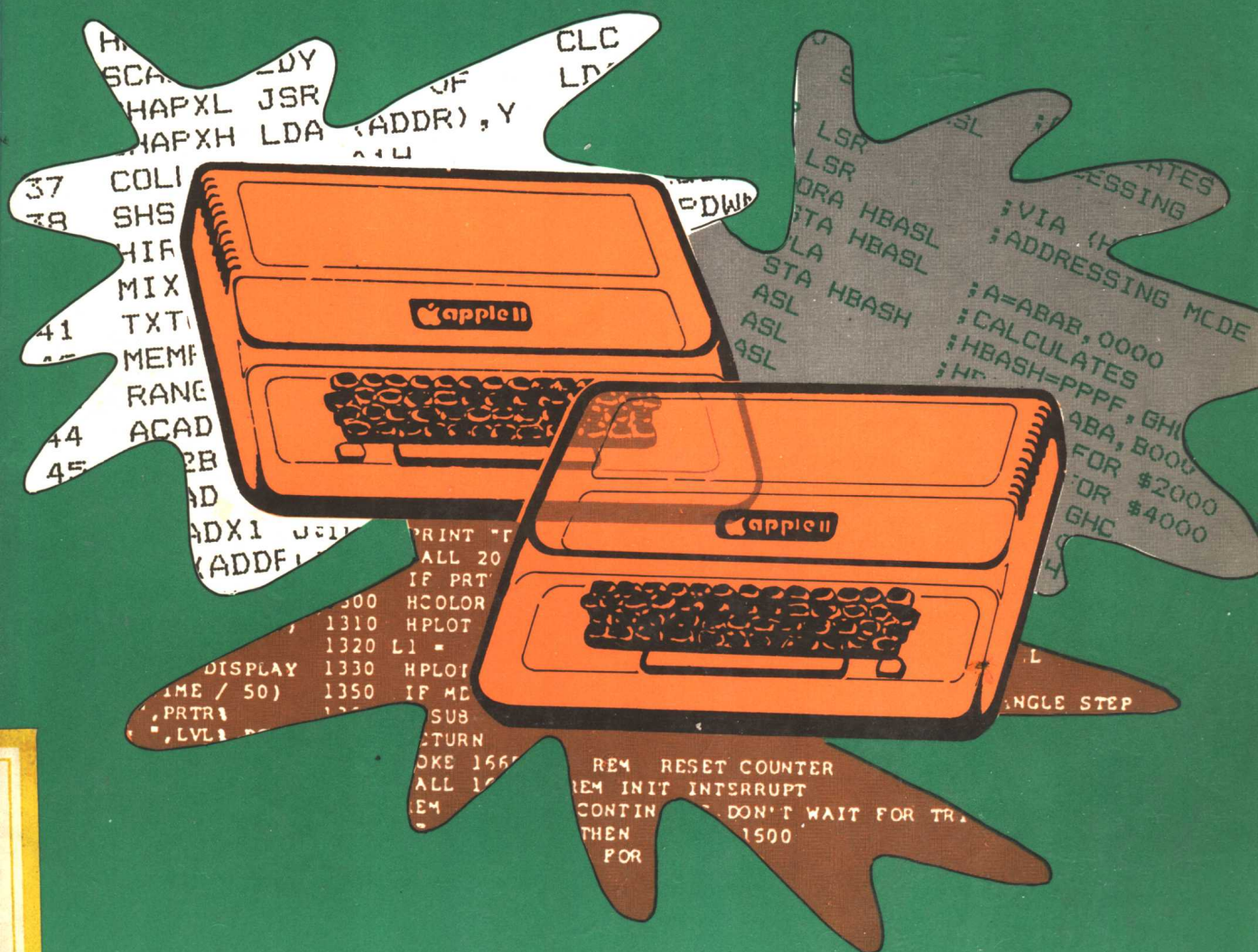


Apple組合語言 實習手冊

駱德廉 編著



全華科技圖書股份有限公司 印行



全華圖書 版權所有 翻印必究
局版台業字第0223號 法律顧問：陳培豪律師

APPLE 組合語言
實習手冊

駱德廉 編著

出版者 全華科技圖書股份有限公司
北市龍江路76巷20-2號
電話：581-1300·541-5342
581-1362·581-1347
郵撥帳號：1 0 0 8 3 6
發行人 陳 本 源
印刷者 華一彩色印刷廠
定價 新臺幣 220 元
再 版 中華民國73年2月

序 言

在科技急遽發展的時代中，電子計算機已成為人們所注意的焦點，各行各業的從業人員無不競相爭取學習有關的知識，以迎合今日社會上的需要。因此，在短短的幾年中，社會上已充斥了各形各類的電腦補習班和電腦公司，各家的報章雜誌亦發表了許多有關的報導和書籍，藉以提供給社會上各層面的學習者，為我國全面性的工業升級而努力。

近年來由於蘋果二號微電腦的影響，提升了社會各層面人士對學習電腦的興趣，而造成家用電腦空前未有的普遍性，幾乎每一個人都想擁有一套家用電腦，用以學習及處理自己身邊的事務。

因此，作者和全華圖書公司的總編輯商議，希以最淺顯的方式撰寫一些實習手冊，讓讀者能按步就班的一個實習一個實習的學習電腦，再由其中探知電腦內部的真正涵義。若要達到此真正的目的，作者認為讀者應首先去除怕學習低階語言的心裏，因為學習了低階語言之後就可以去除與電腦間的障礙，可提升大家在概念上的著眼點，更可充分地運用高低階語言來完成各項設計工作。

本書計分兩篇，第一篇為組合程式LISA的操作使用方法的介紹，籍以讓讀者能掌握LISA的各種命令來進行各種程式的設計工作。第二篇為二十個實習，期以系統化循序漸近的方式來介紹各種組合程式的設計方法及其設計目的，適於各層面之欲學習組合語言的讀者。

本書初版若有不當之處，敬請賜教！

編輯部序

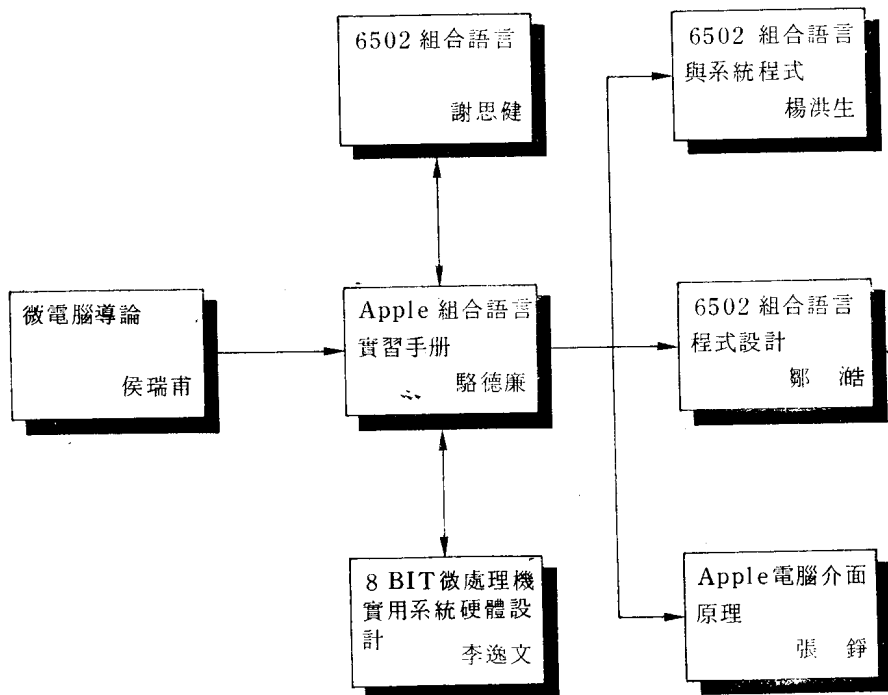
「系統編輯」是我們的編輯方針，我們所提供給您的，絕不只是一本書，而是關於這門學問的所有知識，它們由淺入深，循序漸進。

現在，我們將這本「Apple 組合語言實習手冊」呈獻給您。本書係針對Apple 組合語言的初學者所撰寫，以淺顯的說明和例題來指引初學者學習Apple 組合語言，探知電腦內部的真正涵義，可充分地運用高、低階語言來完成各種設計工作。文中並依據各種程式的實值意義來分類，使讀者能清晰明瞭組合語言的層次性與技巧性，進而完成各種程式的設計。

本書共分兩篇，第一篇專門介紹Apple 組合程式的LISA 使用方法，藉以讓讀者利用LISA 來設計組合程式語言。第二篇為20 個實習，用系統漸進方式來介紹Apple 組合語言，使其能明瞭各種程式設計的目的與技巧。

同時，為了使您能有系統且循序漸進研習組合語言方面叢書，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。

流程圖：



感謝您

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、質指資料新穎够水準，我們就是本著這個原則，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。



第一篇

Apple II 組合編輯程式LISA 的使用方法

第一章	組合編譯程式的命令操作	
1-1	啟動	3
1-2	操作命令	4
1-3	特殊字元	12
第二章	組合語言程式的敘述格式	
2-1	敘述格式	13
2-2	定址模式	14
2-3	局部指標	19
第三章	假指令使用方法	

第二篇

Apple II 組合語言實習

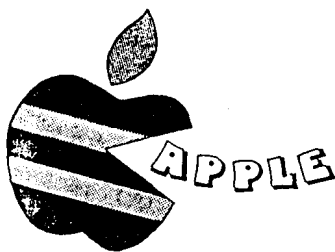
實習 1	資料傳送指令的練習	33
實習 2	算術運算／邏輯運算指令的練習	45
實習 3	跳躍指令與程式環路	57
實習 4	旋轉、移位指令與乘法程式	69
實習 5	副程式指令	83
實習 6	數學副程式	93
實習 7	碼的轉換	109
實習 8	表列與表格資料的搜尋	121

實習 9	監督程式 D4 用(一)	131
實習 10	監督程式 D4 用(二)	141
實習 11	繪圖程式的練習	153
實習 12	聲音產生程式的練習	177
實習 13	磁碟資料存取程式的設計	183
實習 14	磁碟作業系統的呼叫練習	197
實習 15	6502 微處理機硬體製作	217
實習 16	類比至數位轉換器的應用	229
實習 17	並行輸入／輸出界面的製作	235
實習 18	並行輸入／輸出界面的應用(一)	245
實習 19	並行輸入／輸出界面的應用(二)	253
實習 20	鍵盤電子琴之製作	263



第一篇

Apple II 組合編輯程式LISA
的使用方法



組合編譯程式 的命令操作

APPLE- II 組合編譯程式 LISA 是一種直接交談式 (interactive) 的組合語言，一旦程式由磁碟機載入記憶體之後，就可以獨立的直接與之交談訊息，無論編輯文字程式或組合翻譯工作或程式的修改與偵錯工作都是直接在主記憶體中進行的，等整個工作完成之後，才將文字檔案或工作碼檔案存放於磁碟機中做永久性的儲存。當然，有時程式撰寫到一半，爲了其它事情要暫時離開，亦可以將未完成的程式保存起來，下次再將之載入以進行未完成的工作。

所以在 LISA 中，磁碟機祇是負責一種後存資料的任務，程式的主要工作大都可主記憶體中進行，實在需要時亦可透過磁碟作業系統 (DOS) 來存取磁碟機中的資料。

1-1 啓 動

當我們擁有 LISA 程式時，一般以下列兩種方式來啓動之。

- (1) 在舊的版本中 (LISA 2.0)，以 “BRUN MXFLS <RETURN>” 命令來啓動之；在新的版本中 (LISA 2.5)，以 “BRUN LISA V2.5 <RETURN>” 來啓動之。
- (2) 在磁碟片的 HELLO 檔案中，以 BASIC 程式來執行 MXFLS 或 LISA V2.5 檔案，並在開機的時候直接啓動之。

啓動之後，LISA 程式即被載入記憶體中，且印出下列所示的訊息和交談標誌 “ !

”。

```
LAZER SYSTEM'S
INTERACTIVE
SYMBOLIC
ASSEMBLER
(C) 1981 LAZER SYSTEMS
VERSION 2.5 G - - 64 K
```

!

1-2 操作命令

在完成 LISA 的啟動工作之後，我們開始要輸入一個命令以便能編輯組合語言程式，在這種情況下我們首先輸入的命令是“ I < RETURN > ”。一旦“ I < RETURN > ”命令被輸入之後，會立即顯現行號，在開始時因沒有輸入任何資料，所以出現的行號為 1。此時，我們即可將組合語言程式逐步編寫進去。在編寫的過程中，語言必須要合乎語法的規定，否則其定然會出現一些訊息，用以警告程式設計者，並要求重新輸入這一行的程式，一直到語法沒有錯誤為止，然後行號自動增數，表示可以再輸入下一行的程式。

當我們的程式被全部編寫完之後，我們必須以 CTRL - E 鍵來跳出“ I ” (Insert) 命令控制，而重新進入命令層次（以“ ! ”為交談標誌的層次）。在命令層次中，除了“ I ”以外另有 14 種命令，可用來控制整個操作的過程，它們分別是：

```
N(EW)   LO(AD)   SA(VE)   W(RITE)   ^D(CTRL-D)
L(IST)  I(NSERT) D(ELETE) M(ODIFY) ^P(CTRL-P)
A(SM)   AP(PEND) LE(NGTH) BRK       F(IND)
```

在上面所示的命令中，我們祇需輸入括號前面的字，無需輸入整個字，LISA 即可判別命令為何種。下列我們將輔以例題來介紹各種命令的使用方法。

1. I(NSERT)

其命令格式為： I {line #}

在此，{line #} 用以代表一個行數，它是可選擇性的，如果不使用行數，則其會自動將資料插入原有資料的尾端。否則其會依所指定的行數來插入資料。

【例】

```
! I
1 LBL   LDA 00           插入新的資料兩行
2      END
```

<code>^E</code>		跳出插入 I 的控制
<code>! I 2</code>		插入第 2 行中
<code>2</code>	<code>STA 00</code>	插入的資料
<code>^E</code>		跳出插入 I 的控制
<code>! L</code>		表列所有的組合程式
<code>1</code>	<code>LBL LDA 00</code>	
<code>2</code>	<code>STA 00</code>	
<code>3</code>	<code>END</code>	
<code>!</code>		

注意，當我們用 `I {line #}` 命令插入新資料於某一行之後，其原來資料的行數會自動調整。

2. `D(ELETE) line #1 { , line #2 }`

這個命令用以將所指定範圍的行數資料刪除。

【例】

```
! D 2      將第 2 行刪除
! D 2,6    將第 2 至 6 行刪除
```

注意，這個命令和 `I` 命令一樣，在刪除資料之後，它會自動調整行數號碼。

3. `L(IST) {line #1 { , line #2 } }`

這個命令用以將所編輯完的資料表列出來，它的命令格式有三種情形：

- (1) `L` 用以表列所有的資料
- (2) `L line #1` 用以表列某一行資料
- (3) `L line #1 , line #2` 用以表列從 `line #1` 至 `line #2` 的資料

在表列多行資料的過程中，我們可以用 `CTRL-C` 來中止表列資料的動作，且將之恢復至命令層次中。而 `SPACE` 僅用以停止表列動作而已，另一個 `SPACE` 則又可繼續原先的表列動作。

4. `LO(AD) filename`

這個命令用以將 `LISA` 種類的檔案載入記憶體中。通常這個命令是與 `SA(VE)` 命令配合著使用的，用以將編輯好存檔的程式拿出，以便處理。

5. `SA(VE) filename`

6 第一篇 APPLE-I 組合編譯程式 LISA 的使用方法

這個命令用以將編輯或修改好的文字程式存入檔案中，而這個檔案的種類為“L”。

6. AP (PEND) filename

這個命令用以從磁碟機中讀入一個文字檔案，並將之附加於記憶體中原有資料後端。

7. A (SM)

這個命令用以將記憶體中的文字檔案組合翻譯成機器語言。如果在翻譯的過程中出現了錯誤，則 LISA 會顯現錯誤的訊息，並詢問是否要繼續進行翻譯的工作，此時你可以回答“C”表示要繼續，亦可以回答“A”表示要停止。

8. W (RITE) filename

這個命令用以將所編寫的程序存入磁碟檔案中，而這個檔案的類型為 TEXT。形成這種檔案的目的是爲了要讓 APPLE 中的文字編輯程式 PIE 能修改處理這些程序，亦能讓擅長使用 BASIC 或 APPLESOFT 的程序設計者處理 LISA 的程序。

使用 W 命令時，其會在第一行中加入“INS”的字於文字檔案中，如此使用者在 LISA 中可以用 DOS 的命令“EXEC”來將文字檔案重新載入記憶體中。注意，在 LISA 中可以使用 DOS 的命令，祇是在使用命令之前要先打一個 CTRL-D 方可。

9. LE (NGTH)

這個命令用以顯示 LISA 中文字檔案的長度，亦即是顯示文字檔案有幾行。

10. M (ODIFY) line #1 {, line #2}

這個命令實際上是以下列的順序來執行前面所述及的三個命令。

L (IST) line #1 {, line #2}

D (ELETE) line #1 {, line #2}

I (NSERT) line #1

所以正如同前面 L, D, I 命令的敘述一樣，M 命令可修改單獨一行的資料，亦可修改某個範圍中的資料。如果在使用 M 命令時未加行號，則其會將整個檔案先表列一次，再顯示 ILLEGAL NUMBER 的錯誤訊息。

11. N (EW)

這個命令用以清除整個文字檔案，其在清除之前會出現一些訊息來詢問操作者是否真的要將整個檔案清除掉。

12. BRK

這個命令用以從 LISA 中跳出，而進入監督程式的控制中。若要再從監督程式中跳回 LISA，可以用 CTRL-B 或 E000G 命令，亦可用 CTRL-C 或 E003G 命令，然而前者 (CTRL-B) 會將進入監督程式之前的文字檔案刪除，後者 (CTRL-C) 則不會。

13. F(IND)label

這個命令會促使 LISA 去搜尋文字檔案，將所有包含與 label 標記相同的行數印出。

14. AD(CTRL-D)

這個命令用以在 LISA 中執行 DOS 命令，如：

AD PR #n 打開輸出裝置。
 AD IN #n 打開輸入裝置。
 AD EXEC filename 將 W 命令寫入的檔案尋回，載入 LISA 的記憶體中

15. AP(CTRL-P)

這個命令用以促使 LISA 程式跳至 \$E009 的位址執行。通常，這 \$E009 位址中存放著一個 JMP 命令，其所指向的跳躍位址即是使用者的副程式位址，所以在輸入 CTRL-P 之後，它就會跳躍至使用者的副程式中執行。在執行完使用者的副程式之後，必須執行一個 RTS 指令，或 JMP \$E003 指令，它即會再次進入 LISA 的命令層次中。

注意：使用者副程式的使用位址在 \$9490 ~ \$95FF 之間。

【例題】

```
! I 1 START        LDA #$00        用 I 命令輸入程式
2                    STA $1000
3                    LDA #$01
4                    STA $1001
5                    LDA #$02
6                    STA $1002
7                    LDA #$03
8                    STA $1003
9                    LDA #$04
10                   STA $1004
11                   LDA #$05
12                   STA $1005
```



```
13          LDA  ##06
14          STA  $1006
15          LDA  ##07
16          STA  $1007
17          END
18          ^E
```

!

```
!L 1 START      LDA  ##00
2          STA  $1000
3          LDA  ##01
4          STA  $1001
5          LDA  ##02
6          STA  $1002
7          LDA  ##03
8          STA  $1003
9          LDA  ##04
10         STA  $1004
11         LDA  ##05
12         STA  $1005
13         LDA  ##06
14         STA  $1006
15         LDA  ##07
16         STA  $1007
17         END
```

用 L 命令查看表列

!D 11,12.

```
!L 1 START      LDA  ##00
2          STA  $1000
3          LDA  ##01
4          STA  $1001
5          LDA  ##02
6          STA  $1002
7          LDA  ##03
8          STA  $1003
9          LDA  ##04
10         STA  $1004
11         LDA  ##06
12         STA  $1006
13         LDA  ##07
14         STA  $1007
15         END
```

用 D 命令刪除程式

```
!SA TEST
BSAVE TEST,A$1800,L$0094
```

用 SA(VE) 命令將程式存入
TEST 檔案中

```
!N
CLEAR?Y
```

用 N 命令清除程式

```
!LOAD TEST
BLOAD TEST,A$1800
```

用 LO(AD) 命令重新尋回程式