



I trucchi del CP/M

a cura di Pierluigi Panunzi

I file

Ci siamo lasciati la scorsa puntata dopo aver visto le caratteristiche delle cosiddette "directory entry" relative ai file memorizzati su disco: abbiamo visto che ad ogni file memorizzato corrisponde un record formato da 32 byte, che descrive le caratteristiche del file stesso e dà utilissime informazioni sull'allocazione fisica dei vari "frammenti" in cui un file viene diviso dal sistema operativo CP/M.

In questa puntata ci occuperemo dei due tipi di file supportati dal CP/M, che, senza necessità di alcuno sforzo di immaginazione, troviamo essere i "file sequenziali" ed i "file random": non poteva che essere così, dal momento che questi tipi di file costituiscono una specie di standard e dato che la dualità sequenziale — random si trova praticamente a qualsiasi livello di programmazione, dall'Assembler ai linguaggi ad alto livello quali il Basic, il Pascal, ecc.

L'organizzazione dei file

Il CP/M, come detto, gestisce i file sequenziali e random, considerandoli entrambi costituiti da insiemi di "record" da 128 byte l'uno: di solito si identifica il concetto di "record" con quello di un settore da 128 byte. In genere però si lavorerà con "record logici" di lunghezza qualunque, che verranno sempre ricondotti da parte dei programmi applicativi ad uno o più "record" da 128 byte, che si potranno pensare perciò come "fisici". Comunque a prescindere da queste considerazioni, se vogliamo marginali, di solito, se i record contengono dei testi, allora ogni linea termina con una coppia di caratteri "Carriage Return" e "Line Feed" ed inoltre l'ultimo settore di un file di testo viene riempito di byte pari a 1AH.

Per lavorare perciò con un file, il CP/M ha bisogno di una struttura particolare, il "File Control Block" (che abbrevieremo con FCB), un insieme di byte contenenti parecchie informazioni riguardanti il file in questione e che ricorda un po' la struttura di una "directory entry".

Facendo riferimento alla figura 1, vediamo il significato dei singoli byte e le eventuali differenze con le "entry" che abbiamo conosciuto la scorsa puntata.

Innanzitutto la prima differenza la troviamo proprio nel primo byte, che in que-

sto caso rappresenta il drive in cui si trova il file desiderato, senza però che sia data informazione sull'"user number": in realtà infatti, quando si sta lavorando su di un file, il numero di user è ovviamente il "nostro" e perciò è inutile averlo memorizzato nell'FCB.

C'è da dire però che il drive viene indicato in una maniera inconsueta: un valore 0 indica che il file è posto nel disco di default corrente, settato o con una chiamata opportuna del BDOS oppure direttamente da tastiera: sappiamo però che, una volta cambiato il disco di default con il comando

B:

successivi riferimenti a file contenuti in tale disco non necessitano della "B:" davanti al nome del file. In questi casi il CP/M ed in particolare il BDOS considererà un valore 0 per il primo byte dell'FCB, conoscendo già da altre fonti di quale disco si tratti.

Invece un valore diverso da 0 indica proprio esplicitamente il drive: "1" corrisponde ad "A:", "2" corrisponde a "B:", ecc.

I successivi 8 byte sono riservati al nome del file, seguiti da altri 3 byte in cui è posto il "tipo" del file in questione, ancora una volta contenenti lo "stato" del file come nel caso delle "directory entry".

I successivi byte, ognuno avente un certo significato, possono generalmente essere posti a 0, in quanto saranno poi riempiti dal CP/M. In particolare gli ultimi tre byte, relativi ai file random al limite potranno anche mancare nel caso in cui decidiamo di gestire il file come sequenziale.

byte	significato
0	disk drive (0= default, 1= A:, 2= B:, ecc.)
1...8	nome del file
9...11	tipo del file
12	numero dell'extent
13,14	riservati al CP/M
15	record usati nell'extent corrente (max 128)
16...31	elenco degli "allocation blocks" usati
32	record sequenziale da leggere/scrivere
33,34	record random da leggere/scrivere
35	byte di overflow per record random

Figura 1 - Struttura del "File Control Block" (FCB), con il quale si comunicano al BDOS le caratteristiche del file su cui si opera.

Ecco dunque la prima sorpresa che avevamo preannunciato la scorsa puntata: per il CP/M, un file random è tale in quanto noi lo vogliamo gestire in tale modo, dato che non esistono file intrinsecamente random o sequenziali. In definitiva, a dispetto delle enormi differenze di gestione specie ad alto livello tra un file sequenziale ed uno random, i file stessi non hanno differenze "fisiche" e niente in essi li può far distinguere.

Analizziamo ora in dettaglio le differenze di gestione.

I file sequenziali

Come dice il nome, un file sequenziale può essere visto come una sequenza di "record" da 128 byte, consecutivi.

Per creare un file sequenziale bisogna perciò scrivere un FCB con il nome del file ed il suo tipo e poi bisogna richiedere al BDOS di "creare" tale file: per adesso parliamo in generale di tali chiamate al BDOS, mentre già fin dal prossimo numero continueremo ad analizzare in dettaglio le singole funzioni, tra le quali appunto c'è la "creazione" di un file.

Successivamente potremo richiedere al BDOS di "scrivere", record dopo record successivi settori di 128 byte nel file stesso: nel caso in cui si arrivasse al termine di un "extent" (del quale abbiamo parlato la scorsa puntata in dettaglio) il BDOS stesso avrà cura di aprirne uno successivo automaticamente, perciò senza nostro intervento.

Dopo aver scritto tutti i record del nostro file non ci resterà altro che "chiudere" il file stesso con un'altra chiamata al BDOS.

Viceversa per leggere un file già esistente, abbiamo ancora bisogno di un FCB indicante il nome ed il tipo del file, dopodiché richiederemo al BDOS di "aprire" il file indicato nell'FCB. Successivamente potremo leggere i record con le apposite chiamate al BDOS di "read sequential": tutto ciò sarà possibile fino a che arriveremo alla fine del file, della qual cosa ci accorgeremo sia analizzando i dati, sia quando il BDOS si accorgerà che i record di quel file sono finiti.

Fatto ciò, non ci sarebbe nessun obbligo di "chiudere" il file con un'apposita chiamata al BDOS, ma ciò è consigliato innan-

zitutto per completezza e poi perché in fondo è un'operazione innocua ed infine perché così si ha la compatibilità con l'MP/M, la versione multi-utente del CP/M.

Per quanto riguarda l'"aggiunta" di record ad un file sequenziale, vi sono due strade, delle quali la prima è quella di solito seguita nei linguaggi ad alto livello.

In particolare si può creare un nuovo file, copiare su di esso il contenuto del file da aggiornare e poi continuare a scrivere i record sul nuovo file, subito dopo la fine del file ricopiato.

Per fortuna però con il CP/M non è necessaria questa gestione di file "doppi", in quanto nell'FCB è possibile indicare il numero dell'"extent": facendo così, a seguito di una chiamata al BDOS per l'"apertura" di tale file, il BDOS stesso cercherà di aprire l'extent indicato e, se effettivamente lo troverà, allora potremo tranquillamente "scriverci" sopra. Dobbiamo però prima analizzare il byte "record usati nell'extent" (il byte n° 15) per vedere se ancora ce ne sono di liberi (cioè se tale valore è minore di 128).

Supponendo che tale valore sia inferiore a 128, allora basterà porlo nel byte n° 32 ("record sequenziale da leggere/scrivere") dell'FCB, forzando perciò il CP/M a scrivere nuovi record a partire da quello indicato e perciò proprio alla fine del file sequenziale originario.

Non ve l'immaginate un CP/M così potente?!

Comunque ritorneremo in dettaglio su tutti questi argomenti quando analizzeremo le funzioni del BDOS.

I file random

I file random si gestiscono in maniera molto simile a quella vista per l'altro tipo di file: la differenza principale è che nell'FCB si deve porre il valore del "numero del record random" nei byte nn° 33 e 34 ("record random da leggere/scrivere"): il BDOS terrà conto automaticamente del numero di extent all'atto della chiamata per "Read/Write Random" (ritorneremo nelle prossime puntate sull'argomento).

Comunque vediamo ora come si gestiscono i file random, dall'inizio. Innanzitutto, dopo aver creato il file come fatto per i file sequenziali, si deve porre, come detto, il numero del record nei due byte già visti

dell'FCB, prima della chiamata alla routine "Write Random" del BDOS.

A seguito di tale richiesta il BDOS effettuerà una serie di operazioni:

- leggerà il numero del record interessato

- calcolerà in quale extent si trova il record in questione

- se necessario creerà un'opportuna "directory entry" se l'extent non fosse già presente

- alla fine scriverà i dati nel record.

Sapendo che il BDOS effettua questo meccanismo autonomamente e perciò senza nostro intervento, ecco che potremo saltare qua e là per il file, ponendo dei record a caso nello spazio destinato al file, essendo sicuri che il CP/M creerà le necessarie "directory entry" non appena ci avventureremo in parti del file non ancora scritte.

Praticamente lo stesso meccanismo si ha nel caso in cui vogliamo leggere dei record da un file random: posto il numero del record desiderato nell'FCB, a seguito dell'opportuna chiamata, il BDOS aprirà il corretto extent del file e leggerà i dati: segnalerà invece errore nel caso in cui non trovasse quel particolare extent oppure il record stesso.

Bisogna a questo punto sottolineare un fatto sul quale porre molta attenzione.

Prima di effettuare una qualsiasi lettura o scrittura di un file random, bisogna "aprire" il file in questione al numero di extent 0, anche se tale extent non fosse effettivamente usato perché vuoto: ciò si ottiene, per un file nuovo, con la chiamata "Create File" del BDOS, mentre per un file già esistente basta la chiamata alla funzione "Open File" del BDOS.

Inoltre bisogna fare attenzione quando si creano file molto grandi, ma con pochissimi dati in esso e perciò con spazi vuoti tra i dati, rappresentati ad esempio da extent mancanti (in quanto non era stato necessario crearli per mancanza di dati).

In questo caso potrebbero aversi dei problemi con certi programmi che gestiscono sequenzialmente anche i file random: in questo caso, procedendo dall'extent 0, troverebbero degli extent parzialmente usati (mentre si aspetterebbero cioè solo alla fine fisica del file), decretando un'erronea fine del file stesso.

Peggio ancora si ha nel caso in cui manca un certo extent, per cui i successivi extent verrebbero ignorati.

NEW FOR THE
 

SCHEDE PER ACQUISIZIONE DATI

XAD-1 - catalogo: 060160
A/D converter 12 bit 4 canali 0-5V, orologio real-time, 4 uscite reed relay, tempo conversione 10 millisecc./canale.

XAD-2 - catalogo: 060161
A/D converter 12 bit con 2 canali guadagno variabile, 3 ingressi 0-5V, tempo conversione 10 millisecc./canale.

IIO4 - catalogo 060162
A/D converter 12 bit 25 microsec/canale 8 canali differenziali a guadagno variabile.

A/D D/A 12 bit - catalogo 060163
A/D converter 16 canali 0-9V 60 microsec/canale, D/A converter 12 bit tensione regolabile 1 microsec/conversione.

A/D 8 bit - catalogo 060165
A/D converter 16 canali 8 bit 0-5V 100 microsec/canale compatibile AI02.

A/D D/A 8 bit - catalogo 060166
A/D converter 16 canali 8 bit 0-5V 100 microsec/canale, D/A converter 1 canale 8 bit 0-5V 1 microsec/conversione.

D/A 8/16 bit - catalogo 060164
D/A converter 8 bit 2 canali con possibilità miscelazione canali, 0-10V uscita, 2 porte TTL in uscita.

A/D D/A 12 bit per IBM PC/XT - catalogo 095163
A/D converter 16 canali 12 bit 25 microsec/canale 0-9V, D/A converter 12 bit 0-9V 1 microsec/conversione.

DIGICODER per encoder ottici - catalogo 200110
Interfaccia per shaft-encoders 2 canali a 8 digit (100.000.000 di punti per assi X e Y di ROBOTS, TORNI, FRESE, etc.)

VIA CARD - catalogo 050122
I/O TTL con 2 6522 VIA per 32 linee totali, con contatori, shift/regs.

PARALLEL PORT - catalogo 050121
I/O TTL con 8255 per 24 linee totali, facilità di programmazione.

SUPER PARALLEL PORT - catalogo 050123
I/O TTL di potenza con 16 linee OUT 500 mA 50V e 16 linee IN optoisolate.

DX1 SOUND SAMPLING SYSTEM - catalogo 051130
Sistema acquisizione ed elaborazione di pattern sonori e sintetizzatore digitale con sequencer.

MIDI INTERFACE - catalogo 051120
Interfaccia per strumenti musicali e sintetizzatori professionali.

SCHEDE PER ACQUISIZIONE DA TELECAMERA

TELERASTER per APPLE - catalogo 050020
Digitalizzatore immagini video da telecamera 256 x 256 punti con 64 livelli di grigio.

TELERASTER per IBM PC/XT - catalogo 060022
Digitalizzatore immagini video da telecamera 256 x 256 punti con 256 livelli di grigio.

DIGICON per IBM PC/XT - catalogo 050024
Digitalizzatore a colori da telecamera 256 x 256 punti.

STRUMENTAZIONE DA LABORATORIO

I.C. TESTER CARD - catalogo 060127
Scheda con estensione zoccolo esterna per ricerca e prova micrologici serie TTL HCMOS MOS RAM.

EPROM WRITER HK128 - catalogo 060130
Scheda programmatore EPROM 2716, 2732, 2764, 27128 con 3 zoccoli zero-force, software operativo, manuale.

INOLTRE DISPONIAMO DI MICRO E PERSONAL COMPUTERS E RELATIVE PERIFERICHE COMPATIBILI APPLE II + APPLE// e IBM PC/XT.

RICHIEDETECCI CATALOGO COMPLETO.

 **PERTEL** PERIFERICHE TELECOMUNICAZIONI
TORINO - VIA ORMEA, 99 - TEL. 011/655.865
CONDIZIONI PARTICOLARI PER DEALER E HOBBISTI