

Suddivisione in partizioni .....	2031
Inizializzazione di un'unità di memorizzazione .....	2032
Etichetta di un'unità di memorizzazione .....	2032
Analisi e correzione del file system .....	2033
Copia .....	2033
DISKCOPY .....	2033
XCOPY .....	2033
Trasferimento del sistema .....	2034
Modifica delle unità .....	2034
ASSIGN .....	2034
JOIN .....	2035
SUBST .....	2035
Altre particolarità .....	2035
VERIFY .....	2035
APPEND .....	2036
ATTRIB .....	2036
DELTREE .....	2036
FIND .....	2036
MOVE .....	2037
TREE .....	2037
COMP e FC .....	2037

La gestione dei dischi, ovvero delle unità di memorizzazione di massa, è molto particolare nel Dos. In generale, si fa riferimento a queste cose attraverso un lettera che ne rappresenta il dispositivo; tuttavia, tale dispositivo può indicare un disco intero o solo una partizione, riferendosi sempre solo a dischi e partizioni Dos.

### Suddivisione in partizioni

Nel Dos, tutti i dischi rimovibili, come i dischetti, non vanno suddivisi in partizioni, mentre i dischi fissi devono essere preparati in questo modo. Il programma che si usa per queste cose è 'FDISK'.

Secondo il Dos, le partizioni di un disco possono essere solo quattro. Tuttavia, una partizione normale può essere suddivisa in sottopartizioni, che vengono definite tradizionalmente «estese», dove anche queste possono essere al massimo quattro. Una tale struttura ha condizionato in pratica anche altri sistemi operativi, per esempio GNU/Linux. Bisogna tenere in considerazione l'origine storica per comprendere che altri sistemi operativi possono comportarsi in modo completamente differente.

Di solito, 'FDISK' ha una visione delle partizioni tutta orientata verso il Dos. Infatti, consente di creare una sola partizione primaria (ovvero una partizione normale) e altre partizioni estese (ovvero altre sottopartizioni di una seconda partizione primaria).

Bisogna considerare che il settore di avvio del Dos viene collocato nel primo settore della partizione primaria utilizzata per il Dos. In questo modo, manca la sistemazione del primo settore del disco, l'MBR, che deve contenere il codice necessario a raggiungere il settore di avvio.

```
FDISK [ /MBR ]
```

In generale, sembra che le varie edizioni di 'FDISK' per Dos funzionino solo con il primo disco fisso.

Fondamentalmente, il programma è interattivo, per cui si avvia una maschera con la quale si interviene per mezzo di un menù. Di norma

viene consentito di cancellare le partizioni, di crearne una primaria e probabilmente una sola di estesa.

Di solito, è possibile riscrivere il settore di avvio MBR attraverso l'opzione `'/MBR'`.

## Inizializzazione di un'unità di memorizzazione

L'inizializzazione di un'unità di memorizzazione, intesa come un dischetto o una partizione, si ottiene con il comando `'FORMAT'`. Questo si occupa anche di predisporre il file system Dos-FAT ed eventualmente anche di trasferire il kernel, per renderlo avviabile.

```
FORMAT lettera_unità : [/N:settori] [/T:cilindri] [/S] [/U]
```

In alcune edizioni del Dos, questo comando non inizializza l'unità di memorizzazione, ma si limita a sovrascrivere la parte iniziale. Ciò viene fatto per accelerare il procedimento e per permettere eventualmente il recupero dei dati, in caso di ripensamenti. In generale, sarebbe meglio evitare questa scorciatoia quando si tratta di unità corrispondenti ai dischetti; così, per confermare la richiesta di un'inizializzazione tradizionale, si può aggiungere l'opzione `'/U'`.

Segue la descrizione di alcuni esempi.

```
C:\>FORMAT A: /U [Invio]
```

Inizializza l'unità 'A:', corrispondente a un dischetto. L'inizializzazione avviene in modo completo, essendo stata usata l'opzione `'/U'`; inoltre, dal momento che non sono state indicate altre cose, il formato usato è quello predefinito in base alla configurazione del firmware.

```
C:\>FORMAT A: /N:9 /T:40 /U [Invio]
```

Come nell'esempio precedente, con l'aggiunta dell'indicazione della geometria: nove settori per traccia e 40 cilindri; si sottintende la presenza di due tracce per cilindro. Pertanto, dal momento che ogni settore è di 512 byte:  $2 * 40 * 9 * 512 \text{ byte} = 360 \text{ Kibyte}$ .

```
C:\>FORMAT A: /N:9 /T:80 /U [Invio]
```

Come nell'esempio precedente, ma con 80 cilindri:  $2 * 80 * 9 * 512 \text{ byte} = 720 \text{ Kibyte}$ .

```
C:\>FORMAT A: /N:15 /T:80 /U [Invio]
```

Come nell'esempio precedente, ma con 15 settori per traccia:  $2 * 80 * 15 * 512 \text{ byte} = 1200 \text{ Kibyte}$ .

```
C:\>FORMAT A: /N:18 /T:80 /U [Invio]
```

Come nell'esempio precedente, ma con 18 settori per traccia:  $2 * 80 * 18 * 512 \text{ byte} = 1440 \text{ Kibyte}$ .

```
C:\>FORMAT A: /S [Invio]
```

Inizializza il dischetto corrispondente all'unità 'A:', trasferendo successivamente il kernel e probabilmente anche l'interprete dei comandi (`'COMMAND.COM'`). Ciò avviene perché è stata usata l'opzione `'/S'`.

## Etichetta di un'unità di memorizzazione

Tradizionalmente, il Dos prevede la possibilità di attribuire un nome a un'unità di memorizzazione. Questo nome viene definito solitamente «etichetta» e di fatto viene annotato come un file speciale nella directory radice (anche se poi non appare nell'elenco). Per modificare o attribuire questo nome si utilizza il comando `'LABEL'`:

```
LABEL [ lettera_unità : ] [ nome ]
```

Se non si indica la lettera dell'unità di memorizzazione su cui intervenire, si tratta implicitamente di quella da cui è stato avviato il sistema; se non si indica il nome da attribuire, `'LABEL'` funziona in modo interattivo, chiedendo il da farsi.

In linea di principio, l'etichetta di un'unità non serve, salvo il caso di qualche programma che potrebbe utilizzarla per uno scopo particolare (per esempio i programmi di installazione per identificare i dischetti).

Esiste anche un altro comando interno per la verifica del nome di un'unità; si tratta di `'VOL'`:

```
VOL [ lettera_unità : ]
```

Il risultato è solo l'informazione del nome stesso, con l'aggiunta del numero di serie se questo dato è disponibile.

## Analisi e correzione del file system

Esistono pochi strumenti di analisi e correzione degli errori nel file system. In origine si tratta del comando `'CHKDSK'`, a cui in seguito si è aggiunto `'SCANDISK'`.

```
CHKDSK lettera_unità : [/F]
```

`'CHKDSK'` può essere usato solo con l'indicazione di un'unità di memorizzazione; in tal caso restituisce le informazioni disponibili su questa. Se si aggiunge l'opzione `'/F'`, si richiede esplicitamente la correzione, per quanto possibile, degli errori rilevati.

L'errore tipico di un file system Dos-FAT si traduce in «concatenamenti perduti», ovvero file, interi o parziali, di cui non si può conoscere il nome. Questi file potrebbero essere solo dati temporanei che è bene siano cancellati, ma questa non è la regola. `'CHKDSK'` tende a salvare questi file assegnando loro un nome più o meno casuale, lasciando all'utilizzatore l'onere di decidere cosa farne.

## Copia

Nei sistemi Dos la copia è un'attività piuttosto articolata. In pratica, il comando interno `'COPY'` consente solo di copiare file puri e semplici. Per copiare un dischetto occorre il comando `'DISKCOPY'`; per copiare file e directory occorre il comando `'XCOPY'`.

## DISKCOPY

```
DISKCOPY unità_di_origine : unità_di_destinazione :
```

`'DISKCOPY'` permette di eseguire la copia di un'unità di memorizzazione, purché si tratti di un dischetto. Il dischetto di destinazione dovrebbe essere inizializzato preventivamente.

L'unità indicata come secondo argomento, che rappresenta la destinazione, può essere la stessa di quella di origine. In questo caso, i dischetti vanno alternati nel dispositivo che li ospita, seguendo le istruzioni che dà `'DISKCOPY'` stesso.

L'esempio seguente esegue la copia di un dischetto usando lo stesso dispositivo fisico:

```
C:\>DISKCOPY A: A: [Invio]
```

## XCOPY

```
XCOPY percorso_origine [percorso_destinazione] [/E] [/S] [/H] [/V]
```

`'XCOPY'` consente di copiare uno o più file assieme alla struttura di directory. In altri termini, ciò significa che è possibile copiare anche una directory intera.

Tabella u180.1. Alcune opzioni.

Opzione	Descrizione
/S	Copia solo le directory e le sottodirectory non vuote.
/E	Copia tutte le sottodirectory, anche se vuote.

Opzione	Descrizione
/H	Copia anche i file nascosti e di sistema.
/V	Verifica la copia.

L'esempio seguente copia tutta la struttura che si articola a partire dalla directory '\PIPP0', nella directory '\PAPPA', includendo anche i file nascosti e quelli di sistema:

```
C:\>XCOPY \PIPP0\*.* \PAPPA\*.* /E /S /H /V [Invio]
```

## Trasferimento del sistema

Il Dos è un sistema operativo elementare. L'essenziale in assoluto è costituito dal kernel e dall'interprete dei comandi. Per rendere «avviabile» un dischetto o una partizione basta copiare questi file e sistemare il settore di avvio, in modo che punti correttamente al kernel. Questo si può ottenere con il comando 'FORMAT', quando lo si usa con l'opzione '/S' (cosa che naturalmente implica anche l'inizializzazione dell'unità), oppure con il comando 'SYS', fatto appositamente per questo:

```
FORMAT lettera_unità: /S
```

```
SYS lettera_unità:
```

A seconda del tipo di Dos, vengono copiati solo i file del kernel, oppure anche l'interprete dei comandi (necessario per avviare il sistema).

## Modifica delle unità

La caratteristica del Dos per cui si distinguono le unità di memorizzazione, introduce l'esigenza di comandi particolari, che vengono descritti brevemente nelle sezioni seguenti. In particolare, si tratta della possibilità di attribuire una lettera di unità differente e di poter inserire un'unità in una directory come avviene con l'innesto di un file system nei sistemi Unix.

## ASSIGN

```
ASSIGN lettera_unità_1[:]=lettera_unità_2[:]
```

```
ASSIGN /STATUS
```

```
ASSIGN
```

Il comando 'ASSIGN' permette di modificare il nome di un'unità di memorizzazione. Per ottenere questo risultato, rimane attivo come programma residente in memoria. Quando si usa senza argomenti, 'ASSIGN' elimina tutte le ridefinizioni; con l'opzione '/STATUS' si ottiene lo stato attuale delle ridefinizioni; quando si indicano le lettere di unità, la prima è l'unità virtuale che viene creata come riproduzione della seconda.

Segue la descrizione di alcuni esempi.

```
C:\>ASSIGN E:=A: [Invio]
```

Dopo questo comando, per accedere all'unità corrispondente al primo dischetto, è possibile indicare l'unità 'E:'.

```
C:\>ASSIGN [Invio]
```

Cancella tutte le ridefinizioni delle unità di memorizzazione.

## JOIN

```
JOIN lettera_unità: percorso
```

```
JOIN lettera_unità: /D
```

```
JOIN
```

Il comando 'JOIN' permette di attaccare un'unità di memorizzazione in corrispondenza di un percorso (una directory). Si tratta in pratica di innestare l'unità, come avviene nei sistemi Unix.

Quando si usa 'JOIN' senza argomenti, si ottiene un elenco degli innesti attivi; quando si usa l'opzione '/D', si vuole annullare il collegamento dell'unità.

Segue la descrizione di alcuni esempi.

```
C:\>JOIN A: C:\MNT\A [Invio]
```

Innesta l'unità 'A:' nella directory 'C:\MNT\A'.

```
C:\>JOIN A: /D [Invio]
```

Distacca l'unità 'A:' da un collegamento precedente.

## SUBST

```
SUBST lettera_unità: percorso
```

```
SUBST /D
```

Il comando 'SUBST' permette di creare un'unità virtuale a partire da una directory di un'altra unità. In pratica, si fa in modo di permettere l'identificazione di una certa directory attraverso l'uso di una lettera di unità.

Quando si usa 'JOIN' con l'opzione '/D', si vuole annullare l'unità virtuale relativa.

Segue la descrizione di alcuni esempi.

```
C:\>SUBST E: C:\EXTRA\E [Invio]
```

Crea l'unità virtuale 'E:' a partire dal contenuto delle directory 'C:\EXTRA\E'.

```
C:\>JOIN E: /D [Invio]
```

Elimina l'unità virtuale 'E:'.

## Altre particolarità

La gestione del Dos di file e directory è molto strana. Nelle sezioni seguenti vengono descritti alcuni programmi tipici dei sistemi Dos riguardanti la gestione di file e directory, che non hanno trovato un'altra collocazione in questo documento, a causa della loro particolarità.

## VERIFY

```
VERIFY [ON|OFF]
```

Il comando interno 'VERIFY' permette di richiedere al sistema operativo di verificare la registrazione nelle unità di memorizzazione. Come si vede dallo schema sintattico, si attiva o si disattiva la modalità, attraverso l'uso delle parole chiave 'ON' oppure 'OFF'. Di solito, questa modalità è disabilitata ed è difficile definire la reale importanza di questa impostazione.

Se si usa il comando senza alcun argomento, si ottiene di sapere quale sia l'impostazione attuale.

## APPEND

```
APPEND directory
```

```
APPEND ;
```

```
APPEND
```

Il comando **'APPEND'** consente di definire un percorso per la ricerca dei file di dati. In pratica, si vuole permettere ai programmi di accedere a file di dati anche quando questi si trovano fuori della collocazione prevista. **'APPEND'** può essere usato più volte, per aggiungere altre directory.

Se viene usato con l'argomento **';**, si intende cancellare tutto l'elenco di directory di ricerca dei file di dati. Se viene usato senza argomenti, si ottiene l'elenco di queste directory.

L'esempio seguente aggiunge la directory **'C:\DATI\'** all'elenco dei percorsi di ricerca per i file di dati:

```
C:\>APPEND C:\DATI [Invio]
```

## ATTRIB

```
ATTRIB [+R|-R] [+A|-A] [+S|-S] [+H|-H] file
```

Il comando **'ATTRIB'** permette di visualizzare o cambiare gli attributi del file. In pratica, utilizzando la forma **'+x'** si attiva l'attributo **x**, mentre con **'-x'** si disattiva l'attributo stesso.

Segue la descrizione di alcuni esempi.

```
C:\>ATTRIB *.* [Invio]
```

Mostra gli attributi di tutti i file contenuti nella directory corrente.

```
C:\>ATTRIB +R *.* [Invio]
```

Imposta l'attributo di sola lettura per tutti i file della directory corrente.

## DELTREE

```
DELTREE directory
```

Il comando **'DELTREE'** consente di eliminare una directory con tutto il suo contenuto, ricorsivamente.

L'esempio seguente elimina la directory **'C:\TEMP\CIAO\'** assieme a tutto il suo contenuto:

```
C:\>DELTREE C:\TEMP\CIAO [Invio]
```

## FIND

```
FIND [opzioni] "stringa" [file]
```

Il comando **'FIND'** è uno dei più complessi nei sistemi Dos. Serve per fare una ricerca di una stringa in uno o più file, in base a quanto indicato nell'ultimo argomento, oppure all'interno dello standard input. Il risultato normale della ricerca è l'emissione delle righe che contengono la stringa cercata, assieme all'indicazione del file a cui appartengono.

Tabella u180.2. Alcune opzioni.

Opzione	Descrizione
/V	La ricerca avviene per le righe che <b>non</b> contengono la stringa cercata.
/C	Mostra solo il totale delle righe che contengono la stringa cercata.
/N	Mostra il numero di ogni riga che contiene la stringa cercata.

Opzione	Descrizione
/I	Ignora la differenza tra maiuscole e minuscole per il confronto con la stringa di ricerca. In alcune edizioni del Dos, questa modalità di funzionamento è predefinita.

Segue la descrizione di alcuni esempi.

```
C:\>FIND "ciao" *.* [Invio]
```

Cerca la stringa **'ciao'** in tutti i file della directory corrente.

```
C:\>FIND "ciao" < MIO.TXT [Invio]
```

Cerca la stringa **'ciao'** nel file **'MIO.TXT'** che viene fornito attraverso lo standard input.

## MOVE

```
MOVE file_origine directory_destinazione
```

```
MOVE directory_origine directory_destinazione
```

Il comando **'MOVE'** consente di spostare file o directory in altre collocazioni. In generale, **'MOVE'** si occupa di spostare e non di rinominare i file, che invece è una funzione del comando **'REN'**.

Il comando **'MOVE'** è ambiguo e si comporta in maniera differente da una realizzazione all'altra dei sistemi Dos. In generale bisogna considerare che la destinazione può esistere o meno, implicando dei comportamenti differenti da valutare.

L'esempio seguente sposta i file e le directory contenute in **'C:\CIAO\'** nella directory **'C:\MIA\'**. Se la directory di destinazione non c'è, questa dovrebbe essere creata automaticamente, ma la cosa va verificata:

```
C:\>MOVE C:\CIAO\*.* C:\MIA [Invio]
```

## TREE

```
TREE [directory]
```

Il comando **'TREE'** consente di visualizzare la struttura della directory corrente, oppure di un'altra directory indicata come argomento.

L'esempio seguente mostra la struttura della directory **'C:\CIAO\'**:

```
C:\>TREE C:\CIAO [Invio]
```

## COMP e FC

```
COMP file_1 file_2 [opzioni]
```

```
FC file_1 file_2 [opzioni]
```

I comandi **'COMP'** e **'FC'** permettono di verificare se due file sono identici, oppure no. Non sono molto facili da utilizzare, specialmente il primo; probabilmente vale la pena di sapere che ci sono, senza poi pretendere di sfruttare tutte le loro possibilità.

**'FC'** assomiglia molto vagamente a un comando **'diff'** di Unix, dal momento che di fronte a file di testo cerca di comprendere quale cambiamento è stato fatto. In questo senso, è probabile che **'FC'** sia il più utile tra questi due.

