制指令	16-26
16.4.3	撰寫自訂的中斷服務常式	16-27
16.4.4	TSR 常駐程式	16-30
16.4.5	應用 : No_reset 程式	16-31

16.4.6 自我評量	16-37
16.5 本章總結	16-38
第 17 章 進階主題	
17.1 使用 I/O 埠控制硬體	17-2
17.1.1 輸入/輸出埠	17-2
17.2 Intel 指令編碼	17-6
17.2.1 單位元組指令	17-8
17.2.2 立即運算元	17-8
17.2.3 暫存器模式指令	17-9
17.2.4 記憶體模式指令	17-10
17.2.5 自我評量	17-15
17.3 浮點運算	17-16
17.3.1 IEEE 二進位浮點運算表示法	17-16
17.3.2 指數	17-18
17.3.3 標準化尾數	17-19
17.3.4 建立 IEEE 位元表示法	17-20
17.3.5 十進位分數轉換為二進位實數	17-21
17.3.6 IA-32 浮點運算架構	17-23
17.3.7 指令格式	17-25
17.3.8 浮點程式範例	17-28

附錄

附錄 A. 安裝及使用組譯器	A-1
A.1 安裝本書所附的光碟	A-2
A.2 組譯連結 32-Bit 保護模式程式	A-3
A.2.1 為保護模式的程式除錯	A-4
A.2.2 make32.bat 批次檔	A-4
A.3 組譯連結 16-Bit 實體位址模式程式	A-6
附錄 B. Intel 指令集	B-1

B.1 導論	B-2
B.2.1 旗標	B-2
B.2.2 指令說明和格式	B-3
B.2 指令集	B-5
附錄 C. BIOS 及 MS-DOS 中斷	C-1
C.1 導論	C-2
C.2 PC 中斷	C-3
C.3 INT 21h 功能 (MS-DOS 服務)	C-5
C.4 INT 10h 功能 (視訊 BIOS)	C-12
C.5 INT 16h 功能 (鍵盤 BIOS)	C-13
C.6 INT 33h 功能 (滑鼠功能)	C-3
附錄 D. MASM 參考資料	D-1
D.1 導論	D-2
D.2 MASM 保留字	D-3
D.3 暫存器名稱	D-4
D.4 Microsoft 組譯器 (ML)	D-5
D.5 連結	D-8
D.6 CodeView 除錯器	D-11
D.7 MASM 指引	D-12
D.8 預先定義的符號	D-31
D.9 運算子	D-34
D.10 執行時期運算子	D-39