A cura di Valter Di Dio

Catalogo parziale da Basic

In molti programmi che richiedono l'uso di file sul disco capita spesso di trovarsi di fronte alla fatidica domanda: "NOME DEL FILE..."

A questo punto si deve assolutamente rispondere con il nome completo ed esatto del file da caricare altrimenti il computer si 'incavola' e, a seconda del programma, o si blocca o risponde con un FILE NOT FOUND e ripropone l'infame domanda : "NOME DEL FILE..." riportandoci al punto di partenza. Alcuni programmatori un po' più comprensivi, in casi del genere, al primo errore ripropongono il catalogo del dischetto sullo schermo; ma se il file si chiama FHIT/12.04.85/XZURTE le possibilità di riscriverlo esattamente, senza averlo sullo schermo, sono abbastanza remote. Inoltre il catalog del dischetto presenta, oltre ai file utili, anche altri file che o non possono essere caricati perché di tipo diverso (ad esempio file R) o, seppure caricabili, si sovrappongono al programma in corso rovinando tutto il lavoro. Se un utente esperto riesce a cavarsela anche in casi del genere, lo stesso non si può dire di chi usa per la prima volta il programma o addirittura il computer.

Il problema si risolve invece facilmente con il seguente programma che permette di effettuare il catalog del disco scrivendo solo quei file che hanno certe caratteristiche: o sono Binari, o Text o contengono nel nome certi caratteri o una certa sottostringa come nel caso delle immagini in HGR che, in genere, contengono il prefisso PIC nel nome.

Il programma è particolarmente semplice; a parte la routine di lettura del disco che deve essere necessariamente in linguaggio macchina, tutto il resto è scritto in Basic. In questo modo risulta molto facile estrarre dal programma completo le sole parti che, di volta in volta, ci interessano.

Il programma di figura 1 effettua solo il catalogo parziale sul video, ma è facile fare in modo che i nomi dei file trovati vengano messi in un vettore stringa sì da poter anche eliminare il problema della riscrittura del nome quando questo, per la sua lunghezza, possa causare problemi all'utente.

In figura 2 trovate un programmino di prova che, se si risponde con un punto interrogativo alla domanda "NOME DEL FILE:" mostra, uno dopo l'altro, tutti i file di tipo PIC.nome presenti sul disco e carica automaticamente quello scelto. Per passare da un file all'altro si usano le frecce ← e → mentre il <RETURN> effettua la selezione. L'immagine scelta viene allora caricata e visualizzata.

Come funziona

Il catalog del disco si trova sulla traccia diciassette a partire dal settore 15. Con la routine RWTS, ampiamente spiegata nel manuale del DOS, andiamo a leggere il primo settore del catalogo e lo mettiamo in memoria a partire dalla locazione 16384; da qui, tornati al Basic, andiamo a prelevare il nome dei file e il tipo (binario, applesoft o text). Il significato del contenuto di un settore del catalogo è mostrato nella tabella 1, mentre il tipo del file è in tabella 2.

Se stiamo effettuando una ricerca per tipo controlliamo solo questo byte, altrimenti occorre controllare nel nome del file la presenza della sottostringa desiderata.

```
100  TEXT : HOME
110  PRINT : PRINT " PROGRAMMA CHE LEGGE NEL"
120  PRINT " CATALOGO DEL DISCO TUTTI"
130  PRINT " I FILE DI UN CERTO TIPO."
140  PRINT : PRINT "TIPO DI FILE:"
150  PRINT "1 - TEXT FILE (T)
160  PRINT "2 - BINARIO (B)"
170  PRINT "3 - WORD TEXT (TEXT.)"
180  PRINT "4 - PER CHIAVE (CHR$)"
190  PRINT : INPUT "     QUALE? ";T
200  IF T = 1 THEN TF = 128: GOTO 330
210  IF T = 2 THEN TF = 132: GOTO 330
220  IF T = 3 THEN C$ = "TEXT.":LL = 5:PC = 1: GOTO 330
230  IF T > 4 THEN  END
240  PRINT : PRINT " CHIAVE DI RICERCA: ";C$:LL =  LEN (C$)
250  PRINT : INPUT " POSIZIONE NEL NOME (1-30) :";PC
260  IF PC < 1 OR PC > 30 THEN 100
270 A$ = C$: PRINT
280  PRINT "     [:.....:.....:.....:]"
290  VTAB  PEEK (37): HTAB PC + 6
300  PRINT C$: PRINT
310  PRINT : INPUT " VA BENE ? (S/N) ";A$
320  IF A$ = "N" THEN 100
330  FOR I = 768 TO 803
340  READ A: POKE I,A
350  NEXT
360  DATA 169,3,160,10,32,217,3,96,0,0,1,96,1,0,17,15,32,3,0,64,0,0,1
     ,0,0,96,1,0,0,0,0,0,0,1,239,216
370 TL = 17:SL = 15:MO = 49385
380  POKE 782,TL: REM  TRACCIA
390  POKE 783,SL: REM  SETTORE
400  CALL 768: POKE MO,0:F =  FRE (0)
410  IF  PEEK (791) THEN  PRINT "ERRORE IN LETTURA ": END
420 TL =  PEEK (16385)
430 SL =  PEEK (16386)
440 N = 16395
450  IF  PEEK (N) = 255 THEN 560
460  IF  PEEK (N) +  PEEK (N + 1) = 0 THEN 580
470 F =  PEEK (N + 2): IF F < 128 THEN F = F + 128
480 A$ = ""
490  FOR I = 3 TO 32
500 A$ = A$ +  CHR$ ( PEEK (N + I) - 128)
510  NEXT
520  IF TF = F THEN 550
530  IF PC = 0 THEN 560
540  IF  MID$ (A$,PC,LL) <  > C$ THEN 560
550  PRINT A$
560 N = N + 35: IF N < 16630 THEN 450
570  GOTO 380
580  POKE MO - 1,0
```

Figura 1 - Programma in Basic Applesoft che consente di effettuare un catalogo parziale del disco visualizzando solo i file di un certo tipo.

```
100  TEXT : HOME : DIM CF$(30):D$ =  CHR$ (13) +  CHR$ (4)
110  INPUT "NOME DEL FILE:";FL$
120  IF FL$ <  > "?" THEN 480
130 C$ = "PIC.": REM  PREFISSO
140 PC = 1: REM  POSIZIONE NEL NOME
150 LL =  LEN (C$)
160  FOR I = 768 TO 803
170  READ A: POKE I,A
180  NEXT
190  DATA 169,3,160,10,32,217,3,96,0,0,1,96,1,0,17,15,32,3,0,64,0,0,1
     ,0,0,96,1,0,0,0,0,0,0,1,239,216
200 TL = 17:SL = 15:MO = 49385
210  POKE 782,TL: REM  TRACCIA
220  POKE 783,SL: REM  SETTORE
230  CALL 768: POKE MO,0:A =  FRE (0)
240  IF  PEEK (791) THEN  PRINT : PRINT "ERRORE IN LETTURA ";H = 0: GOTO
     370
250 TL =  PEEK (16385)
260 SL =  PEEK (16386)
270 N = 16395
280  IF  PEEK (N) = 255 THEN 350
290  IF  PEEK (N) +  PEEK (N + 1) = 0 THEN 370
300 A$ = ""
310  FOR I = 3 TO 32
320 A$ = A$ +  CHR$ ( PEEK (N + I) - 128)
330  NEXT
340  IF  MID$ (A$,PC,LL) = C$ THEN CF$(H) = A$:H = H + 1
350 N = N + 35: IF N < 16630 THEN 280
360  GOTO 210
370  POKE MO - 1,0
380  IF H = 0 THEN  PRINT : PRINT "NESSUN FILE DI QUESTO TIPO": END
390  VTAB 10: HTAB 1: PRINT  SPC( 30)
400  VTAB 10: HTAB 1: PRINT CF$(HH)
410  GET A$: IF A$ =  CHR$ (13) THEN 470
420  IF A$ =  CHR$ (8) THEN HH = HH - 1
430  IF A$ =  CHR$ (21) THEN HH = HH + 1
440  IF HH < 0 THEN  PRINT  CHR$ (7):HH = 0
450  IF HH = H THEN  PRINT  CHR$ (7):HH = H - 1
460  GOTO 390
470 FL$ = CF$(HH)
480  PRINT D$"BLOAD"FL$",A$4000"
490  POKE  - 16304,0: POKE  - 16299,0
500  POKE  - 16302,0: POKE  - 16297,0
```

Figura 2 - Esempio di utilizzo della routine che estrae dal catalog solo determinati file per scegliere dal disco un file di tipo PIC.

Tabella 1

Catalogo del disco: pista 11

Byte	contenuto
0	Non usato
1	Link: pista successiva
2	Link: settore succ.
3..A	non usati
B..2D	file 1: indirizzario
2E..50	file 2 "
51..73	file 3 "
74..96	file 4 "
97..B9	file 5 "
BA..DC	file 6 "
DD..FF	file 7 "

Indirizzario

Byte (relativo)	contenuto
0	pista del file (FF=cancellato).
1	settore del file.
2	Tipo del file.
3..20	nome del file.
21..22	lunghezza del file in settori.

Se la troviamo stampiamo il nome del file (o lo depositiamo in un vettore) altrimenti si passa al nome successivo.

Una volta terminato il primo settore del catalogo si passa al successivo e così via fino all'ultimo settore che contiene 0,0 come Link al settore seguente.

Due parole ancora su una strana poke presente alla riga 400. Come valore riporta 49385 che in esadecimale corrisponde alla locazione $C0E9; questa locazione cade all'interno di quelle destinate alla scheda di interfaccia del disco e, ogni volta che viene indirizzata, fa partire il motore del drive. In questo modo si evita che, rientrati nel Basic, il motore del drive si arresti, dimodoché alla prossima chiamata della RWTS (CALL 768), essendo già in movimento il disco, si risparmia tempo.

Per fermare definitivamente il motore del drive basta, al momento opportuno, effettuare una poke nella locazione 49384 (motor off). MC

Tabella 2

Tipo del file

00	TEXT
01	INTEGER
02	APPLESOFT
04	BINARY
08	RELOCATABLE
10	S (n.u.)
20	A (n.u.)
40	B (n.u.)
+80	file LOCKED

Adventure per tutti - n.38

Chi ha provato a giocare con il programma di Adventure per Apple pubblicato nel numero di febbraio, avrà notato che si creano degli inconvenienti qualora si giochi più di una volta consecutivamente un'avventura che contenga verbi che non possono essere eseguiti più di una volta, perché non vengono azzerate le relative variabili. Mancavano in effetti due righe, prontamente inviateci dallo stesso autore, Guglielmo Nigri. Le pubblichiamo qui sotto e ci scusiamo per l'accaduto.
(Nota: vanno aggiunte al Player pag. 120/121).

```
1481 FOR I=1 TO NV
1482    EX(I)=0
1483 NEXT I
```

Le routine dell'Applesoft

Inizia con questo riquadro una serie di cartelle che spiegheranno in modo dettagliato il funzionamento e l'uso delle principali routine in L.M. dell'interprete Applesoft. Ciascuna cartella sarà composta da una tabellina contenente il nome della routine, l'indirizzo in memoria e tutti i parametri che la routine usa per ricevere i dati e restituire i risultati. Le routine saranno perciò viste come delle scatole nere di cui non è necessario sapere come sono fatte dentro, ma solo cosa fanno.

La prima routine che analizziamo è piuttosto stupida, infatti fa pochissime cose, ma è nello stesso tempo il cuore del Basic ed è richiamata praticamente da tutte le altre routine dell'interprete. Proprio per questo motivo la routine risiede in pagina Zero, in quella pagina cioè in cui il microprocessore 6502 lavora alla massima velocità.

La routine si chiama GETCHAR (prendi un carattere) e restituisce nell'accumulatore il carattere (un byte preso dalla memoria) immediatamente successivo a quello preso quando era stata chiamata l'ultima volta. La locazione di memoria da cui preleverà il carattere si legge in due byte che sono interni alla routine in quanto, per risparmiare tempo e memoria e soprattutto per non modificare il contenuto di nessun altro registro oltre l'Accumulatore, la routine è stata scritta in modo automodificante.

GETCHAR $00B1

Parametri in entrata	parametri in uscita
$B8 e $B9 (facoltativi) puntano alla locazione precedente al carattere da prelevare	idem
Registri in entrata	**in uscita**
Accumulatore qualsiasi	carattere in ($B8,$B9)
Reg. X "	non modificato
Reg. Y "	non modificato
STATUS "	come da LDA
Stack point "	non modificato
ENTRY alternative	$B7 : riprende lo stesso carattere

Note: Il Carry settato indica il prelievo di un carattere non numerico.
Il contenuto del puntatore $B8,$B9 viene incrementato prima di prelevare il carattere dalla memoria.

Descrizione

GETCHAR - Preleva il carattere contenuto nella locazione successiva a quella puntata da $B8, $B9 (B8 parte bassa e B9 parte alta) e lo mette nell'Accumulatore. Al ritorno dalla routine $B8 e $B9 puntano alla locazione da cui è stato prelevato il carattere. Il Carry è settato se il carattere prelevato è alfabetico. Tutti gli altri flag seguono le normali regole di un LDA.

$00B7 - È un altro entry point della getchar solo che, scavalcando le prime istruzioni che incrementano $B8 e $B9, restituisce lo stesso carattere precedentemente prelevato. Si usa di solito per recuperare il valore originale quando, per svariati motivi, lo si sia dovuto modificare.

Esempio

```
300:A9 FF      INIZIO  LDA #$FF
302:85 B8              STA $B8
304:A9 01              LDA #$01
306:85 B9              STA $B9
308:20 B1 00   LOOP    JSR $00B1
30B:20 ED FD           JSR $FDED
30E:C9 8D              CMP #$8D
310:D0 F6              BNE LOOP
312:60                 RTS
```

Stampa il contenuto del buffer di riga.