

CD File System



Era un po' di tempo che la rubrica di memorie ottiche non trattava argomenti meramente tecnici, forse un po' più «duri», ma decisamente più interessanti.

L'appuntamento di questo numero vuole introdurre i File System dei CD-ROM, presentandone alcuni dei più significativi, sino al raggiungimento dello standard High Sierra.

Reprise CD

Cosa sia un CD-ROM è ormai cosa risaputa: un compact disc uguale a quello audio con memorizzate informazioni per computer anziché musica in forma binaria. Le capacità di questo tipo di memoria ottica sono notevolissime: circa 550 MegaByte di dati utente, con i difetti di essere a sola lettura, più lenta dei sistemi magnetici e poco costosa.

Prima di addentrarci nell'argomento di questa puntata è bene ricordare alcune delle caratteristiche salienti dei CD: tre sono le sezioni tipiche di un qualunque disco: la parte di inizio, l'area utente e quella di fine. Ciascuna sezione

contiene informazioni codificate in forma digitale: nella prima c'è la TOC (Table Of Contents) che indica il numero di tracce registrate ed il tempo che intercorre dall'inizio del disco a ciascuna traccia. Nell'area dati utente vi sono, ovviamente, tutti i dati che saranno adoperati dal fruitore del CD: informazioni, immagini, suoni, programmi. L'ultima area contiene le informazioni che indicano al player la fine della lettura, quindi inutili dal nostro punto di vista.

I dati nel CD sono codificati in una spirale che si svolge dall'interno all'esterno, con la possibilità di dividerli sino a 99 tracce. Il metodo di lettura è il CLV (Constant Linear Velocity) che legge a

1,25 m/s, scelto poiché consente la massima capacità di registrazione, ma d'altro canto l'accesso è più lento rispetto al metodo CAV, velocità angolare costante. Il CD ruota ad un numero di giri che è funzione della zona di lettura: da 200 a 430 al secondo. CD audio e CD-ROM funzionano sugli stessi principi e hanno caratteristiche fisiche uguali. Adoperano lo stesso sistema di modulazione detto «otto a quattordici» (EFM, Eight to Fourteen Modulation) e il metodo CIRC per la correzione degli errori. Le differenze fra i due tipi di dischi si trovano solo nel metodo di utilizzo del settore fisico dello stesso. Ogni settore fisico contiene 98 frame di 24 byte, che

generano 2352 byte, così utilizzati:

- 12 byte di Sync Field
- 4 byte di header
- 2048 byte di dati
- 288 byte ausiliari

Tutti i dati, tranne i byte di sincronismo sono interallacciati. La differenza fra i CD audio e quelli ROM sta negli ultimi 288 byte: i CD-ROM li usano per un ulteriore livello di correzione d'errore (il cosiddetto Mode 1), mentre nei CD audio vengono sfruttati per scrivervi degli altri dati utente (Mode 2). I due modi possono coesistere sullo stesso media: molti dei CD-ROM attuali contengono anche suoni.

File System

Il compito del File System è quello di fornire un supporto che consenta all'applicativo in uso di accedere alle informazioni sul CD. È compito del File System far vedere il disco come un insieme di file e non come una sequenza di settori da 2 kilobyte ciascuno, questo punto di vista è quello del controller hardware del drive, non certo del software. Se il software applicativo dovesse gestire l'accesso al disco in termini di indirizzo fisico del settore (minuti, secondi, settore), l'applicativo non potrebbe girare con null'altro che quel particolare CD-ROM. Se la nuova edizione dello stesso avesse i settori spostati, l'applicativo sarebbe inutilizzabile. Va da sé che un simile approccio è sbagliato, occorre quindi inserire un'interfaccia logica fra il controller e l'applicazione: il File System. Tale interfaccia consiste in programmi e strutture dati che devono

tradurre il formato fisico dei dati in uno utilizzabile dall'applicativo. Quindi dal punto di vista logico il disco è un insieme di file ciascuno col suo nome univoco, almeno all'interno della directory, una volta aperto il file il programma applicativo può leggere un singolo byte così come l'intero file, e ciò deve avvenire senza che l'applicativo sappia che i dati sono blocchi da 2k sul CD. Il risultato in uscita dal File System deve essere identico a quello di un disco magnetico.

I componenti fondamentali del File System sono tre:

- il formato logico, il quale determina come e dove mettere le informazioni identificative fondamentali sul disco, come sia strutturata la directory, eccetera. Questo non ha nulla a che vedere con il formato fisico, che è stato definito dalla

Philips e dalla Sony e abbiamo descritto nel paragrafo precedente, è infatti emerso come standard di fatto fra le aziende del settore («High Sierra Format») diventato poi norma ISO 9660.

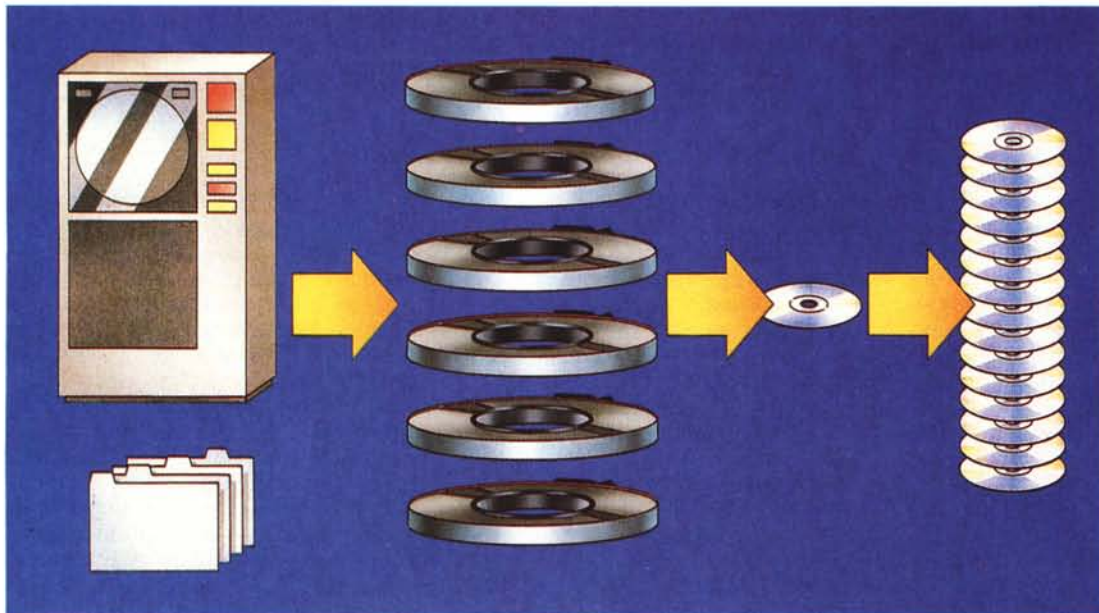
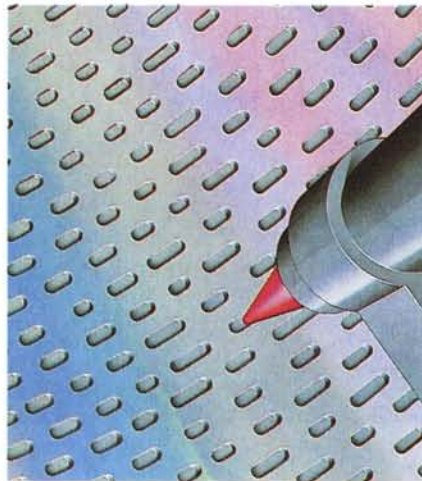
- Il software originante; infatti prima di fare un CD-ROM è necessario che i file da trascrivere siano «tradotti» secondo le regole del formato logico definito, così da poterli scrivere sul media ottico in maniera utilizzabile.

- Il software utilizzante: cioè quello che leggerà le informazioni dal disco, quindi quello che sarà adoperato dall'utente finale per poter utilizzare il CD-ROM, attualmente il più diffuso è quello della Microsoft: «MS-DOS Extension».

Differenze

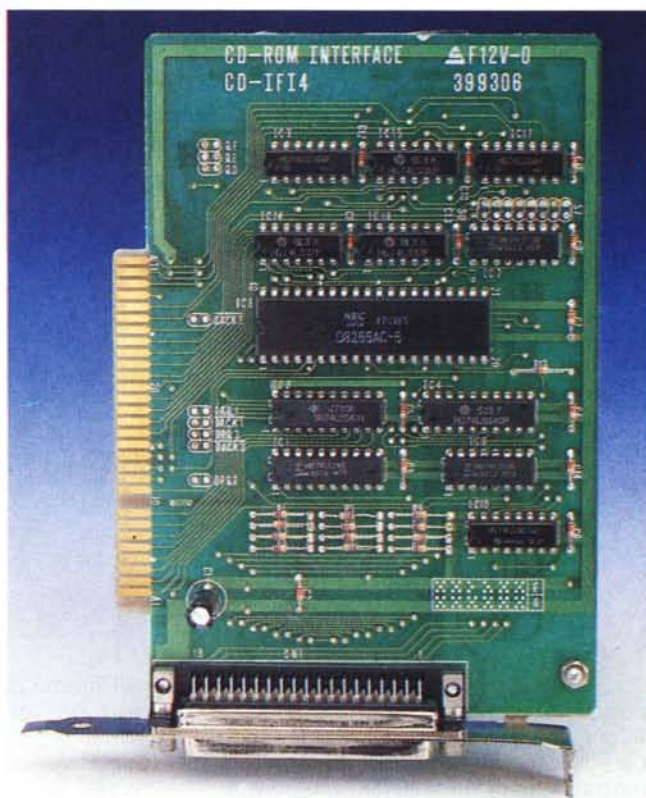
Per chiarire meglio i concetti relativi al File System è bene fare una rapidissima carellata sulle caratteristiche di un media ottico a sola lettura rispetto ad uno magnetico. In un hard disk il File System deve essere molto più flessibile, poiché deve consentire di cancellare un file e perciò rendere di nuovo disponibile quello spazio per qualcun altro, lasciare aggiungere informazioni ad un file già esistente, ma magari non è possibile scriverle nel settore adiacente perché è già occupato da un altro file, quindi il File System deve provvedere alla creazione di una mappa di puntatori alle varie locazioni fisiche (la FAT dell'MS-DOS), insomma deve gestire in modo molto più dinamico il tutto.

Sul CD-ROM non c'è la possibilità di cancellare o di aggiungere informazioni, tutto è già lì statico ed immutabile,



L'illustrazione mostra la struttura fisica del CD ROM. È compito del File System gestirlo ad hoc.

Il controller funge da interfaccia fisica tra il File System ed il drive.



quindi i compiti del sistema sono molto più semplici.

È anche importante notare che il CD-ROM è un media cosiddetto «asimmetrico», cioè non c'è simmetria fra le macchine che lo hanno scritto e quelle che lo leggono. Infatti i dati di un CD-ROM sono spesso processati da un computer potente in modo batch, mentre sono utilizzati su personal in modo interattivo. È quindi un'operazione relativamente poco costosa e molto saggia eseguire in fase di scrittura dati alcune funzioni che possono ottimizzarne poi la lettura: il costo è «one-time», mentre il beneficio appare ad ogni lettura. Un tipico esempio è quello di ordinare i file in ordine alfabetico, in maniera di accedere ad uno di essi, in fase di lettura, usando un algoritmo di ricerca dicotomica, decisamente più veloce di una ricerca sequenziale. Proprio il problema ricerca è particolarmente pesante sui media ottici, in quando una seek tipica di un CD-ROM è di mezzo secondo: eseguire una ricerca lineare con simili tempi di seek diventa un lavoro lentissimo ed inefficiente. Inoltre se un file fosse suddiviso in molti blocchi non adiacenti ci vorrebbe un'infinità di tempo e di accessi (di media mezzo secondo l'uno) per poterlo leggere e trasferire tutto.

Per risolvere questi problemi, il CD-

ROM è strutturato in modo da avere i file ordinati secondo certi criteri e tutti su blocchi continui, in modo tale da usare una sola seek per accedervi e sfruttare appieno i 150 Kbyte/secondo di transfer rate che consente.

Strutture

Il disegno della struttura delle directory è funzione del tipo di file che si utilizzeranno all'interno del sistema. Consideriamo la gestione di grossi file: in un sistema operativo quale l'MS-DOS se i dati non possono essere scritti contigualmente, vengono sparpagliati in molti pezzettini fisicamente distanti; ultimamente mi è capitato di avere grosso database in un unico file di oltre 10 Mega, che era stato splittato su varie parti del disco: l'applicativo che lo gestiva ci metteva circa 15 minuti a inserire un nuovo fascicolo: il database era formato da 27 tabelle con moltissime relazioni esplicite fra di esse, quindi per ogni inserimento, il programma doveva scrivere i dati nelle apposite tabelle e aggiornare tutti i puntatori delle explicit relationship. Questo lungo tempo era causato da un intensissimo accesso al disco, sul quale il file era suddiviso in troppi pezzi. È bastato fare un backup del file, cancellarlo, compattare l'hard disk e restorare il file, il quale era diven-

tato l'ultimo prima dei settori liberi e per di più era tutto in blocchi contigui, per scendere ad un tempo di aggiornamento di 3 minuti, compresa la scrittura di alcune decine di immagini su un WORM! Il tutto senza neanche ricompilare un programma, poiché il problema era causato dal File System dell'MS-DOS, non progettato per gestire grossi file in maniera efficiente. Il CD-ROM non ha questo tipo di problemi, in quanto in fase di scrittura dati si organizzano i grossi file in modo da averli tutti su blocchetti adiacenti. Il media, però, è particolarmente adatto a distribuire sia file enormi, al limite di 500 Mega, così come una grande quantità di piccoli file di dimensioni più consuete tipo 10.000 file da 50 Kilobyte. I problemi di gestione di un gran numero di file sono totalmente differenti da quelli prima esposti, ma sono anche i più comuni: MS-DOS, UNIX e VMS sono sistemi operativi ottimizzati proprio per gestire questo tipo di problematica. L'uso di subdirectory permette di avere tanti file quanti uno vuole, almeno finché non si incontra un limite fisico. Il CD-ROM è anche il media adatto per distribuire una grande quantità di file, ma come si può garantire l'accesso al file desiderato con il minimo numero di seek (ciascuna di almeno mezzo secondo)? Tutto sta a definire la struttura ed il metodo di gestione delle directory: con un sistema efficiente si può accedere al file con un solo accesso. I metodi più comuni sono 4:

- Sistema gerarchico esplicito a file multipli
- Sistema gerarchico esplicito a file singolo
- Directory con path a chiave hash
- Directory con path indicizzato.

Il primo è il più comune a familiare: UNIX, VMS, MS-DOS eccetera utilizzano questo metodo per la gestione dei dischi magnetici. Le subdirectory sono gestite come se fossero dei file e l'accesso ad un file richiede diverse seek, cosa non accettabile sui CD-ROM a causa della lentezza, ma trascurabile su magnetico. Per contro gestisce bene le ricerche con wildcard (es: dr*.txt), soprattutto se i file sono ordinati.

Il metodo a file singolo è una variazione semplice ma efficace del precedente: tutta la struttura delle directory è in un file solo, e le directory sono trattate come record all'interno del file. Un esempio di simile File System è basato sul cosiddetto «figlio a sinistra, paria a destra» per inserire subdirectory e file nella struttura record. In una simile configurazione, il puntatore di sinistra del record punta ad un elemento figlio nella subdirectory, mentre il destro punta o

ad un subdirectory o ad un file allo stesso livello, chiaramente i file non hanno puntatore sinistro (figli), perciò in quel campo viene memorizzato l'indirizzo del file stesso. Questo metodo permette di memorizzare l'intera gerarchia delle directory in un solo file quindi, se le dimensioni lo consentono, lo si può caricare in RAM e fare una sola seek sul file desiderato. Naturalmente se si hanno migliaia di file si hanno gli stessi problemi del metodo precedente ed anche gli stessi vantaggi. L'unico modo per garantire l'accesso al file scelto con un'unica seek è quello di creare una chiave hash per ciascun path e file. Questo metodo è efficacissimo per accedere ad un preciso file, in quanto l'accesso a chiave primaria è di una rapidità proverbiale, per contro la gestione di una ricerca generica è terribilmente onerosa. Chiedere "dir*.c" significa leggere tutti i record (file name) per determinare se il suffisso è «.c». Que-

ID Record	Directory	ID Padre
0	\	-1 (radice)
1	Angelo	00
2	Ugo	00
3	Mario	00
4	Testi	01
5	C	01
6	Prolog	02
7	Fax	03
8	Fatture	03
9	Src	05
10	Obj	05
11	Prj	06
12	Obj	06
13	In	08
14	Out	08

Figura 2

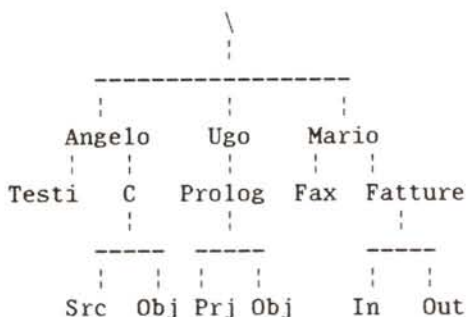


Figura 1

sto metodo è utile solo in certi casi, ma per applicazioni con CD-ROM deve essere coadiuvato da altri sistemi più consueti.

Il miglior sistema per File System su CD-ROM è quello a path indicizzato; il miglior metodo per descriverlo è quello di immaginare una semplice struttura di directory (vedi figura 1).

Consideriamo quindi una rappresentazione tabellare della stessa.

Path ID	Filename	Sector Start	Lenght
14	Acme.txt	16000	10
14	Acme2.txt	16010	50
14	Bingo.txt	16060	35
14	Folder.txt	16095	15

Figura 3

La tabella di figura 2 rappresenta la cosiddetta «path table», che ci permette di indentificare con un solo int di due byte il path che vogliamo. Il metodo di funzionamento è abbastanza semplice: poniamo di voler indentificare il path "\Mario\Fatture\Out": per prima cosa cerchiamo il path "Mario" nella path table e ne leggiamo il suo ID: 3, adesso puntiamo direttamente ai record che hanno per ID padre 03, sino a che non indentifichiamo "Fatture", il quale è rappresentato dal ID record 8. Ripetiamo il processo precedente per i record che hanno ID padre 03 e si chiamano "Out" e determiniamo il loro ID:14. A questo punto se volessi accedere al file "\Mario\Fatture\Out\Acme.txt", il sistema permetterebbe di tradurlo in un semplice "14 Acme.txt". Questo metodo di compressione del path name è efficacissimo: con soli due byte si possono rappresentare path lunghissimi, inoltre si possono memorizzare in 3 settori del CD-ROM (8Kbyte) ben 256 record. Organizzando file di una sottodirectory in ordine alfabetico e facendoli precedere dal «path ID», l'accesso alle informazioni diventa rapidissimo: consideriamo

che all'interno della subdirectory "\Mario\Fatture\Out" vi siano i seguenti file:

- Car.txt
- Acme2.txt
- Bingo.txt
- Folder.txt
- Acme.txt

Il File System dovrà avere una «file table» organizzata come in figura 3.

In questo modo l'accesso al file "\Mario\Fatture\Out\Acme.txt" si tradurrebbe in una prima ricerca per determinare il path ID, cioè 14, poi una seconda ricerca, dicotomica, fra i vari file che hanno lo stesso path ID determina quale sia il settore di start (16000) e il numero di settori (10) da leggere, ed ecco che si può inviare il comando di seek, uno solo: vai al settore 16000 (tempo di esecuzione meccanica, circa mezzo secondo) e leggi 10 settori, in mezzo secondo più 40 millisecondi l'accesso al file è stato effettuato.

Conclusioni

È chiaro che la soluzione a path indicizzato rappresenta la più facile e gestibile, soprattutto perché con pochi kappi di RAM si bufferizzano le tabelle, garantendo pochissimi accessi al disco. Le norme High Sierra hanno poi stabilito di consentire un massimo di 40 file per subdirectory, in modo da accelerare il più possibile il tempo di ricerca.

Finalmente a Roma un punto di riferimento professionale per chi si occupa di CAD e grafica! Kyber sarà presente

con la sua vasta gamma di soluzioni professionali, **Da oggi** per soddisfare le

esigenze di ogni categoria di **Kyber è anche** utenti, dalla

grande industria agli studi professionali di **a Roma** piccole e medie di-

mensioni, alla Kyber potrete trovare le stazioni grafiche GS-2000 e GS-3000, una vasta "scuderia" di plotters, a partire dal formato A0 di Ioline LP 3700, LP 4000 e del nuovissimo Mutoh F-910 che lavora anche con mine. E ancora stampanti grafiche, come la stupenda Accel-500 che ha numerose possibilità di emulazione nonché la capacità di lavorare come plotter HP-GL, la scheda grafica Leonard e numerose altre, monitors ad alta risoluzione, il pacchetto software DataCAD, il sistema da taglio HotPlot, e...

Kyber®

**SI CERCANO
RIVENDITORI**

VIA VIRGILIO, N. 30 - 00193 ROMA - TEL. (06) 6875878
SEDE CENTRALE: PISTOIA, VIA L. ARIOSTO, 28 - TEL. (0573) 368113