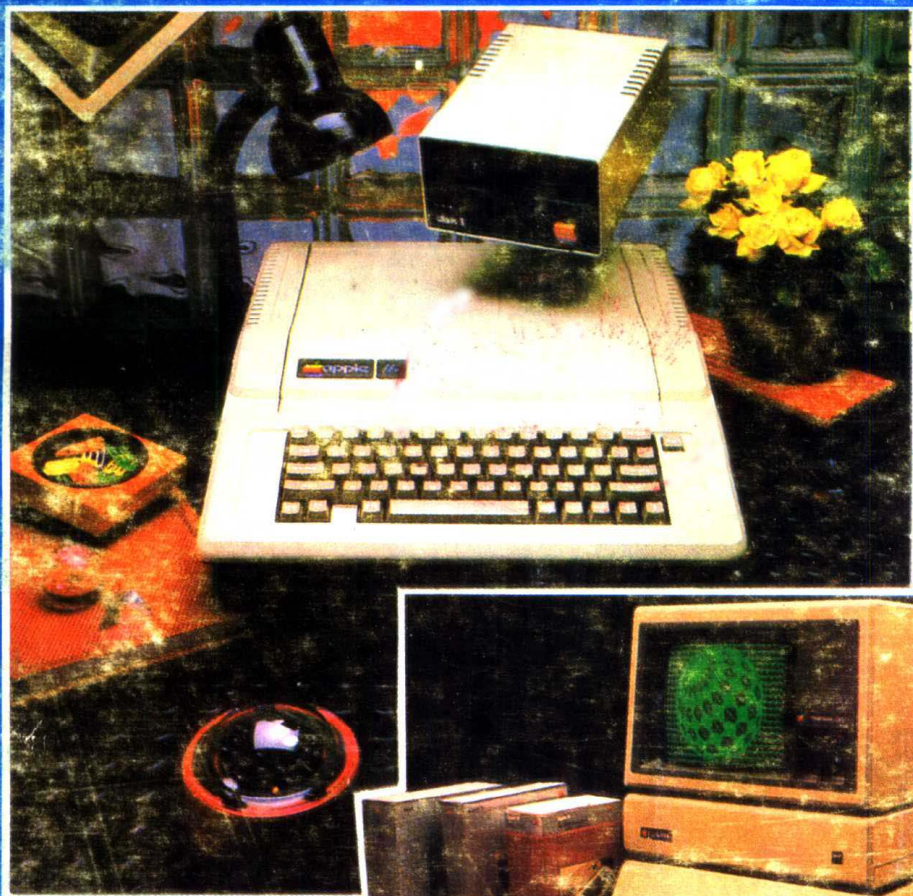


LISA 2.5 实用手册



资料组

APPLE LISA 2 5

實用手冊

北方电脑公司信息資料部

目 錄

第〇章	談談 LISA 及 PIE 的背景.....	1
第一章	前言.....	7
第二章	重要觀念.....	13
第三章	位址模式.....	19
第四章	LISA 的使用.....	33
第五章	組譯程式指令 / 虛擬運算碼.....	45
第六章	附加特點.....	73
第七章	進階知識.....	77
第八章	使用 LISA 的一些“怪招”.....	83
第九章	軟體.....	95
附錄 A	記憶體的应用.....	97
附錄 B	命令總結.....	99
附錄 C	錯誤訊息的解釋.....	101
附錄 D	標準 ASCII 字元表.....	107

附錄 E	ASCII 字元組	109
附錄 F	APPLE 螢幕顯示 ASCII 字元組	115
附錄 G	按鍵及其相關的 ASCII 碼	117
附錄 H	6502 指令及符號說明	119
附錄 I	MCS 650X 程式規劃模式	121
附錄 J	依字母順序排列的 6502 助譯符號碼 (即 6502 指令)	123
附錄 K	6502 運算碼 (即十六進位機器碼)	127
附錄 L	6502 指令定址法及其使用	133
附錄 M	使用者虛擬指令	153
附錄 N	XREF / 65 公用程式	157
附錄 O	SCTOLISA	159
附錄 P	雷射系統之反組譯程式 DISASM / 65	161
附錄 Q	SWEET-16 介紹	181
附錄 R	RANDY'S 提供的高解析度副程式	197
附錄 S	用 6522 來驅動 EPSON MX-80 列表機的應用程式 例	209
附錄 T	改變 APPLE 鍵盤功能的 MACRO 應用程式例	213
附錄 U	6522 REAL TIME CLOCK 應用程式例	219



第〇章 談談LISA及PIE的背景



LISA 係 Randall Hyde 以組合語言所完成的一個組譯程式，這裡所談的 LISA 是 Software，與 APPLE 公司幾年前開始發展，而於最近推出相當 powerful 的 Hardware “LISA” 機型祇是同名，據說 LISA 是取自 APPLE 公司設計者 Steven Wozniak 他女兒的名字呢！當初 Steven 與 Job 放棄哈佛大學不唸，賣掉車子創立 APPLE 公司把微電腦弄得紅透半邊天，這位 Steven 聽說還是個只吃水果的奇人呢，不知道是否只吃“蘋果”？今天台灣將近有十萬以上的人深受其利。

LISA 組譯語言，是初學組合語言者的福音，因為她是為初學者而設計的，曾經被一些專家及初學者使用在家庭中或辦公室中發展 6502 組合語言至少有四年以上時間。她的編輯及組譯都是在記憶體中同時來操作，因此所產生的原始程式碼可立即做組譯，如果組譯時沒有錯誤發生，可不必存檔，這只是對小程式而言，假設程式經組譯後只佔 4 K 長的目的碼，則原始程式檔不致有多大問題，在 48 K 系統中，保留給 user 用的 RAM 空間有 16 K 長，在 Language Card 系統中則保留給 user 可用的 RAM 空間有 26 K，特別是當組譯後的目的碼大於 4 K 時，程式師不能將檔案超過上述值，方

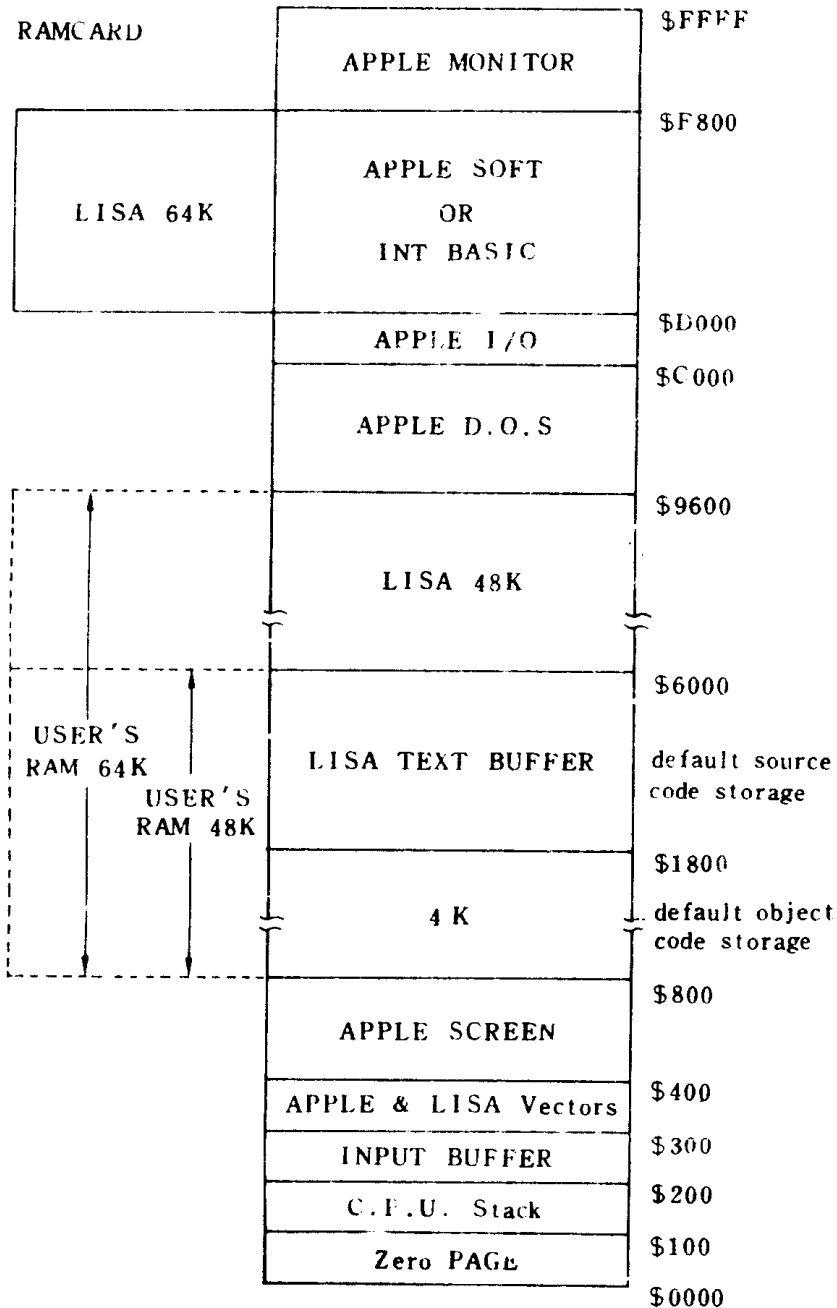
能確保每一個檔案能聯結 (linked) 虛擬運算碼，除了 LISA 1.5 以外，在 LISA 2.5 版本中不印出起始位址及組譯長度，因此 ORG 及 OBJ 虛擬運算碼必須小心使用以免 overwriting LISA 或原始碼。詳見下圖：

LISA 為何會這麼受歡迎，除了 LISA 命令很容易學會之外，其組譯速度快是一個主因，據 Hyde 宣稱大約每分鐘可以組譯 20,000 行，在 LISA 2.5 版本中可使用 30 個以上的虛擬運算碼（能提昇你的 APPLE II 變成一部 16 位元的微電腦，只需極少的指令就可以達成你所需的工作），其他的組譯程式就沒有這麼多。更多擴充的助譯符號碼（需有較大的記憶體）以及更多的命令讓你在 APPLE DOS 48K 中使用起來更具彈性。

在作編輯時，LISA 會掃描每一行敘述，馬上進入且與該行敘述 “ tokenizes ”（未來發展可作更多的擴充），這種方式的原理，有點類似 INTEGER BASIC 的解譯器，在編輯某行時，一有錯誤亦會馬上警告，這個結果當然有利有弊，在此不多贅述。

在作編輯的同時，LISA 將每一行的標名、指令、操作元等，儘可能的將之壓縮，所以 LISA 內有一些保留，常用的 LABEL，當編輯時若發現這種型的標名時，即以 TOKEN 取代之。因此，我們在編輯中寫入 LISA 的本文檔，就像 APPLE SOFT 一樣，被用代號取代；而在列印的時候，再經由 LISA 將之轉換成 LABEL 或助譯碼，如此，省下了許多的記憶體，這有些像 MICRO-SOFT 的 BASIC Interpreter 的做法。

由於所佔用的記憶空間縮小了，在目標工作（組譯）時，當然就省下了許多 SORTING 的時間，所以，LISA 的速度快的驚人。但是；有人還是喜歡用行編輯程式（LINE ORIENTED EDITOR



）來做本文檔編輯。程式小的時候，這問題還算小事，當程式大到某種程度時，LISA 本身的螢幕編輯實在很不好用，因此，能與組譯程式搭檔的本文檔編輯程式變得很需要。PROGRAMMA 公司也提供了 PIE 來供做 LISA 的本文檔編輯之用。

以兩個不同的程式來處理一個本文檔，其間本文檔（TEXT FILE 或 SOURCE FILE）的連結（LINKING）便非常重要。因此，LISA 同時也引用了 APPLE D.O.S 的幾個文檔命令，它可以用 WRITE 來將文檔寫入磁片，此時，在磁片上所存的文檔為標準 APPLE D.O.S 的“T”文檔，因此，幾乎可以將之傳輸給任何本文檔處理程式，APPLE PIE 就是一個很好用的本文檔編輯程式。

能夠寫出本文檔，就得具有讀回本文檔的能力，LISA 同樣的也可以經由 APPLE D.O.S 的 EXEC 命令讀回本文檔。

既然，她能讀回本文檔；如果覺得 LISA 的本文檔編輯能力不錯的話，也可以用她來編輯 BASIC 程式。

LISA 組譯程式提供三個公程式（Utilities）：（詳見附錄 N,O,P）（組譯程式亦稱組譯器）

(1) XREF/65：是一個 cross-reference tool for labels，她會將那些 label 在那幾行及那幾行用到這個 label 印出來。

(2) TRACE/65：是一個除錯工具。

(3) DISASM/65：是一個非交談式的反組譯程式，她可將 6502 目的碼反解回原始本文檔，同時 user 可以修改此原始本文檔。

至於 PIE 係用來做為軟體發展系統的編輯之用，可在 APPLE

II 及 APPLE II PLUS 電腦的 DOS 3.2 版本上執行，同時亦發售卡帶版的 PIE。

PIE 可用來 Edit BASIC 原始程式及 PROGRAMMA 公司出版的各種語言系統的原始程式，像 TINY PASCAL，ASM/65，LISA 2.5 及 CLS/65 Compiler。

PIE TEXT EDITOR 是幾年前由 Sunnyvale, California 的 SOFTWEST 的系統軟體顧問 Tom Crosley 在 SPHERE 6800 微電腦系統上首次設計出來，作者在寫 PIE 時曾在 APPLE II，Tany TRS-80，Exidy Sorcerer，及 SWTPc 6800 微電腦上執行過。

PIE 的觀念係建立在 Walter Bilofsky 的 NED Editor 上(原始的 NED 的觀念是目前在耶魯大學的 Edgar T. Irons 教授所構想出來。)，然後在 Bolt Beranek 及 Newman 公司的工程師協助下，再轉包到 Rand 公司的資訊科學部門贊助此計劃。

NED Editor 原先在 PDP-11 的 UNIX 操作系統下執行，UNIX 操作系統是貝爾實驗室的註冊商標，UNIX 以相當強的 C-Language 寫成的，目前部份 16 位元微電腦採用它，未來 32 位元微電腦採用 UNIX 操作系統將有著美好的遠景。

APPLE PIE Text Editor 由 SOFTWEST 公司的 Tom Crosley 執筆，並由 LISA 的作者 PROGRAMMA 公司的 Randall Hyde 撰寫 Command Processor 部份，PIE CONVERT 程式及 Configuration Modules 部份由 PROGRAMMA 公司的 Harry Tarnoff 執筆，PIE 的使用手冊則由 PROGRAMMA 公司的 Melvin L. Norell 與 Hauppauge, New York 的 Mike Weinstock 等人合力完成。

LISA 編輯及組譯中你若想在螢幕上看得更清楚，使註解欄不會被分割夾在每一行指令之間而混淆你的視覺，（因APPLE 螢幕通常每行只顯示40個字元）此時可加上 80 column Card，使得螢幕變成每行80個字元。

當然你要經由列表機列印出來時，通常列表機設定為每行印80字，亦常將太長的註解欄分割，這時可以將列表機輸出變為每行132個字元，其做法是先看列表機介面卡裝在那個slot上（假設裝在slot # 1），再送一個控制字元到列表機即可，如果你使用EPSON列表機，其做法如下：

```
| PR #1
```

```
] ?CHR$(15)
```



第一章 前 言



§ 1.0 前言

- (1) 什麼叫 LISA ?
- (2) LISA 需要怎樣的設備來配合其執行 ?
- (3) LISA 對 APPLE 監督程式及 DOS 提供何種界面方式 ?

§ 1.1 什麼叫 LISA ?

LISA (發音是 LI ZA , 而不是 LE SA) 係發展在 APPLE II 上作為 6502 交談式組合語言編譯程式之用。它係經過精心設計以適合於初學者或者是高級的程式設計師。由於它的結構、簡潔的語碼、交談式以及有著內藏式的功能, 因此 LISA 是所有提供給 APPLE II 使用者作為組合語言編譯程式上最有利的工具。

使用 LISA 來寫組合語言, 令機器語言程式設計變得像寫 BASIC 語言那樣地容易。正如其它的組合語言編譯程式一樣, LISA 能令你工作順利。因為 LISA 本身是一種符號式的組譯程式 (組合語言編譯程式, 爾後簡稱為組譯程式), 它不需要像內藏式的舊監督程式 (Old Monitor ROM) 上所提供的迷你組譯程式 (mini Assembler) 那樣, 需費勁的去記住各個位址。同時, LISA

比起其它在APPLE II 上所發展的組譯程式更具有豐富的內藏功能！有許多像EQU等虛擬運算碼（pseudo opcodes），這些能令你在寫組合語言時更加方便，像Sweet 16助譯符號（mnemonics），令你的APPLE II 變成一部16-bit（位元）的電腦，祇需極少的指令語碼就能完成所需的工作，有更多擴充的助譯符號，諸如BTR，BFL……等，以及更多的指令能更具彈性地使用DOS 3.2版本。

LISA 能用於新式的APPLE II PLUS 系統，像具有自動起始監督程式（Autostart ROM）或者具有語言卡界面的語言系統（Language System）中使用。如果你的APPLE II 電腦中具有雷射系統小寫字母的裝置（Laser Systems lower case adapter），則你可以鍵入並顯示出所有96個大／小寫的ASCII字元以及各種特殊符號，而且可以從未加特殊修飾裝置的APPLE 鍵盤上輸入。不僅如此，如果你需要以小寫字母加入你的組合語言程式中，上述的雷射系統將提供原始程式列印（source listing）的“LISA P2.L”副程式（使用LISA 2.5）令你更加方便。

§ 1.2 LISA 需要怎樣的設備來配合其執行？

LISA 係以磁碟建立的，使用時至少需要有一部磁碟機及48 K RAM記憶體。LISA 2.0 以上版本則需要一個語言卡以利適當的操作，當然沒有時也可操作，LISA 2.3係DOS 3.2版本的系統。雷射系統小寫裝置可有可無，有則更佳，在此建議採用。列表機（80行的最佳）亦可有可無，亦建議採用。另外由Programma公司所發行的APPLE PIE II Text Editing System（本文編輯系統）用於修飾較大的LISA 原始程式檔案上極為需要。如果你

祇是寫幾個 byte 的組合語言玩一玩，那麼 APPLE PIE 倒不必派上用場，因為 LISA 有它自己內藏的編輯能力，但是再度提醒，如果你是位行家而且要維護你的大程式，則 APPLE PIE 將甚具功能。

§ 1.3 LISA 提供怎樣的磁碟設備？

LISA 提供多個磁碟的取捨。使用者可以文字檔或二進位型式檔將 LISA 文字檔存到磁片內。二進位檔速度快且在磁片上所佔記憶空間較小，但與其他工具並不配合。文字檔可由 APPLE PIE 或 BASIC 程式讀入，但比起二進位檔則速度慢，然存取簡單。此外，在磁片上的 LISA 原始檔案可以利用“AP(PEND)”指令附加在記憶體已存檔案上。在組譯程式時可以從磁碟上使用“ICL”虛擬運算碼去“聯結”(Chain)原始檔案。如此當文字檔相當大過於 APPLE II 中可用的記憶體時，讓使用者亦可組譯此文字檔。像使用“DCM”虛擬運算碼，它可以儲存產生碼到磁片中，此種碼幾乎使任意長度皆可產生。LISA 提供卡帶的 load/save/append 使文字檔得以拷貝及交換。

§ 1.4 LISA 對 APPLE 監督程式及 DOS 提供何種界面方式？

LISA 可以在 DOS 3.2 或 DOS 3.3 版本下做檔案的維護及週邊控制的操作，任何 DOS 命令都可以在 LISA 的命令形態(Command level)下被直接使用。因為 PR# 及 IN# 都是 DOS 命令，同時 PR# 及 IN# 亦用於週邊控制。此外，CONTROL-P 係保留給使用者定義的副程式之用，這些副程式可能是利用 I/O 裝置作列表機的驅動用，而非建立在板子上的 ROM 來使用，或者係用

在GAME I/O 插座上的驅動裝置，甚至任何使用者定義的公用程式像慢速列印，進入BASIC等等。LISA 僅使用已存在APPLE 監督程式中的標準副程式，因此LISA 可以在標準的APPLE 監督程式（即舊監督程式）及自動起始監督程式兩者中工作。

LISA 修飾指標在DOS 3.2 版本中，當LISA 磁片被起動（Boot the DOS）時，它同時被儲存到記憶體內，此時BASIC 及TINY PASCAL 程式都不能使用。LISA 以特殊的“LISA”格式儲存原始程式檔。當你打出磁片目錄時，這些檔將以“L”檔型出現。當你在未加修改的DOS 上執行時，這些檔看來像二進位檔，但是它們都不可以被BLOAD或BRUN。LISA 2.5 係僅提供給DOS 3.3 版本。

```
! CATALOG
  CATALOG
DISK VOLUME 254
A 002 HELLO
B 002 LISA V2.5
B 050 ASM.CODE.1
B 016 ASM.CODE.2
B 037 MPTOLISA.CODE
L 050 RANDYS HIRES ROUTINES
L 044 HIRES.2
L 039 DISASM.TXT
L 038 DISASM.2
L 026 SCTOLISA.TXT
B 003 SORT
B 006 SCTOLISA
B 058 ASM.CODE
L 034 LISA P2.L
```

! CATALOG
CATALOG

DISK VOLUME 254

I 002 LISA V2.5 FLIPSIDE
L 044 TRACE.TXT
L 042 XREF.TXT
L 013 SORT.TXT
A 006 XREF
B 010 XREF.OBJ V2.0
L 079 LISA P3.L
B 014 TRACE
L 020 UTILITIES
L 024 TXTED/SWEET-16
L 064 SOFTWARE TOOLS
L 069 LISA P1.L
L 006 MXFLS.TXT
T 010 P6.TXT



第二章 重要觀念



§ 2.0 重要觀念

- (1)原始程式格式。
- (2)標示欄。
- (3)助譯符號欄。
 - a) 標準助譯符號。
 - b) 擴充助譯符號。
 - c) SWEET-16 助譯符號。
 - d) 虛擬運算碼。
- (4)運算元欄。
- (5)註解欄。

§ 2.1 組合語言原始程式格式

在LISA的原始程式敘述中，係以“自由格式”的形式輸入。

LISA的原始程式敘述採用下列格式：

標示 助譯符號 運算元 ；註解

每一個原始程式的敘述組成稱之為“欄”。分別為標示欄，助