

Modifichiamo il DOS

Dopo le precedenti modifiche che ci permettevano di richiamare un file con il nome abbreviato e che ci avvisavano se si tentava di salvare un file Applesoft con un nome già esistente, cancellando quindi la precedente versione, ancora due modifiche (le ultime lo prometto!) per semplificare un po' la vita anche a chi, pur non essendo un programmatore, usa tutti i giorni l'Apple per il suo lavoro.

La prima modifica consente di ottenere il CATALOG del dischetto con la pressione di un solo tasto (due col RETURN); per non stare troppo a smanettare le routine di input si è scelto il tasto & che è già di per sé un comando Applesoft.

La seconda modifica permette di lanciare in esecuzione un programma Basic con il tasto / (slash) seguito dal nome del programma; in pratica la / equivale perfettamente al comando RUN.

La prima utility si è ottenuta semplicemente scrivendo una piccola routine per accedere direttamente al FILE MANAGER, cioè quella parte del DOS che gestisce realmente il dialogo tra unità a disco e computer. La routine è lunga appena 5 byte, infatti deve solo caricare nell'accumulatore il codice 6, che per il FILE MANAGER è il comando del CATALOG, ed eseguire un salto alla routine di ingresso del FILE MANAGER.

Per ottenere il collegamento al comando & si è sfruttato il fatto che il DOS contiene una copia del contenuto delle locazioni che vanno da \$3D0 a \$3FF, locazioni che contengono i vari puntatori alle routine di RESET, NMI (interruzione non mascherabile), IRQ (richiesta di interruzione) e altre utility tra cui il comando &. In pratica, quando l'interprete Applesoft trova un comando & esegue un salto alla locazione \$3F4; qui, di solito, si trova un comando tipo JMP \$FF58, cioè salta alla locazione \$FF58 o in generale alla locazione il cui valore è scritto nelle locazioni successive all'istruzione di salto (JMP) che sono la \$3F5 (dec. 1014) e \$3F6 (dec. 1015). Se sostituiamo a \$FF58 il punto di entrata della nostra routine otterremo il CATALOG del disco in risposta al comando &; per ottenere automaticamente l'aggiornamento di questo valore al momento del lancio del dischetto con il DOS modificato, ci basta cambiare il contenuto delle locazioni corrispondenti a \$3F5 e \$3F6 nella copia che si trova all'interno del DOS.

Naturalmente il lancio di un programma che riassume la funzione del tasto & disabilita questa possibilità.

Il programma in Basic di figura 1 esegue la modifica del DOS ma non inizializza il

dischetto, può quindi essere usato come programmino a sé da aggiungere magari al programma di HELLO.

Per eseguire l'equivalenza "/" = "RUN" abbiamo dovuto rinunciare al comando CHAIN (tanto in Applesoft non funziona) in quanto non si poteva semplicemente cambiare la parola "RUN" in "/" perché si sarebbe persa la compatibilità con il comando RUN dato da programma (per es: PRINT D\$ "RUN PARTE.2").

Abbiamo invece sostituito "/" al comando CHAIN e abbiamo "imbrogliato" il DOS facendogli credere che CHAIN significhi anche lui RUN. Per fare ciò ci basta copiare nei puntatori del comando CHAIN quelli del comando RUN. Dato però che CHAIN è di cinque caratteri mentre / è uno solo abbiamo dovuto accorciare di quattro byte la lista dei comandi, aumentando di quattro locazioni i puntatori all'inizio lista e riscrivendo i primi cinque comandi. Il programma in Basic di figura 2 fa appunto ciò e presenta inoltre la lista dei comandi del DOS in cui si vede che al comando RUN segue / anziché CHAIN. Anche questo programma non effettua l'inizializzazione del dischetto, sicché è possibile usarlo anche indipendentemente. Se invece si preferisce avere una modifica permanente prendete un dischetto nuovo e, dopo aver fatto girare i due programmi, o

uno solo a scelta, scrivete un programmino di Hello e date il comando:

```
INIT HELLO
```

il nuovo dischetto avrà il DOS con le modifiche sempre presenti, e quindi automaticamente caricate al momento del lancio.

```
10 REM &=CATALOG
20 :
30 POKE 46989,169
31 POKE 46990,6
32 POKE 46991,76
33 POKE 46992,119
34 POKE 46993,162
35 :
40 POKE 40567,141
41 POKE 40568,183
42 :
43 POKE 1014,141
44 POKE 1015,142
45 :
```

Figura 2 - Listato del programma che rende il tasto / equivalente al comando RUN. Si può scrivere perciò /PROVA1 per lanciare in esecuzione il programma PROVA1.

Un po' di Apple-posta

Sto seguendo dal n. 20 gli articoli di Valter Di Dio sull'Assembler del 6502. Premetto per quel che viene dopo che possiedo un'Apple IIe.

Oltre agli articoli in sé, mi ha molto interessato la possibilità di "estrarre" (n. 23 e n. 18) il Miniassembler dell'Integer Basic.

La prima domanda che faccio è questa. Sarebbe possibile con la stessa tecnica estrarre l'AUTO dell'Integer e usarlo, come BAUTO, in Applesoft?

L'autonumerazione proposta da Di Dio, infatti, sul n. 14 è molto rigida, parte solo da 100 ed incrementa di 10 in 10; mentre l'Auto dell'Integer è molto più flessibile, qualunque inizio, qualunque incremento.

Le routine in assembler proposte finora quali l'Auto, appunto, la Superlist, il &Debug, il Renumber (Bo Arnkllit), vengono fatte partire da certe locazioni di memoria.

Come fanno a non interferire con i programmi che si scrivono in Basic; le locazioni indicate sono quelle generalmente "migliori" perché ciò non succeda?

Una scheda da 64K aiuterebbe a scrivere i programmi in Basic senza preoccupazioni di stare distruggendo, in caso di richiamo di una delle routine citate parte del lavoro già fatto? Se sì, come si modifica la mappa di memoria dell'IIe con la scheda aggiuntiva sopracitata?

```
100 TEXT : HOME
110 PRINT : PRINT "MODIFICA IL D
OS IN MODO DA AGGIUNGERE"
120 PRINT : PRINT "IL COMANDO /
EQUIVALENTE AL "RUN"
130 PRINT : INPUT " PROCEDO ? " ;
S$
140 IF LEFT$(S$,1) < > "S" THEN
END
180 HOME
1000 ST = 43144
1010 READ A$: IF A$ = "*" THEN 1
500
1020 I = 1:E = 0
1030 C$ = MID$(A$,I,1)
1040 IF I < LEN(A$) THEN C = ASC
(C$): GOTO 1060
1050 C = ASC(C$) + 128:E = 1
1060 POKE ST + I - 1,C
1070 I = I + 1
1080 IF NOT E THEN 1030
1090 ST = ST + I - 1: GOTO 1010
1500 REM
1520 POKE 40956,136
1530 POKE 43281,160
1540 POKE 40230,208
1550 POKE 40942,136
1590 HOME
1600 FOR I = 43144 TO 43270
1610 PRINT I,
1620 C = PEEK(I)
1630 PRINT CHR$(C);
1640 IF C > 127 THEN 1660
1650 I = I + 1: GOTO 1620
1660 PRINT : NEXT
1670 END
2000 DATA "INIT","LOAD","SAVE","
/","RUN","*"
```

Figura 1 - Listato del programma che abilita la funzione &=CATALOG.

Terza domanda e spero di non essere stato troppo lungo. Sarebbe possibile registrare le routine di cui sopra o altre su un'EPROM e richiamarle solo col nome, se sì, come si scrivono sull'Eprom una di seguito all'altra per non farle interferire col Basic e, infine, a che costi?

Sarebbe interessante, per concludere, che MC producesse un kit di tal genere con AUTO, RENUMBER, SUPERLIST, HARD COPY, &DEBUG ai soliti prezzi competitivi.

Grazie per l'attenzione ed auguri per l'ottima rivista.

Francesco Maraziti - Perugia

L'auto - Non è possibile utilizzare la routine di AUTONUMERAZIONE dell'INTEGER BASIC perché è per i programmi in INTEGER, si può invece modificare la mia routine di AUTO in modo da renderla un po' più flessibile. Per fare ciò basta aggiungere il programmino in linguaggio macchina di figura 3 in testa alla routine di AUTONUM o da qualsiasi altra parte; nell'esempio di figura 1 si è usata la pagina \$3. Per usare ora l'AUTO si effettua una CALL indirizzo, nn, ii dove: indirizzo è la locazione di partenza della routine che abbiamo aggiunto (per quella di figura 1 ind = 768), nn è il numero da cui vogliamo inizi la numerazione e ii è il valore dell'incremento. Se diamo 0 (zero) al valore dell'incremento si esce automaticamente dall'Autonumerazione.

Basic e Memoria - Il Basic non usa tutta la

RAM che trova presente nel computer (come fanno altri sistemi) ma solo la porzione compresa tra HIMEM e LOMEM. All'accensione il sistema mette LOMEM = 2051 e HIMEM = 49151; quando un programma che non sia in Basic ha bisogno di riservarsi una parte della RAM in modo che non venga "sporcata" dai dati o dai programmi del Basic, provvede a spostare verso il basso HIMEM di quel tanto che basta a contenere i dati e lo stesso programma in Linguaggio Macchina.

Un altro sistema è di spostare verso l'alto il punto di inizio dei programmi in Basic. Il metodo è il seguente: Detta LOC la locazione di inizio dei programmi in Basic si batte:

```
] POKE LOC - 1,0
] POKE 104, LOC/256
] POKE 103, LOC - PEEK(104)*256
] NEW
```

Da questo momento in poi tutti i programmi in Basic verranno caricati a partire da LOC, lasciando quindi libera tutta la RAM compresa tra 2049 e LOC.

Language Card - Non è possibile usare i 16K di RAM per i dati o i programmi in Basic in quanto la L.C. si sostituisce alle ROM dell'Applesoft. È invece possibile utilizzare questa memoria per i programmi in linguaggio macchina a patto che non usino routine del Basic. L'uso della L.C. è stato spiegato nel numero 29 nella rubrica della posta.

Eprom - È possibile scrivere alcune utility su una Eprom (anzi esiste già in commercio una eprom Programmer Aid) da 2K e la compatibilità è totale, dato che la Eprom si alloca nella zona riservata alle espansioni che va da \$C800 a \$CFFF. Non si possono chiamare le routine per nome, come se fossero dei programmi su disco, ma è possibile agganciarle al comando & per ottenere dei nuovi comandi tipo &AUTO, &RENUMBER, &SHOW ecc. Per quanto riguarda farne un kit ci penseremo.

*Spett. MC Posta,
salto volentieri la parte "preliminari e convenevoli".*

Chiedo se sia possibile rintracciare nella memoria dell'Apple II quale è il nome del programma che si trova momentaneamente residente in RAM.

Io so che è buona norma del perfetto programmatore quella di riportare sotto REM il nome del programma e qualche altra utile indicazione, ma non tutti lo fanno, ed anch'io spesso gli dò un nome ma nelle succes-

sive versioni questo subisce immancabilmente dei mutamenti (guai se no!).

Esiste qualche locazione in cui viene registrato il nome prelevato dalla directory o ci si deve servire di qualche artificiosa routine in L.M.?

Certo dovrebbe trattarsi di qualcosa di poco macchinoso altrimenti vale la pena di usare carta e matita nel comando immediato ed il MON.C.I.O nel comando differito ...

Salvatore Pinelli - Milano

È possibile sapere da programma il nome dell'ultimo File caricato dal disco, infatti per effettuare la lettura del file il DOS (Disk Operative System) ricopia in una propria area di lavoro il nome da cercare sulla Directory. L'area in questione si chiama FILE NAME PRIMARY BUFFER e, nel DOS 3.3. 48K, comincia all'indirizzo esadecimale \$AA75 e prosegue per trenta caratteri (appunto la lunghezza lecita dei nomi di file). Per sapere allora quale nome aveva l'ultimo file caricato basta andarlo a leggere nella F.N.P.B. con un programmino in Basic. Se il file è un Binario può essere utile conoscere anche l'indirizzo di partenza e la lunghezza; questi dati si trovano rispettivamente nelle locazioni \$AA72/73 e \$AA62/63.

Il programmino di figura 4 stampa appunto il nome, l'indirizzo iniziale e la lunghezza del file appena caricato.

Attenzione al fatto che un RUN senza alcun nome pulisce la F.N.P.B. prima di far partire il programma che si trova in memoria.

*300G					
0300-	20	BE	DE	JSR	\$DEBE
0303-	20	67	DD	JSR	\$DD67
0306-	20	52	E7	JSR	\$E752
0309-	A5	50		LDA	\$50
030B-	BD	F8	95	STA	\$95F8
030E-	A5	51		LDA	\$51
0310-	BD	F9	95	STA	\$95F9
0313-	D0	07		BNE	\$031C
0315-	A5	50		LDA	\$50
0317-	D0	03		BNE	\$031C
0319-	4C	4A	95	JMP	\$954A
031C-	20	BE	DE	JSR	\$DEBE
031F-	20	67	DD	JSR	\$DD67
0322-	20	52	E7	JSR	\$E752
0325-	A5	50		LDA	\$50
0327-	BD	FA	95	STA	\$95FA
032A-	A5	51		LDA	\$51
032C-	BD	FB	95	STA	\$95FB
032F-	4C	3A	95	JMP	\$953A
0332-	00			BRK	

Figura 3

```
100 REM ROUTINE DI LETTURA
110 A = PEEK (43634) + PEEK (43
      635) * 256
112 F = PEEK (46582)
115 L = PEEK (43616) + PEEK (43
      617) * 256
120 A$ = ""
130 FOR I = 43637 TO 43667
140 A$ = A$ + CHR$ ( PEEK ( I) )
150 NEXT
160 PRINT
170 PRINT "IL NOME E' ";A$
180 PRINT : IF F = 2 OR F = 130 THEN
      END
190 PRINT "L'INDIRIZZO E' ";A
200 PRINT : PRINT "LA LUNGHEZZA
      E' "L" BYTE"; CHR$ (83 * (L >
      1))
```

Figura 4

al servizio
dei tuoi problemi...



Azeta - Roma

...easy·byte ti suggerisce
il nome, le periferiche, il software
ed il prezzo.

RIVENDITORE AUTORIZZATO:

Apple Lisa Apple Macintosh Apple IIc
apple computer VICTOR apricot
olivetti M20 sinclair Commodore

DISTRIBUTORI
PREFERENZIALI
Verbatim

HOT-LINE è:
AOSTA - Informatique
BRESCIA - Il computer
MANTOVA - Antek Computers
RIMINI - Computer Center

ROMA/LATINA - Easy Byte
TORINO - AB Computer
TORINO - Cominfor
TRENTO - SI. GE. Computer Shop



risorse, idee e soluzioni.