

i trucchi del CP/M

a cura di Pierluigi Panunzi

Le funzioni del BDOS

Nell'analisi delle funzioni del BDOS, che continueremo in questa e nelle prossime puntate, abbiamo finora incontrato parecchie caratteristiche comuni ad altri Sistemi Operativi, ma che nel CP/M, a nostro avviso, trovano una semplicità di applicazione veramente notevole.

Ricordiamo infatti che per attivare una certa funzione del BDOS dobbiamo innanzitutto porre il suo numero d'ordine nel registro C, poi dobbiamo passare gli eventuali parametri nel registro specificato (in genere il registro E, la coppia DE o HL nonché l'accumulatore) ed infine dobbiamo effettuare la chiamata alla subroutine posta all'indirizzo 0005 (con una CALL 0005), subroutine che in realtà è un semplice salto (JP) all'«entry point» del BDOS stesso.

A seconda della funzione chiamata e dipendentemente da eventi che accadono durante l'esecuzione della routine stessa (lettura da disco, input da console, ecc), si avranno uno o più valori di ritorno posti in ben determinati registri: a seconda dei valori ottenuti si potranno prendere le opportune decisioni. Ad esempio dopo aver effettuato un'operazione che coinvolge l'unità a dischi si avrà in genere nell'accumulatore un valore indicante l'esito dell'operazione stessa.

Proseguiamo dunque l'analisi del BDOS dalla funzione 22.

Funzione 22: Create File

Ecco un'altra funzione tipica della gestione dei file su disco e presente anche nei linguaggi ad alto livello ed in altri Sistemi Operativi.

In questo caso il valore da porre nel registro C è ovviamente 22, mentre la coppia DE deve puntare all'FCB (File Control Block) del file in questione, secondo quella che è la prassi comune per le funzioni relative ai file. Non ripeteremo in questa sede le caratteristiche dell'FCB, in quanto già l'abbiamo fatto in più riprese nelle scorse puntate.

Come è più che ovvio, la funzione in esame serve per creare un file (che perciò deve essere *nuovo* e cioè non già esistente): ci si deve infatti assicurare che il file già non esista o cercando di aprire un file avente tale nome (e

perciò aspettandosi una risposta negativa = file non esistente) oppure andando a cancellare l'eventuale file in questione senza pensarci troppo su. Supponendo perciò che non esista già il file nel disco, la funzione 22 crea il file e la sua directory entry ed inoltre effettua l'operazione di «apertura» del file in questione, sul quale ci si potrà scrivere subito.

Come risultato della chiamata, la funzione «Create File» fornisce un valore nell'accumulatore: un «directory code» (0,1,2 o 3) indica che la funzione è stata completata con successo, mentre un valore pari ad FFH indica che non vi è spazio sufficiente nel disco oppure nella directory e *non* che il file già esisteva.

Terminiamo perciò l'analisi di questa funzione dicendo che non si possono usare nomi «ambigui» nell'FCB (e cioè non si può usare il carattere «?») e raccomandando l'uso della «Search File» congiuntamente alla «Erase File» nel caso in cui non si è certi che il file «non esista già»

Funzione 23: Rename File

Ecco dunque all'appello l'ultima funzione tipica di un DOS e cioè quella che permette di rinominare un file: in questo caso, unico peraltro in tutto il BDOS, il nome del file «vecchio» ed il nome di quello «nuovo» devono stare all'interno di un unico FCB, al cui indirizzo punterà come al solito la coppia DE.

In particolare l'«old file» troverà posto nei primi byte (8 per il nome e 3 per il tipo, senza dimenticare il disco, come è prassi corrente), viceversa il «new file» troverà posto nei successivi 16 byte dei 32 a disposizione.

In parole povere il nome del file «di arrivo» dovrà trovare posto a partire da FCB+16:

FCB + 16	disco
FCB + 17	} nome del file nuovo
...	
FCB + 24	
FCB + 25	} tipo del file nuovo
...	
FCB + 27	

Come risultato si otterrà un ormai usuale valore in A: 0,1,2 o 3 se tutto ha

funzionato a dovere, FFH se non è stato trovato il file «vecchio».

Questo fatto è ancora una volta importante: la funzione in esame testa soltanto se il file «vecchio» esiste, mentre non si assicura assolutamente che eventualmente il «nuovo file» possa già esistere. Ciò può comportare gravi malfunzionamenti se non si prendono le dovute precauzioni: in particolare si dovrà innanzitutto tentare l'apertura del nuovo file e solo in caso negativo effettuare la funzione di «rename». In caso positivo invece il CP/M (orrore!!) creerebbe due file aventi lo stesso nome e tipo con le conseguenze del caso...

Ecco che involontariamente abbiamo scovato un altro punto debole di questo sistema Operativo: programmatori sadici potranno infierire creando così decine di file identici dai quali però non si caverà il classico ragno dal buco...

Altra pecca della funzione è che *non* si possono usare nomi ambigui (i soliti nomi costellati di «?») pena disgraziatissimi effetti irreparabili su file magari incolpevoli che rimarrebbero inevitabilmente danneggiati: ecco ancora un'ottima occasione per i programmatori-distruttori, i quali potrebbero devastare una directory di un altrettanto incolpevole ed ignaro utente.

Come fattore positivo (e cioè nel caso in cui si opera correttamente), la funzione «Rename» rinominerà tutte le occorrenze del file «vecchio» ed in particolare (come è ovvio) tutti i suoi eventuali «extent».

Funzione 24: Get Active Disks

Ecco una funzione particolarissima, che consente di conoscere quale delle varie unità a dischi a disposizione del sistema è attiva (e cioè, come si dice in gergo tecnico, «on line») dall'ultimo «warm boot» (a seguito di un Control-C) o dall'ultima chiamata alla funzione 13 del BDOS, la «Reset Disk System».

Con il termine «attivo» o «on line» intendiamo in questo caso le unità logiche che sono state selezionate a partire da uno dei due eventi menzionati poco sopra.

In questo caso la funzione non richiede altri parametri in input se non il solito registro C, che deve ovviamente contenere il valore 24.

Viceversa come output si ottiene il cosiddetto «login vector» nella coppia HL.

In particolare i 16 bit di tale coppia rappresentano ognuno una unità a dischi, a partire dall'unità «A:» al quale corrisponde il bit meno significativo (il bit 0 di L), fino ad arrivare alla massima unità a dischi permessa dal CP/M e cioè la «P:» alla quale corri-

sponde il bit più significativo della coppia (il bit 7 di H). Scendendo più in dettaglio, se un'unità a dischi è stata selezionata allora il corrispondente bit della coppia HL lo troveremo settato e viceversa un bit posto a 0 indicherà che la corrispondente unità a dischi non è stata mai selezionata.

Nel caso più comune di un personal computer dotato di due dischetti, rispettivamente «A:» e «B:», chiamando la funzione in esame potremo trovare ad I solo i due bit meno significativi, relativi appunto ad A: ed a B:.

Funzione 25: Get Current Default Disk

È questa un'altra funzione particolare che si utilizzerà solo in determinati programmi applicativi: essa fornisce in accumulatore un valore, compreso tra 0 e 0FH, indicante appunto qual è l'unità a dischi correntemente attiva, in uso.

Un valore 0 indicherà come tale l'unità a dischi «A:», un 1 indicherà l'unità «B:», ecc., fino al valore 0FH che indicherà come attiva l'unità «P:».

Questi valori tra l'altro sono molto utili nel caso in cui vogliamo derivarne la codifica ASCII, magari per stamparla sul video: è banale osservare che basta sommare al valore ottenuto dalla funzione il valore esadecimale 41H.

Funzione 26: Set DMA Address

Questa funzione è importantissima in quanto permette di stabilire l'indirizzo di memoria a partire dal quale verrà memorizzato un record letto da disco oppure a partire dal quale leggere i dati da porre su disco con un'operazione di Write Disk: come si vede è di particolare importanza nella gestione di file, in genere più lunghi di 128 byte (ricordiamo, la lunghezza di un record logico per il CP/M), ad esempio per caricare un certo file in memoria a partire da un certo indirizzo stabilito dal nostro programma.

Il nome strano di questa funzione deriva dal fatto che originariamente il CP/M era nato su di un sistema di sviluppo dell'Intel, nel quale l'interscambio di dati da e verso l'unità a dischi avveniva appunto tramite DMA (Direct Memory Access) e non perciò tramite I/O guidato dalla CPU.

Ora che invece si utilizzano controller intelligenti, il DMA viene usato raramente ed ecco perché non avrebbe senso mantenere il nome arcaico di questa funzione: semmai si dovrebbe cambiare in «Set Disk I/O Address» o qualcosa di simile.

In particolare, come parametri di ingresso, oltre all'ormai ovvio registro C, abbiamo la coppia DE, che deve

contenere appunto l'indirizzo desiderato.

C'è da dire che il CP/M, a seguito di un «warm boot» (scatenato con un Control-C) oppure a seguito della chiamata alla funzione 13 di Reset Disk, pone il DMA Address all'indirizzo 0080H.

Aggiungiamo inoltre che non esiste alcuna funzione che permette di sapere qual è il valore corrente del DMA Address: per essere sicuri che sia posto laddove desideriamo noi, allora basta effettuare la chiamata alla funzione in esame, forzando appunto il valore voluto.

Funzione 27: Get Allocation Vector

Prima di vedere il risultato di questa funzione, consigliamo i lettori di fare riferimento per i dettagli tecnici, alla rivista n. 42, sempre nell'ambito della rubrica «I trucchi del CP/M», nella quale parliamo degli «Allocation Blocks».

In questa sede ricordiamo solo che, per snellire l'altrimenti pesante compito di gestire file sulla base di record di 128 byte, i progettisti del CP/M hanno ideato appunto gli Allocation Blocks, semplicemente un raggruppamento di un certo numero di record, che in tal modo rimarranno per sempre vincolati.

In particolare tutta la gestione dei file avviene su base Allocation Blocks per cui un certo file sarà ad esempio formato da 10 unità logiche: quando andremo a cancellare tale file, allora i dieci Allocation Blocks precedentemente usati dal file ritorneranno a disposizione del Sistema Operativo.

Per tener conto appunto di quali Allocation Blocks sono stati effettivamente usati e quanti invece sono disponibili, il CP/M tiene costantemente aggiornato il cosiddetto «Allocation Vector», il cui indirizzo si ottiene nella coppia HL per effetto della chiamata alla funzione in esame.

Sulle caratteristiche di tale «Vector» non possiamo essere precisi in quanto nel testo, dal quale traiamo lo spunto per questi articoli (vedi finestra), l'autore riporta la seguente frase: «Digital Research considers the actual layout of the allocation vector to be proprietary information», come dire che i progettisti della «premiata» non si vogliono sbottonare più di tanto... Ai lettori l'arduo compito di analizzare l'«Allocation Vector» e di sviscerarne i segreti.

Terminiamo questa puntata analizzando un'altra funzione del BDOS.

Funzione 28: Disk Write Protect

Questa funzione, come è facile arguire dal suo nome, permette di cam-

biare lo stato del disco corrente, da R/W a R/O, e cioè da «Read/Write» a «Read/Only» ed ancora meglio da disco a cui si può accedere sia in lettura che in scrittura a disco dal quale si può solamente leggere, indipendentemente dallo stato dei singoli file che lo compongono.

In particolare a seguito della chiamata di tale funzione, i successivi tentativi di eseguire una «Write Sequential» o «Write Random» saranno vanificati dal CP/M, il quale bloccherà l'esecuzione del programma e mostrerà sul video l'irritante messaggio

BDOS ERROR ON d: R/O

dove abbiamo lasciato indicato con «d:» un generico disco: nel nostro caso al posto della «D:» troveremo proprio l'indicazione relativa al disco progettato dalla scrittura.

C'è da dire che una volta imposto lo stato di «Read Only» al disco corrente, tale stato rimarrà anche per i successivi dischi selezionati, fino a che non si avrà un «warm boot» oppure una chiamata alla funzione «Reset Disk System», analogamente ai casi precedenti.

Quasi-bibliografia

Alcuni lettori ci hanno chiesto la fonte delle informazioni che riportiamo nelle varie puntate di questa rubrica: si tratta di un interessantissimo volumetto di 500 pagine, una vera miniera di informazioni preziosissime, condite con esempi applicativi in Assembler 8080.

Riteniamo che tale testo non sia alla portata del programmatore alle prime armi, anzi è richiesta un'ottima conoscenza della programmazione a livello assemblativo: il tutto però è scritto (in inglese) in maniera molto chiara anche laddove i concetti espressi sono difficilmente digeribili se non dopo svariati tentativi.

A chi interessa diciamo che si tratta di: «The Programmer's CP/M Handbook» dell'infaticabile Andy Johnson-Laird, edito dalla casa editrice Osborne/McGraw-Hill.

Comunque molte delle informazioni sono a loro volta tratte dalle pubblicazioni originarie della Digital Research, quali:

- CP/M 2.0 User's Guide
- CP/M Assembler (ASM)
- CP/M Dynamic Debugging Tool (DDT)
- CP/M Alteration Guide
- CP/M Interface Guide

oggettivamente più difficili da reperire da parte del programmatore singolo (per le «Software House» dovrebbe essere più semplice) che non il libro, acquistabile presso le librerie specializzate in testi scientifici.

DATAFLEX

PROFESSIONAL

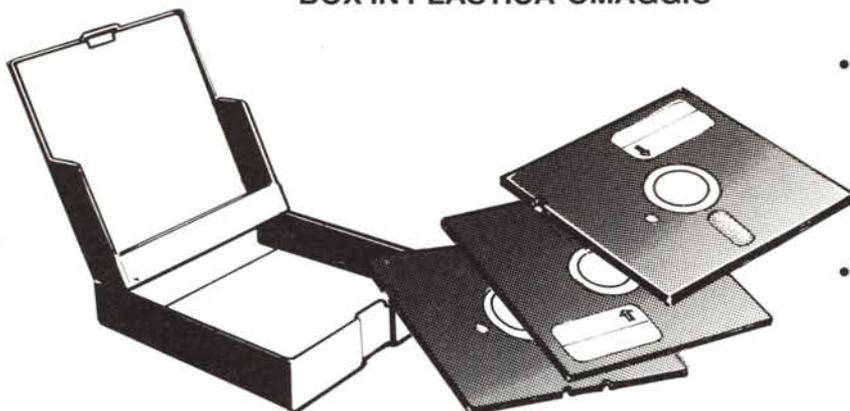
5 1/4

PREZZI
IVA
ESCLUSA

CASELLA POSTALE 142
56025 PONTEDERA (PI)
VIA MISERICORDIA, 84
TEL. 0587 - 212.312



BOX IN PLASTICA OMAGGIO



TRATTAMENTO
MULTICOT



- I DISCHETTI DATAFLEX SONO PRODOTTI DA UNO DEI PIÙ GROSSI FABBRICANTI AMERICANI CHE GARANTISCE L'ALTISSIMA QUALITÀ ED AFFIDABILITÀ.
- UNO SPECIALE ED ESCLUSIVO STRATO «MULTICOT» PROTEGGE LA SUPERFICIE DALL'USURA DEL CONTATTO CON LE TESTINE GARANTENDO MINIMO BEN 10.000.000 DI PASSAGGI!!!
- LA SICUREZZA DEI VS. DATI È ASSICURATA DALL'INECCEPIBILE SUPPORTO MAGNETICO DI PRIMISSIMA QUALITÀ.

PREZZO NETTO IMPOSTO

SS/DD 30 PEZZI	LIT. 2.550
SS/DD 100 PEZZI	LIT. 2.350
DS/DD 30 PEZZI	LIT. 3.400
DS/DD 100 PEZZI	LIT. 3.150

- I DISCHETTI DATAFLEX SONO DISPONIBILI IN CONFEZIONI DA 10/PZ. CON BOX IN PLASTICA MORBIDA (TIPO NYLON) ANTIURTO CHE NON SI ROMPE IN OCCASIONE DI CADUTE ACCIDENTALI.

INCREDIBILE!

DRIVE X APPLE 140K
TRAZ. DIRETTA SLIM
(TIPO TEAC) - LS-39A



L. 168.000



DRIVE CHINON
L. 288.000



DRIVE MITAC
FULL SIZE
L. 268.000

STAMPANTE X IBM

80 COLONNE
130 CPS
SET GRAFICO IBM
L. 399.000



STAMPANTE X IBM

132 COLONNE
130 CPS
SET GRAFICO IBM
L. 678.000



PC/XT COMPATIBILE L. 1.750.000



N. 2 DRIVE DS/DD 860K, CONTROLLER
MAIN BOARD 128K ESP. A256K,
ALIM. 130W., TASTIERA STAFF-K7

MONITOR PHILIPS X IBM	L. 227.000
MONITOR CABEL COLORI X IBM	L. 139.000
MONOCR GR. PRINT. HERCULES II	L. 364.000
COLOR GR. PRINTER	L. 395.000
MULTIFUNCTION 256K (ø RAM)	L. 274.000
512K RAM	L. 166.000
DRIVE 360K DS/DD	L. 299.000

**AMPIO ASSORTIMENTO DI
INTERFACCE APPLE IBM
TELEFONATECI**

EPROM WRITER X IBM L. 560.000

8255 CARD. X IBM
L. 290.000

ELITE - E (IIE COMPATIBILE)



L.
599.000

SCHEDA SISTEMA + 80 COL. + 64K L. 170.000
ELITE - E TASTIERA INTELLIGENTE
L. 690.000

ELITE - E TASTIERA STACCATA
* CABINET STAFF L. 880.000
* CABINET PC/XT L. 960.000

DATO L'INSTABILE MERCATO DEI CAMBI PREGASI TELEFONARE PER CONFERMA PREZZI E DISPONIBILITÀ
— RICHIEDETEVI IL CATALOGO —