VIC da zero +64

a cura di Tommaso Pantuso

Molti di voi, possessori di un drive 1541 della Commodore, saranno curiosi di conoscerne il funzionamento e i segreti. Questa e la prossima puntata di VIC da zero+ 64 saranno dedicate a questo (interessante, crediamo) argomento.

Il 1541

di Luigi Tavolato

La formattazione

Certamente moltissimi di coloro che in questo momento ci stanno leggendo, avranno usato in precedenza, per salvare o caricare i propri programmi, il registratore CN2 e si saranno abituati al suo caratteristico modo di operare, modo che non presuppone alcuna «preparazione» preliminare del nastro per predisporlo a ricevere i dati.

Niente di strano quindi se l'operazione di «formattazione» possa aver un po' disorientato i possessori del 1541 nei primi tempi in cui si sono scontrati (o semplicemente incontrati) con questo nuovo modo di operare.

Lo scopo della formattazione è quello di registrare sul floppy (il dischetto) tutta una serie di riferimenti che successivamente faciliteranno il drive nelle operazioni di lettura e scrittura dati.

È proprio tramite essa che viene predisposta la base per la creazione di un sistema di gestione dati rapido ed efficiente (nei limiti consentiti dalle prestazioni e dal costo della macchina).

Un accesso più veloce alle varie informazioni, maggiori possibilità nella manipolazione dati, eliminazione di errori di sovrascrittura... non sono che alcuni dei vantaggi derivanti dalla formattazione e dalle soluzioni operative ad essa connesse. Dunque, per capire bene come un floppy viene gestito dal DOS (Disk Operating System), è quindi importante comprendere in che cosa essa consista.

Durante l'operazione di formattazione, il DOS suddivide il floppy in 35 zone concentriche, dette tracce, equidistanti fra loro.

Il posizionamento corretto iniziale sulla traccia è effettuato sfruttando le caratteristiche del motore che muove la testina di lettura/scrittura del 1541 (R/W HEAD).

Questo motore (stepper motor o motore passo passo), pilotato dalla routine di Interrupt del DOS, fa avanzare la testina non di moto continuo, ma ad intervalli (o passi) di lunghezza costante il che permette il corretto posizionamento, via software, su ciascuna traccia del disco.

Posizionata la testina sulla traccia, la routine di formattazione comincia a suddividerla nei settori previsti.

Nella figura A possiamo vedere come appare suddivisa fisicamente una traccia, tenendo però presente che questa rappresentazione è solo teorica ed esemplificativa: vediamo perché.

Quando viene posizionato il primo settore su ciascuna traccia, e di conseguenza i successivi, il punto di inizio viene scelto in maniera del tutto casuale.

Alcuni drives fanno riferimento al forellino presente in prossimità del centro del disco, detto foro indice (Index o Timing Hole), per effettuare tutte le operazioni di lettura/scrittura (e quindi anche di formattazione): tramite esso sanno sempre in quale punto della traccia si trovano.

Al contrario di questi ultimi, il 1541 riconosce un settore non dalla sua posizione rispetto all'Index Hole, bensì dalla lettura dei riferimenti registrati su di esso via software. Proprio per questo motivo la tecnica di formattazione adottata da esso è detta Soft-Sectored, mentre l'altra, che utilizza riferimenti fisici fissi, è detta Hard-Sectored.

La velocità di rotazione del disco (velocità angolare) rimane sempre costante, non lo è però la *velocità lineare* con cui viene scorsa ciascuna traccia, che aumenta man mano che la testina si sposta dalle tracce interne verso quelle esterne.

Se mantenessimo costante anche la velocità di trasmissione dati (n.byte per unità di tempo), regolata dal CLOCK del drive, la densità di registrazione (n.byte per pollice), ottimale sulla traccia più interna, andrebbe diminuendo nello spostarsi verso le tracce più esterne, con uno spreco non indifferente di memoria di massa.

Per ovviare a questo inconveniente, la superfice del floppy è stata suddivisa in quattro zone concentriche a ciascuna delle quali corrisponde un diverso tempo di CLOCK, al fine di rendere il più possibile costante la densità di registrazione: la frequenza del CLOCK aumenta con l'aumentare della velocità lineare. Di conseguenza, in corrispondenza di ciascuna di queste quattro zone, si avrà anche una variazione del numero di settori per traccia.

Nella tabella sono riportati il numero di settori presenti su ciascuna traccia e i corrispondenti valori del CLOCK.

Zona disco	Tracce	Numero settori	Frequenza del CLOCK
0	01-17	21	307.692 Kb/s
1	18-24	19	285.714 Kb/s
2	25-30	18	266.667 Kb/s
3	31-35	17	250.000 Kb/s

Grazie a questa soluzione possiamo disporre di quasi 90 settori in più per i

nostri dati, equivalenti ad oltre 22K byte.

Dunque, non vi è alcuna relazione di posizionamento fisico tra settori corrispondenti ma di tracce diverse.

In alcuni drive il problema viene invece risolto mantenendo costante il tempo di CLOCK e variando la velocità di rotazione del disco in funzione della traccia su cui viene posizionata la testina.

Finita questa prima operazione, vengono quindi registrati la BAM, il nome del dischetto, l'Identificatore, la versione del DOS e viene predisposta la Directory (per la quale viene riservata tutta la traccia 18): vedremo successivamente il significato, lo scopo e l'uso di queste informazioni.

Dati caratteristici di un settore

Ciascun settore è costituito da due zone distinte:

> Il Blocco Indirizzi

> Il Blocco Dati

Il primo, usato per identificare univocamente il settore e quindi poter accedere correttamente alle informazioni del Blocco Dati richiesto, viene scritto una volta per tutte durante la formattazione e contiene le seguenti specifiche (fig. B);

— SYNC: carattere di sincronismo, costituito da una particolare successione di «1». Viene utilizzato dal DOS per riconoscere l'inizio di un Blocco Indirizzi o Dati.

— 08: è una costante utilizzata per distinguere il Blocco Dati dal Blocco Indirizzi. È caratteristica di quest'ultimo.

— ID1: è il Primo carattere dell'Identificatore del Disco (ID low).

— ID2: è il secondo carattere dell'ID (ID high).

— Track: contiene il numero della traccia su cui si trova il blocco Dati.

— Sector: è il numero di settore del blocco Dati.

— Checksum: è un numero di controllo ricavato dai valori dei dati che lo precedono. Serve a controllare che le informazioni di un blocco siano registrate correttamente (o che non si siano deteriorate nell'uso del floppy). Ciascun blocco (Indirizzi o Dati) ha il suo checksum.

— GAP 1 o Header Gap: è uno spazio vuoto che separa il Blocco Indirizzi dal Blocco Dati del settore.

Il Blocco Dati viene usato per registrare opportunamente tutte le informazioni che comunichiamo al drive ed è così composto:

— SYNC: idem come sopra.

— 07: costante che identifica il Blocco Dati

— 256 Byte di dati: è la zona del settore riservata alla memorizzazione dei nostri dati ed è l'unica a cui possiamo accedere direttamente tramite i comandi implementati nel DOS. Tutte le altre sono gestite direttamente dal Sistema Operativo a meno di non costruire delle routine apposite (in linguaggio macchina naturalmente).

— Checksum: idem come sopra.

— GAP 2 o Intersector Gap: è uno spazio vuoto presente tra due settori.

Ogni qualvolta il DOS voglia leggere un settore di una certa traccia, per prima cosa posiziona la R/W HEAD su quest'ultima e quindi comincia a leggere tutti i blocchi indirizzi finché non trova quello corrispondente al settore cercato. Se non lo trova, o si accorge che qualche cosa non va (un checksum errato, l'ID diverso da quello previsto, un riferimento mancante...), comunica l'errore riscontrato e lo segnala facendo lampeggiare il LED rosso.

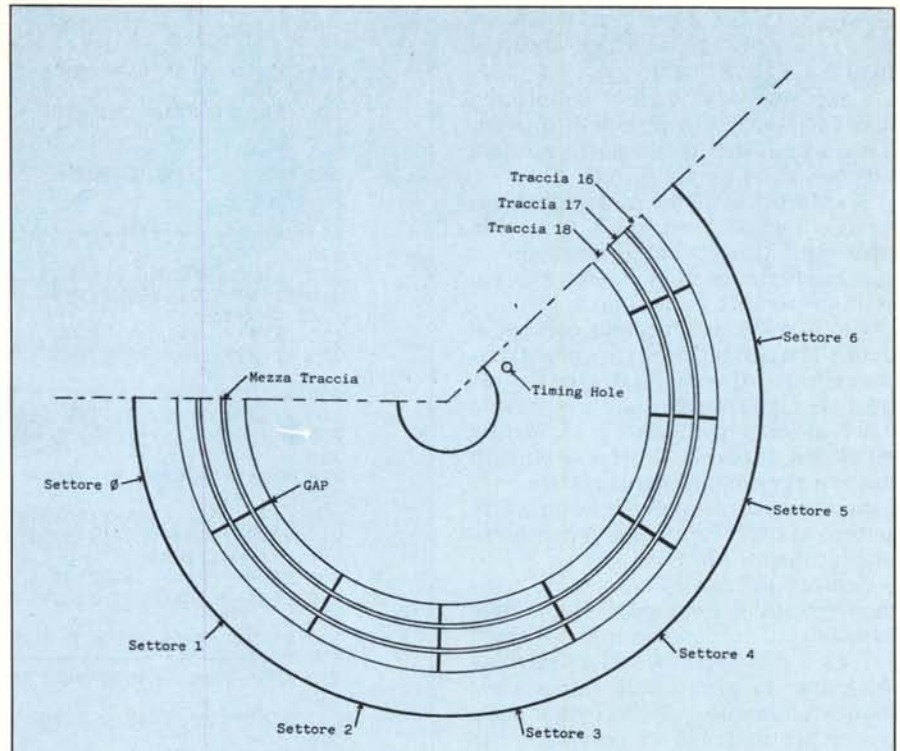


Figura A - Organizzazione teorica delle tracce e dei settori di un disco da 5¼ (disegno non in scala).

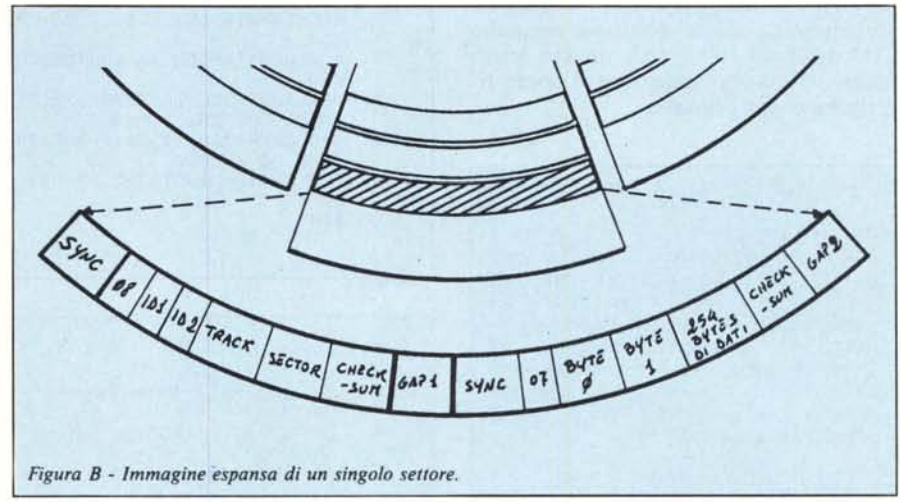


Figura B - Immagine espansa di un singolo settore.

La view BAM

La BAM o Blocks Availability Map (Mappa di Distribuzione dei Blocchi) viene usata per rappresentare via via la situazione di disponibilità di memoria del floppy, quanti e quali settori sono stati utilizzati e quanti e quali sono ancora liberi. In questo modo il DOS sa sempre, nella gestione dei file, quali operazioni è lecito effettuare e

quali no. La BAM è registrata sul settore zero della traccia 18 ed occupa i byte dal 4 al 143 compresi.

Centoquaranta byte complessivi quindi, suddivisi in gruppi di quattro, a ciascuno dei quali corrisponderà una ben determinata traccia.

Il valore ASCII del primo di questi quattro byte dà il numero di blocchi liberi della traccia corrispondente, i successivi tre quali di questi blocchi sono disponibili e quali no.

Ciò è realizzato facendo corrispondere a ciascun bit del byte un ben preciso settore. Al secondo di questi quattro byte corrisponderanno i settori da 0 a 7, al terzo quelli da 8 a 15, mentre all'ultimo quelli da 16 fino al numero massimo consentito per la traccia.

Se il bit corrispondente ad un certo settore è in ON (=1) questo è disponibile altrimenti no (OFF=0).

Sempre nel medesimo settore sono memorizzati il nome del disco (byte 144-160), l'identificatore (byte 162-163) e la versione del DOS (byte 3). E qui un primo fatto importante: modificando quest'ultimo byte il drive non è più in grado di riconoscere le modalità di registrazione adottate su quel floppy ed inibisce tutte le operazioni di scrittura.

La versione del DOS è scritta anche nei byte 165 e 166, ma non viene da esso utilizzata, come non lo è neanche l'ID dei byte 162 e 163: queste informazioni servono unicamente come riferimento per l'utente.

Nota

I codici di controllo nei listati sono riportati in forma «esplicita», in conseguenza dell'impiego della stampante Star NL-10 e relativa interfaccia per Commodore. Ovviamente, nella digitazione del programma è necessario usare i consueti tasti che corrispondono alle indicazioni fra parentesi: ad esempio cursore destro per (RGHT), CTRL-3 per (RED) eccetera.

(CLR)	= ⌐	(YEL)	= ╦
(HOME)	= ≡	(RVS)	= ◘
(DOWN)	= ◘	(OFF)	= ▀
(UP)	= ⌐	(ORNG)	= ⌐
(RGHT)	= ▐	(BRN)	= ▛
(LEFT)	= ▌	(LRED)	= ☒
(BLK)	= ■	(GRY1)	= ╗
(WHT)	= ═	(GRY2)	= ▒
(RED)	= ◙	(LGRN)	= ▐
(CYN)	= ◣	(LBLU)	= ⌐
(PUR)	= ▓	(GRY3)	= ▪
(GRN)	= ≡	(SWLC)	= N
(BLU)	= ▓		

```
100 REM*******************
110 REM*                 *
120 REM* PROGRAM SCRATCH *
130 REM*                 *
140 REM*                 *
150 REM*    PROTECTOR    *
160 REM*                 *
170 REM*                 *
180 REM*******************
190 :
200 CLR:POKE53280,0:POKE53281,0:POKE808,225
210 PRINT"(CLR)(RVS)(CYN)
":Z$=CHR$(0)
215 REM*** (SWLC) = CTRL + N
220 PRINT"(SWLC)(WHT)(RVS)(UP)          I HE •CRATCH ¬ROTECTOR
"
230 PRINTTAB(8)"(UP)────────────────────────────
240 PRINTTAB(18)"BY"TAB(51)"▬▬▬▬▬▬▬▬▬▬▬▬
250 PRINTTAB(11)"(RVS) L⌃I↘ I♣X⌐L♣I⌐ "
260 :
270 REM**** FLASHIN' DELLA SCRITTA ****
280 :
290 C$="(HOME)(LGRN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)(RGHT)↘NSERISC
I IL ¯ISCHETTO "
300 PRINT"(RVS)"C$:GETJ$
310 IFJ$=""THENFORI=1TO200:NEXT:PRINTC$:FORI=1TO200:NEXT:GOTO300
320 :
330 PRINT"(UP)(RVS) <¬> INSERT <¯> ERASE <↑> GO ON <-> END "
340 W$="────────────────────":PRINTTAB(7)"(DOWN)┌"W$"┐"
350 FORI=1TO8:PRINTTAB(7)"I"TAB(31)"I":NEXT:PRINTTAB(7)"└";W$:"┘"
360 :
370 REM****** INIZIO ELABORAZIONE *****
380 :
390 OPEN1,8,15,"I0":OPEN6,8,6,"#":T%=18:S%=1
400 :
410 REM**** LEGGE TRACCIA E SETTORE ***
420 :
430 PRINT#1,"U1:";6;0;T%;S%:GET#6,TT$,SS$::REM*** GET NEXT TRACK
& SECTOR
440 INPUT#1,E1,E1$,E2,E3:IFE1<>0THEN950
450 :
460 REM**LEGGE IL CONTENUTO DEL BLOCCO*
470 :
480 FORI=2TO226STEP32:PRINT#1,"B-P";6;I :REM*** POSIZIONE DEL FIL
E NEL BUFFER
490 GET#6,P$,T$,S$:P=ASC(P$+Z$):REM*** TIPO FILE + TRACCIA E  SET
TORE DI INIZIO
500 ON-(P=0)GOTO790:PR=PAND64:REM*** CONTROLLA IL BIT DI PROTEZIO
NE
510 :
```

```
100 REM********************************
110 REM**                            **
120 REM**    CHANGE   DISK   NAME    **
130 REM**                            **
250 REM********************************
260 :
270 :
280 :
290 :
300 POKE53281,0:POKE53280,0:CLR
310 PRINT"(YEL)(OFF)(CLR)(DOWN)(DOWN)(DOWN)            ┌──────────
──┐
320 PRINT"          I(RVS) CHANGE DISK NAME (OFF)I
330 PRINT"          └─────────────────────┘
340 PRINT"(DOWN)(DOWN)                    BY
350 PRINT"(DOWN)               (RVS)LUIGI TAVOLATO(OFF)
360 PRINT"(DOWN)(DOWN) ┌──────────────────────────────┐
370 PRINT" I <1>  DISPLAY THE  DISK NAME          I
380 PRINT" ├──────────────────────────────────────┤
390 PRINT" I <2>   CHANGE  THE  DISK NAME         I
400 PRINT" └──────────────────────────────────────┘
410 WAIT198,1:GETL$:ONVAL(L$)GOTO630,420:GOTO410
420 PRINT"(DOWN)   ────────────────":A$=""
430 INPUT"  (RVS)>> NUOVO NOME(OFF) :  ";A$:IFA$=""THENRUN
440 A$=LEFT$(A$+"                 ?",16)
```

```
520 REM** GESTIONE PSEUDO-FINESTRA ****
530 :
540 FORF=1TO7:POKE211,9:POKE214,12+F:D$(F)=D$(F+1):SYS58732
550 PRINTD$(F)"(OFF)  ":NEXTF
560 :
570 REM** CANCELLA IL VECCHIO NOME ****
580 REM** E SCRIVE IL NUOVO SUL VIDEO *
590 :
600 D$(F)="  ":IFPR>0THEND$(F)="(RVS)P(OFF) "
610 PRINTTAB(9)"                          ":PRINTTAB(9)"(UP)";D$(F);
620 FORJ=1TO16:GET#6,C$:D$=CHR$(ASC(C$+CHR$(0))):D$(F)=D$(F)+D$:P
RINTD$;
630 ON-(C$="")GOTO910:NEXTJ
640 :
650 REM***** SCELTA DELL' OPZIONE *****
660 :
670 GETJ$:ON-(J$="↑")-(J$="F")*2GOTO790,910:IFJ$<>"P"ANDJ$<>"E"TH
EN670
680 :
690 REM****** SCRIVE NEL BUFFER *******
700 :
710 PRINT#1,"B-P:";6;I
720 IFJ$="P"THENPRINT#6,CHR$(ASC(P$)OR64);:REM*** PROTEGGE IL FIL
E
730 IFJ$="E"THENPRINT#6,CHR$(ASC(P$)AND191);:REM*** ELIMINA LA PR
OTEZIONE
740 D$(F)=D$(F)+" (RVS)"+CHR$(ASC(J$)OR128)
750 :
760 REM** SUCCESSIVO NOME PROGRAMMA ***
770 REM** E RISCRITTURA DEL BLOCCO ****
780 :
790 NEXTI:PRINT#1,"U2:";6;0;T%;S%
800 :
810 REM** SE NON E' L'ULTIMO BLOCCO ***
820 REM** VA A LEGGERE IL SUCCESSIVO **
830 :
840 T%=ASC(TT$+CHR$(0)):S%=ASC(SS$+CHR$(0)):IFS%<>0ANDT%<>0THEN43
0
850 :
860 CLOSE1:CLOSE6:POKE211,17:POKE214,23:SYS58732:PRINT"O.K.(BLK)"
870 PRINT"(UP)(UP)RUN(UP)(UP)(UP)":POKE808,237:END
880 :
890 REM*** SCRIVE L'ULTIMO SETTORE ***
900 :
910 PRINT#1,"U2:";6;0;T%;S%:GOTO860
920 :
930 REM******  STAMPA L'ERRORE  ******
940 :
950 PRINT"(DOWN)(DOWN)(YEL)"E1$"  ( ERR."E1"- TR."E2"- SE."E3")(DO
WN)":STOP:   GOTO860
```

```
450 OPEN1,8,15,"I0":OPEN6,8,6,"#":GOSUB610:ONERGOTO470
460 :
470 REM** IMMETTE NEL BUFFER DEL DRIVE
480 REM** IL BLOCCO DA MODIFICARE E SI
490 REM**   POSIZIONA AL BYTE N'144
500 :
510 PRINT#1,"U1:";6;0;18;0:PRINT#1,"B-P";6;144
520 :
530 REM** COMUNICA IL NUOVO NOME E
540 REM**   RISCRIVE IL BLOCCO
550 :
560 PRINT#6,A$;:PRINT#1,"U2:"6;0;18;0
570 CLOSE6:PRINT#1,"I":CLOSE1:RUN
580 :
590 REM**** CONTROLLO ERRORI DISCO
600 :
610 INPUT#1,X$,W$,Y$,J$:IFX$="00"THENER=0:RETURN
620 PRINT"(DOWN)   (RVS) >> "W$" ("X$"-"Y$"-"J$")":ER=1:RETURN
630 :
640 REM***** NOME DEL DISCO
650 :
660 OPEN3,8,0,"$0":OPEN1,8,15:GOSUB610:ONERGOTO680:PRINT"(DOWN)
  (RVS)  0";
670 FORI=1TO32:GET#3,B$:PRINTCHR$(ASC(B$+" "));:NEXT
680 CLOSE1:CLOSE3:WAIT198,1:RUN
```

La Directory

Prima di spiegare cosa è e come funziona la directory, è opportuno fare una piccola premessa su cosa è un file: esso è un insieme finito di informazioni (testi, programmi, dati anagrafici...), espresse sotto forma di codici binari, ma che noi normalmente trattiamo in forma decimale.

A ciascun file, nel momento in cui viene registrato sul floppy, viene assegnato dall'operatore un nome, lungo al massimo 16 caratteri, diverso da tutti quelli dei file già presenti sul dischetto, e che lo identificherà univocamente.

La directory contiene tutte le informazioni necessarie per identificare, richiamare o visualizzare tutti i file presenti sul disco.

Queste informazioni, registrate sulla traccia 18, a cominciare dal settore 1, sono organizzate in gruppi di 32 byte:

byte 0: tipo file. I primi tre bit di questo byte indicano il tipo di file registrato, il bit 7 in ON (=1) indica che è stato registrato correttamente. Il bit 6 settato (1) indica che non è possibile cancellarlo con un normale comando di Scratch. I bit 3-5 non sono significativi.

byte 1-2: traccia e settore di inizio del file.

byte 3-18: nome del file. I byte non usati vengono riempiti di spazi shiftati (CHR$(160)).

byte 19-20: traccia e settore del primo SideSector se il file è relativo.

byte 21: lunghezza del record (file relative).

byte 22-25: non usati. L'utente può eventualmente riservarli per contenere dati a lui utili senza alterare minimamente le funzioni del DOS.

byte 26-27: traccia e settore del nuovo file quando viene effettuato il salvataggio con rimpiazzamento del file (@: Save with Replace).

byte 28-29: numero dei blocchi usati dal file espresso nella forma Byte Basso-Byte Alto.

byte 30-31: non usati.

La corrispondenza tra tipo file e il valore (ASCII o binario) che lo rappresenta è:

Tipo file	Sigla	ASCII	Binario
Deleted	DEL	0	00000000
Sequential	SEQ	1	**000001
Program	PRG	2	**000010
User	USR	3	**000011
Relative	REL	4	**000100

Dunque, se la registrazione del file è stata completata correttamente, viene effettuato un OR tra il valore corrispondente al tipo file e $80 (128 -> bit7), quindi scritto il risultato nella posizione prestabilita. Volen-

dolo protetto da scratch, l'OR viene effettuato anche con $40 (64 -> bit 6) o, complessivamente, con $CO (192 = 64 + 128 -> bit 6 e 7).

Tramite un AND tra il valore ASCII del byte zero e $07 si riottiene il tipo file originale, in quanto viene visualizzato il valore corrispondente ai soli primi 3 bit.

Le modalità operative di lettura/scrittura della directory rispettano esattamente quelle adottate per i File Relativi, le cui caratteristiche vedremo nella parte dedicata alla gestione dei file.

I File

È opportuno fare una distinzione tra File programmi e File Dati.

Tale distinzione è però più concettuale (o convenzionale) che reale, poiché File programmi possono essere letti e trattati come File Dati e viceversa.

I primi, come già il nome suggerisce, contengono programmi (in BASIC o in Linguaggio Macchina o...), mentre gli altri contengono informazioni di vario tipo, legate alle necessità dell'utente (nominativi, indirizzi, importi, titoli di borsa...) e possono essere gestiti unicamente da programma. Queste informazioni sono registrate sotto forma di RECORD.

Un Record è una successione limitata di caratteri alfanumerici (descrizioni, sigle, cifre...). In un file i Record sono registrati uno di seguito all'altro: nel caso in cui il numero di caratteri che ciascuno occupa non sia sempre il medesimo, viene interposto, per distinguere la fine di uno e l'inizio del successivo, uno speciale carattere detto Carriage Return (letteralmente, Ritorno di Carrello = CHR$(13)).

A seconda del tipo di file su cui operiamo (programmi o dati) questi codici assumeranno significati diversi e diversi saranno i modi di accedervi.

Nel DOS del 1541 e nel Sistema Operativo del C-64 vi sono già implementate le routine per l'uso di File Programmi mentre, quelle per la gestione di File Dati (eccezion fatta per quelle elementari di creazione e scansione), devono essere costruite dagli utenti in funzione delle loro necessità.

Queste necessità condizionano anche nella scelta di che tipo di File Dati gestire (Sequenziale, Relativo, User) poiché, a seconda del tipo di file usato, variano le modalità di accesso ai dati.

Un file programma è anche detto BINARIO poiché contiene esattamente il valore di ciascuna locazione di memoria compresa nell'area prevista.

Un file Dati è anche detto ASCII poiché viene scritto come una sequenza di caratteri il cui valore numerico corrispondente rispetta la codifica ASCII: può essere gestito unicamente da programma a differenza del precedente che, grazie alle suddette routine, può esserlo anche in maniera diretta.

I File Dati

Ogni tipo di File Dati è caratterizzato da una diversa modalità di accesso alle informazioni.

La scelta di un tipo piuttosto che un'altro è fatta seguendo prevalementemente criteri di ottimizzazione nell'occupazione di memoria di massa (cioè del floppy), e nei tempi di ricerca ed accesso delle informazione. Una trattazione dettagliata di tali aspetti esula però dallo scopo di questo articolo.

Brevemente possiamo riassumere le caratteristiche dei file:

— SEQUENZIALI (SEQ): le informazioni sono raggruppate in Record.

```
100 REM**********************
110 REM*   VIEW  B.A.M.     *
120 REM*                    *
130 REM*         &          *
140 REM*                    *
150 REM*  DIS/ALLOCAZIONE   *
160 REM*                    *
170 REM*      BLOCCHI       *
180 REM**********************
190 :
200 :
210 POKE53280,0:POKE53281,0:CLR:Z$=CHR$(0)
220 :
230 REM*** FUNZIONI DI UTILITA' ***
240 :
250 DEFFNT(T)=20+(T>17)*2+(T>24)+(T>30):REM** N.MAX SETTORI PER T
RACCIA
260 :
270 DEFFNS(S)=2↑(S-INT(S/8)*8)AND(B(INT(S/8))):REM*** DISPON. SET
TORE
280 :
290 PRINT"(CLR) (RVS)(SWLC)  !↑\ & ¯IS-↑LLOCAZIONE DEI BLOCCHI  "
300 PRINT"                  (DOWN)DI":PRINT"           (DOWN)LUIGI
 !AVOLATO
310 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)     (RVS)↑(OFF)LLOCAZIONE DI U
N BLOCCO
320 PRINT"(DOWN)      (RVS)¯(OFF)ISALLOCAZIONE DI UN BLOCCO
330 PRINT"(DOWN)      (RVS)×(OFF)IEW !AM
340 PRINT"(DOWN)      (RVS)●(OFF)TATUS
350 GETL$:ON-(L$<>"A"ANDL$<>"D"ANDL$<>"V"ANDL$<>"S")GOTO350
360 ON-(L$="V")-(L$="S")*2GOTO670,1140
370 PRINT"(DOWN)(DOWN)  !RACCIA : ";:INPUTTR:IFTR<1ORTR>35THENRUN
380 PRINT"(DOWN)  ●ETTORE : ";:INPUTSE:IFSE>FNT(TR)THENRUN
390 :
400 :
410 REM*******************************
420 REM** DIS-ALLOCAZIONE NELLA BAM **
430 REM*******************************
440 :
450 OPEN1,8,15
460 OPEN2,8,2,"#1"
470 PRINT#1,"U1:"2;0;18;0
480 INPUT#1,E$,M$,T$,S$
490 IFE$<>"00"THENPRINT"(DOWN)(RVS) "M$" (OFF) ("E$"-"T$"-"S$")":
GOTO1100
500 :
510 PRINT#1,"B-P:"2;TR*4
520 GET#2,J$,S$(0),S$(1),S$(2):REM *  LEGGE DISPONIBILITA' TRACCI
A
530 :
540 J$=CHR$(ASC(J$+Z$)+1+(L$="A")*2):REM * INCREMENTA O DECREMENT
A N.BLOCCHI
550 BY=INT(SE/8):REM * CALCOLA QUALE DEI 3 BYTE CONTIENE IL BIT D
EL SETTORE
560 CH=ASC(S$(BY)+Z$):REM * VALORE NUMERICO DEL BYTE TROVATO
570 BIT=2↑(SE-BY*8):REM   * ISOLA IL BIT DEL SETTORE RICHIESTO
580 IFL$="D"THENS$(BY)=CHR$(CHORBIT):GOTO610:REM * DISALLOCA IL S
ETTORE
590 S$(BY)=CHR$(CHAND(255-BIT)):REM * ALLOCA IL SETTORE
600 :
610 PRINT#1,"B-P:"2;TR*4
620 PRINT#2,J$;S$(0);S$(1);S$(2);:REM * AGGIORNA LA SITUAZIONE NE
LLA BAM
```

Questi sono registrati uno di seguito all'altro, divisi dal carriage return e possono essere di lunghezza variabile.

Per accedere ad un qualsiasi record è necessario leggere tutti quelli che lo precedono.

— UTENTE (USR): il loro criterio di uso è il medesimo dei file sequenziali. L'appellativo di USR può servire, ad esempio, per evidenziare un certo file rispetto ad altri presenti sul medesimo dischetto.

— RELATIVI (REL): a differenza del precedente tipo di file, qui i Record usati sono tutti di lunghezza fissa (quindi non è necessario usare carriage return). Ciò rende possibile calcolare esattamente la posizione di ciascun record nel file, velocizzando notevolmente tutte le operazioni di ricerca.

L'handicap è che, se ad esempio abbiamo stabilito di usare record lunghi 30 caratteri e l'informazione da registrare ne occupa solo 10, vi sono un certo numero di caratteri superflui che devono comunque essere utilizzati da quel record, occupando spazio di memoria a vuoto.

N.B.: solo il BASIC 4.0 è dotato di comandi per facilitare la gestione di questo tipo di file. Usando versioni precedenti del BASIC, bisogna costruire le opportune routine.

— RANDOM (o ad accesso diretto): questo è un tipo di file un po' diverso da quelli visti fino ad ora poiché, mentre per i precedenti file l'uso dei blocchi dati sul floppy veniva gestito dal DOS, qui invece è direttamente l'utente che stabilisce tutte le modalità operative.

Non si scrivono più i dati nel file ma direttamente sul disco specificando in quale Traccia e Settore si vogliono allocare. Non sarà più il DOS ad aggiornare la BAM automaticamente, ma dovrà essere l'utente a comunicargli le eventuali variazioni.

Ciò viene realizzato usando i comandi BLOCK, il cui uso vedremo in seguito. I file Random vengono anche chiamati impropriamente Casuali.

```
630 PRINT#1,"U2:"2;0;18;0:REM          * RISCRIVE LA NUOVA BAM
640 CLOSE1:CLOSE2
650 RUN
660 :
670 :
680 REM*********************************
690 REM** VISUALIZZAZIONE DELLA BAM **
700 REM*********************************
710 :
720 Y$="(HOME)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)"
730 PRINTCHR$(142)"(CLR)(DOWN) ▬"
740 :
750 REM*** DISEGNA  ASSI ***
760 :
770 PRINT"▮(RVS) T(OFF)";:FORI=1TO35:PRINTRIGHT$(STR$(I),1);:NEXT
:PRINT
780 PRINT"▮(RVS)S (OFF)";:FORI=1TO35:PRINT"▬";:NEXT:PRINT
790 FORI=0TO20:PRINT;LEFT$(Y$,5+I);RIGHT$(STR$(I),2);"▮";:NEXT
800 :
810 OPEN1,8,15,"I0"
820 OPEN2,8,2,"#1"
830 PRINT#1,"U1:";2;0;18;0
840 :
850 REM*** LEGGE NOME E ID DEL DISCHETTO
860 :
870 PRINT#1,"B-P:"2,144:REM*** POSIZIONE DEL NOME NEL BUFFER
880 PRINT"(HOME)";TAB(7);
890 FORI=1TO23:GET#2,A$:PRINTCHR$(ASC(A$+" "));:NEXT:PRINT
900 :
910 PRINT#1,"B-P:"2;4:REM*** POSIZIONE INIZIALE DEI BYTES DELLA B
AM
920 :
930 FORT=1TO35
940 GET#2,BL$,B0$,B1$,B2$:REM   *** DATI CARATTERISTICI DELLA TRA
CCIA
950 BL=BL-ASC(BL$+Z$)*(T<>18):REM*** BLOCCHI LIBERI
960 B(0)=ASC(B0$+Z$):REM*** DISPONIBILITA' SETTORI 0 - 7
970 B(1)=ASC(B1$+Z$):REM*** DISPONIBILITA' SETTORI 8 - 15
980 B(2)=ASC(B2$+Z$):REM*** DISPONIBILITA' SETTORI DA 16 A FNT(T)
990 :
1000 REM*** VISUALIZZA LA DISPONIBILITA' DEI SETTORI DELLA TRACCI
A
1010 :
1020 FORS=0TOFNT(T)
1030 PRINTLEFT$(Y$,5+S);TAB(2+T);
1040 IFFNS(S)<>0THENPRINT"▟";:GOTO1060:REM*** SETTORE LIBERO
1050 PRINT"(RVS)▟(OFF)";:REM*** SETTORE OCCUPATO
1060 NEXTS
1070 :
1080 NEXTT
1090 PRINTLEFT$(Y$,25);TAB(23);RIGHT$(" "+STR$(BL),3);" BLOCKS FR
EE";
1100 CLOSE1:CLOSE2:POKE198,0:WAIT198,1:RUN
1110 :
1120 REM*** DISK STATUS ***
1130 :
1140 OPEN1,8,15
1150 INPUT#1,E$,M$,T$,S$
1160 PRINT"(DOWN)(DOWN)(DOWN)(RVS) ";M$;" (OFF) (";E$;"-";T$;"-"S
$")"
1170 CLOSE1:POKE198,0:WAIT198,1:RUN
```

Organizzazione dei file su floppy

Ciascun file, Random eccettuati, è registrato in blocchi di 254 caratteri, pur disponendone di 256, poiché i primi due caratteri di ciascun blocco dati indicano quale è la Traccia ed il Settore dove andare a leggere il blocco successivo.

Nell'ultimo blocco del file il primo di questi caratteri contiene il valore ASCII zero mentre il secondo il numero di caratteri significativi del blocco dati. Questo perché, quando viene cancellato un file, vengono riutilizzati, (ma non cancellati) i blocchi che esso occupava. Pertanto è probabile che l'ultimo blocco del nuovo file, sovrascrivendosi parzialmente al vecchio, possa contenere dei caratteri spurii che devono essere ignorati: da qui la necessità di un segnale che indichi quanti byte di quell'ultimo blocco facciano effettivamente parte del file.

Formato di un Blocco Dati di un File:

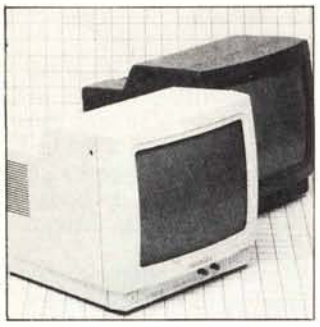
Byte	Contenuto
0	N. Traccia del successivo Blocco Dati (oppure 0)
1	N. Settore del successivo Blocco Dati (o n. byte significativi del blocco)
2-555	Dati del file

I settori non utilizzati hanno normalmente il primo byte contenente il valore ASCII 75 (K) e tutti i successivi contenenti 1 binari (00000001).
Appuntamento il prossimo mese.

MC

MASTERBIT
MIPECO

VENDITA PER CORRISPONDENZA



MONITOR HANTAREX BOXER 12"

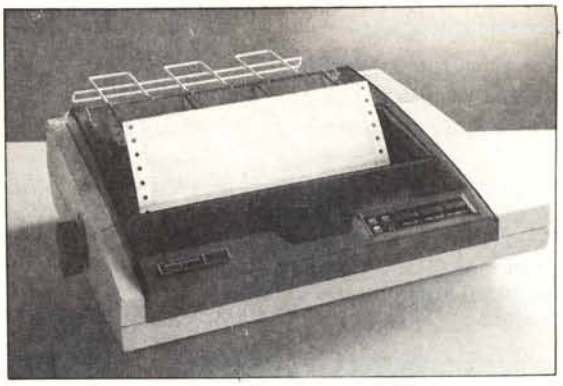
FOSFORI VERDI - ALTA RISOL.
85 COL. - AUDIO - R G B PER QL
COMMODORE - MSX
APPLE II e II+

LIRE 230.000 (tutto compreso)

MONITOR HANTAREX CT 900/1 SR PAL

14" COLORE - MEDIA RISOL.
40 col. - R G B - LINEAR/TTL
COMPAT. PER QL
COMMODORE

LIRE 540.000 (tutto compreso)



MANNESMANN MT 80+

L. 599.000

80 col. - 100 cps - interfaccia Centronics - foglio singolo e modulo continuo - bidirezionale.

QL SINCLAIR 128K 549.000

Tutto compreso 6 mesi di garanzia

CPU MOROTOLA 68008 da 32 BIT e 2 microdrive. Ultima versione con nuovi programmi, alimentatore, manuale in inglese, manuale in italiano, 4 cartucce con i 4 programmi gestionali + 1 cartuccia con giochi originali (PIRATE, ZETA, PED, GUN, SREAKOUT, HUNT) e in regalo un ottimo copiatore per mdv e floppy di Massimo Rossi

SPECTRUM 48K PLUS

299.000 Tutto compreso 6 mesi di garanzia

PC IBM Compat. Varie versioni
Tel. per quotazioni - Prezzi imbattibili

con lo SPECTRUM plus **manuale in italiano** *e in regalo 5 programmi in italiano (conto corrente, grafica, funzioni, bioritmi, esapedone + il Supercopiatore di Massimo Rossi)*

QL 512K 880.000
Espansione da 512K montata internamente, non necessita di alimentazione supplementare e lascia il connettore libero per altre periferiche.

Nuovo SPECTRUM 48K+ 299.000
Manuale in italiano, cavetti alimentatore, cassette dimostrative e oltre 50.000 lire di software originale e in italiano.

Personal AMSTRAD PC W8 256 ..... 1.350.000
256K - 1 Drive 3" - Monitor - Stampante NLQ - 90 cps

PC W8 - 512 ..... 1.599.000
512 - 2 Drive 3" - Monitor - Stampante NLQ - 90 cps

10 RULLI di carta termica 29.000

MANNESMANN TALLY tutti i modelli

MT 80+ 599.000
Foglio singolo e continuo, interfaccia Centronics, 100 cps vari set di caratteri - Biodirezionali.

MT85 ..... 899.000
Interfaccia Centronics o seriale a scelta 180 cps 80/136 col. foglio singolo e continuo.

DISCHI 3"1/2 ..... 13.000

DISCHI 3"1/2 10 pezzi 110.000
Garantiti doppia faccia e doppia densità.

INTERFACCIA PER JOYSTICK UNA PRESA 25.000
Tipo Kempston, per tutti i joystick stand. 9 PIN D.

INTERFACCIA PER JOYSTICK DUE PRESE 35.000
Tipo Kempston, per tutti i joystick stand. P PIN D.

JOYSTICK STANDARD 9 PIN D 14.000

CONVERTITORE 99.000
Da RS232 a Centronics per interfaccia 1 o per QL **cavi e connettori speciali compresi.**

INTERFACCIA CENTRONICS SPECTRUM 99.000
Senza software tutto su Rom compreso il copy.

8 CARTUCCE x MICRODRIVE 49.000

TRISLOT 27.000
Presa tripla per connettore Spectrum.

MANUALE IN ITALIANO x SPECTRUM .. 16.000
«Come usare il tuo Spectrum».

ROM «JS» NUOVO TIPO (256K + 128K) . 99.000
Trasforma il tuo QL in un «JS».

MODEM: TUTTI I TIPI dal più economico al più sofisticato.

TUTTI gli articoli EPSON telefonare per quotazioni aggiornate.

INTERFACCIA PARLANTE CURRAH 75.000
Manuale completo in italiano.

ESPANSIONE x 32K x SPECTRUM 59.000
Issue 2 o 3 specificare, facilissima da montare, istruzioni dettagliate in italiano con fotografie, porta il VS Spectrum da 16K a 48K. Montaggio gratis.

STAMPANTE ALPHACOM 32 ..... 149.000
Per Spectrum ZX 81 istruzioni in italiano 2 rulli di carta in regalo.

DISK DRIVE 3"1/2 x INTERF. x QL 619.000
Oltre 700K formattati.

Espansione QL da 512 K con totale 640 K disponibili, montate inter 300.000

KIT DI ESPANSIONE x QL a 512 ..... 249.000
Si monta all'interno del QL, si consiglia l'assistenza di un tecnico specializzato.

ESPANSIONE DEL VOSTRO QL A 512K 349.000
Montata all'interno del vostro QL e collaudata con garanzia di 3 mesi spedite il Computer solo dopo aver avuto un contatto telefonico.

TOOLKIT II x QL SU ROM 89.000

STAMPANTE WELCO DMP - 1100 per QL 650.000
100 cps, foglio singolo e continuo, 80 col. bidirezionale, 192 car interf. RS 232 incorporata

MASTER BIT Viale dei Romagnoli 35
MIPECO 00121 OSTIA LIDO RM
CAS. POST. 3016

AVVERTENZE - Tutti i prezzi sono comprensivi di IVA e spese postali per ordini inferiori alle 50.000 lire aggiungere L. 5.000 per contributo spese di spedizione — pagamento contrassegno al ricevimento del pacco è gratuito il contatto telefonico — **sconti quantità.**
Listino prezzi aggiornato anche su richiesta telefonica.

PARTI DI RICAMBIO PER SPECTRUM E QL

GARANZIA 48H: oltre la normale Garanzia di 6 mesi per i Computer e di 3 mesi per gli accessori, la MASTERBIT MIPECO si impegna a sostituire tutto il materiale trovato malfunzionante, entro 48 ore dal ricevimento.

ORDINI TELEFONICI (ore 8.30/20.30): 06/5611251