

JEC

JEC 821 APPLE II

軟體作業系統

張憲中 編譯



JEC 821 APPLE II
軟體作業系統

張憲中 編譯

香港科技出版社

JEC 821 APPLE II

軟體作業系統

編 者：張 憲 中

出版者：香 港 科 技 出 版 社

發行者：香 港 科 技 出 版 社

九龍彩虹道810號六樓

印刷者：永達印刷公司

香港黃竹坑建明工業大廈九樓D座

定 價：H. K. ^{羅拔書屋} \$24.00

序

電子計算機之問世，掀起了二次工業革命；而近十年來，微電腦之急速發展，更使計算機廣泛地深入每個角落。硬體（Hard ware）製造技術之精進，不僅使微電腦之功能與可靠度大為提高，亦使價格大衆化，因而，未來數年之內，微電腦普及到每個家庭已是必然之勢；在美國，衆多型式微電腦之中，YSAPPLE最受歡迎，其最主要優點爲：價格低廉、體積小，操作簡單且軟體供應充足，無論是電腦專業人才或非電腦專業人才，只要按操作手冊去做，很快便能學會使用方法。

我們很希望這種風行全美的微電腦能迅速地推展到國內，從而對國內之資訊工業有所裨益，然而要使APPLE順利地在國內使用，則首先要克服語文上之障礙，此點對非電腦專業人才尤爲急切，因此，我們特將原文版之操作手冊予以轉譯成中文，並以淺顯之文字敘述之，使每個懂得中文的朋友都能毫無阻礙地學習這套性能卓越的微電腦。

筆者在譯著過程中，保持原著之精神，然後以口語化中文寫出，惟雖校稿數次，仍不免有所差錯，尚請諸位先進，不吝賜教。

11/15/86

目 錄

第一章 程式的起步	1
1—1 節 直接執行指令	1
1—2 節 延遲執行指令	1
1—3 節 數字形態	5
1—4 節 彩色圖形舉例	8
1—5 節 印刷形態	9
1—6 節 變數名稱	11
1—7 節 IF ... THEN	13
1—8 節 另一個彩色例子	16
1—9 節 FOR ... NEXT	17
1—10 節 數列	21
1—11 節 GOSUB ... RETURN	23
1—12 節 READ ... DATA ... RESTORE	25
1—13 節 實數、整數與字串變數	26
1—14 節 字串	28
1—15 節 彩色圖形的進一步討論	34
1—16 節 高解析彩色圖形	37
第二章 定義	43
2—1 節 句法之定義與縮寫	44
2—2 節 計算式的規則	54
2—3 節 形態的轉換	55

2—4 節	執行方式.....	55
第三章	系統和共用的指令.....	57
3—1 節	LOAD 和 SAVE	58
3—2 節	NEW	59
3—3 節	RUN	59
3—4 節	STOP, END, CtrlC, reset和CONT.....	60
3—5 節	TRACE 和 NOTRACE	62
3—6 節	PEER	63
3—7 節	POKE	63
3—8 節	WAIT	64
3—9 節	CALL	67
3—10 節	HIMEM	69
3—11 節	LOMEM	69
3—12 節	USR	71
第四章	編修程式及與形態有關的指令.....	72
4—1 節	LIST	73
4—2 節	DEL	75
4—3 節	REM	76
4—4 節	VTAB	76
4—5 節	HTAB	77
4—6 節	TAB	78
4—7 節	POS	78
4—8 節	SPC	79
4—9 節	HOME	80

4—10節	CLEAR	80
4—11節	FRE	81
4—12節	FLASH, INVERSE 和 NORMAL	81
4—13節	SPEED	82
4—14節	esc A, esc B, esc C 和 esc D	82
4—15節	repeat	83
4—16節	左箭號與右箭號	83
4—17節	Ctrl X	84
第五章	數列與字串	85
5—1節	DIM	86
5—2節	LEN	87
5—3節	STRS	88
5—4節	VAL	88
5—5節	CHR	89
5—6節	ASC	89
5—7節	LEFTS	90
5—8節	RIGHTS	91
5—9節	MIDS	91
5—10節	STORE 和 RECALL	93
第六章	輸入／輸出指令	98
6—1節	INPUT	99
6—2節	GET	101
6—3節	DATA	102
6—4節	READ	104

6—5 節	RESTORE	105
6—6 節	PRINT	105
6—7 節	IN #	107
6—8 節	PR #	107
6—9 節	LET	108
6—10 節	DEF FN	109
第七章	與控制流程有關的指令	112
7—1 節	GOTO	113
7—2 節	IF ... THEN 和 IF ... GOTO	113
7—3 節	FOR ... TO ... STEP	116
7—4 節	NEXT	117
7—5 節	GOSUB	119
7—6 節	RETURN	119
7—7 節	POP	120
7—8 節	ON ... GOTO 與 ON ... GOSUB	120
7—9 節	ONERR GOTO	121
7—10 節	RESUME	124
第八章	圖形與遊樂控制器	126
8—1 節	TEXT	127
8—2 節	GR	127
8—3 節	COLOR	128
8—4 節	PLOT	130
8—5 節	HLIN	131
8—6 節	VLIN	132

8—7 節	SCRN	132
8—8 節	HGR	134
8—9 節	HGR 2	135
8—10 節	HCOLOR	136
8—11 節	HPLOT	137
8—12 節	PDL	139
第九章	高解析模型 (HIGH-RESOLUTION SHAPES)	141
9—1 節	如何建立模型表	142
9—2 節	貯存模型表	151
9—3 節	模型表的使用	153
9—4 節	DRAW	154
9—5 節	XDRAW	155
9—6 節	ROT	156
9—7 節	SCALE	156
9—8 節	SHLOAD	157
第十章	一些數學函數	158
10—1 節	庫存函數 SIN, COS, TAN, ATN, INT, RND, SGN, ABS, SQR, EXP, LOG	159
10—2 節	導出的函數	161

附 錄

附錄A	建立APPLESOFT並予以執行	165
附錄B	程式的校訂	170
附錄C	錯誤訊號	179
附錄D	節省記憶空間的方法	183
附錄E	使程式執行速度加快的方法	186
附錄F	關鍵字與其十進制的代號	190
附錄G	APPLESOFT的預留字	191
附錄H	將BAS IC 程式轉換成APPLESOFT	192
附錄I	記憶圖	194
附錄J	PEEK , POKE 和CALL 等	197
附錄K	ASCII 字碼	212
附錄L	APPLESOFT 的第零頁使用情形	213
附錄M	APPLESOFT 和 Integer BAS IC之間的差別 ..	216
附錄N	句法的定義與縮寫	220
附錄O	APPLESOFT 指令的摘要	230

第一章 程式的起步

- 1-1 直接執行指令
- 1-2 延遲執行指令
- 1-3 數字形態
- 1-4 彩色圖形舉例
- 1-5 印刷形態
- 1-6 變數名稱
- 1-7 IF..THEN
- 1-8 另一彩色例子
- 1-9 FOR..NEXT
- 1-10 數列
- 1-11 GOSUB..RETURN
- 1-12 READ..DATA..RESTORE
- 1-13 實數、整數與字串變數
- 1-14 字串
- 1-15 彩色圖形的進一步討論
- 1-16 高解析彩色圖形

1-1 節 直接執行指令

試打入：

PRINT 10-4

然後再按 RETURN 鍵。

則您會看到 APPLESOFT II 直接印出

6

這個例中，在打入 RETURN 之後，APPLESOFT 便開始執行 PRINT 之後的工作 10-6，並將結果 6 予以印出。

再試一例：

PRINT 1/2, 3*10

(* 意為乘，/ 意為除)

按 RETURN 鍵之後，將會印出

.5

30

到此，您可發覺 APPLESOFT 除了減法之外還會執行乘除運算。注意，在 PRINT 之後的逗號 (,) 會使印出的字成為兩個而不是一個，此詳細之用法，將在第六章述及。

1-2 延遲執行指令

有些指令 (如上例的 PRINT) 直接打入便可得到結果，像這種指令稱為“直接執行”指令，但還有另一型式之指令，稱為“延遲執行”指令，這種指令都是以行數作為開端，行數，也就是第幾行之意，以 0 到 63999 的整數來表示。

第一章 程式的起步

例如，請試著打

```
1ϕ PRINT 2+3
```

```
2ϕ PRINT 2-3
```

(記住在打完每一行之後，別忘了打 RETURN 鍵)

一連串延遲執行指令的組合，便成了所謂的“ 程式 ”
(Program) 這些指令並不立即執行，而是存在 APPLE
的記憶單元之內，直到按下“ 執行 ” RUN 時，才按行數次
序，自最小到最大分別執行之。

假使現在按下 RUN，(別忘了在每一行之後要按
RETURN 鍵)

RUN

則在螢光幕上會出現

5

- 1

前例中，我們先打入第 10 行然後再打入第 20 行，但
打入的次序並不重要，存入記憶單元時，是按行數大小來排
列的，因此先打入第 10 行或先打入第 20 行都可以。

當我們想知道記憶單元裏現存有那些程式時，便打
LIST

則顯示：

```
1ϕ PRINT 2+3
```

```
2ϕ PRINT 2-3
```

4 JEC 821 與 Apple II 軟體作業系統

有時，需要刪去程式中的某一行，其方法爲：

在按 RETURN 鍵之後，打入所欲刪去那行的行數，如：打入

1 ϕ

LIST

則顯示：

2 ϕ PRINT 2-3

如此，第 1 ϕ 行便自程式中刪去了，並且無法再令其送回。若要插入新指令，比如此行的行數爲 1 ϕ ，那麼，只要打入 1 ϕ 與新的指令即可，其例爲：打入

1 ϕ PRINT 2*3

LIST

則顯示：

1 ϕ PRINT 2*3

2 ϕ PRINT 2-3

上例是說明當要把第 1 ϕ 行的內容更換時，是採用先刪去後插入之方法，除此之外，尚有一種更簡易的方法，只要打入行數 1 ϕ ，APPLESOFT 會自動地將原來資料刪去並以新資料代替，例：

打入：

1 ϕ PRINT 3-3

LIST

此時，程式變成：

1ϕ PRINT 3-3

2ϕ PRINT 2-3

由於程式依行數大小排列，而往往在每一行之間要插入一些新指令，所以行數間隔最好保持 1 ϕ 左右，以備新資料之插入。

如果要抹去現存在記憶單元內的程式時，打入 NEW 即可；

如果已完成一個程式的執行並準備進行另一個新程式的執行時，先要打 NEW，才不致於使新、舊兩程式混淆不清。

打入：

NEW

則APPLESOFT 會回答這個提醒用的符號：

]

現再打入：

LIST

則APPLESOFT 回答

]

表示前面的程式已不再存於記憶單元了。

1-3 數字形態

數值儲存在內部，其精密度是九位，因此當印出數值時，只印出九位，並且可以科學記號型式印出（也就是十的乘

6 JEC 821 與 Apple II 軟體作業系統

冪方式來表示)。

在APPLESOFT BASIC 之中，能精確處理的範圍是自 -1×10^{38} 至 1×10^{38} ，否則將會有錯誤的指示出現。不過在處理加法或減法時，有時會產生像 1.7×10^{38} 這麼大的數字而不致有錯誤的指示。一數的絕對值在小於 3×10^{-39} 時，則會自動地被視為 ϕ 來處理。在上述之限制範圍內，整數值可處理之範圍為 -32767 到 32767 。

當一數印出時，下列規則可決定其正確形態：

1. 若為負數，則負號 (-) 會印出。
2. 若其絕對值為自 ϕ 至 999999999 的整數，則會印成整數型式。
3. 若其絕對值大於等於 $. \phi 1$ 並且小於 999999999.2，則印成定點記號而沒有冪數。
4. 若數目不屬於 2.) 或 3.) 所述者，則以科學記號表示。

科學記號用來印出實數，其形態為：

SX.XXXXXXXXXESTT

其中X表示 ϕ 到 9 的整數。

S 仍此數的負號，若為正數則不加印任何記號，若為負數，則印 (-)。在小數點之前所印出的字，是非零的數字，其餘部份則印在小數點之後，如此共有八位，接着是 E，E 之後的數是表示 10^{ϕ} 的冪數，當然，冪數有正亦有負，正負號則表示在 S 的位置，至於 T T 兩位，則表示冪數之大小。小數點前面一位必是非零的數字，因此，一連串全是零的

數字是不會印出來的。

下列例子可清楚地看出來：

數 值	輸出型態
+1	1
-1	-1
6523	6523
-23.46 ϕ	-23.46
45.72E5	4572 $\phi\phi\phi$
1*1 $\phi^2\phi$	1E+2 ϕ
-12.34567896*1 $\phi^1\phi$	-1.2345679E+11
1 $\phi\phi\phi\phi\phi\phi\phi\phi\phi$	1E+ ϕ 9
999999999	999999999

在輸入鍵或程式之內，我們可有非常多位的數位，最多可至 3 8 位，但真正有效的只是前十位，過了第十位以後的數都會捨去不用。

例如，假使打入：

```
PRINT 1.23456787654321
```

那麼，僅印出：

```
1.23456788
```