

## Il Sistema Operativo MS-DOS (MicroSoft Disk Operating System)

Sviluppato dalla MicroSoft è un sistema operativo per personal computer basati sulla famiglia di microprocessori *Intel iAPX-86*.

### Un po' di storia

- Il sistema operativo MS-DOS ha subito varie modifiche nel corso degli anni in corrispondenza a diverse versioni.
- La prima versione del sistema operativo MS-DOS era ispirata al CP/M (della Digital), una delle prime realizzazioni di sistema operativo per microcomputer.
- Dalla seconda versione in poi c'è stato un tentativo di uniformarsi alla filosofia del sistema **UNIX**, diventato uno standard di fatto per i sistemi di media dimensione.

- Innovazioni più evidenti che differenziano la seconda versione di MS-DOS dalla prima:
  - struttura gerarchica dei direttori;
  - dispositivi visti in modo omogeneo ai file (e non predefiniti);
  - maggiore organizzazione e capacità espressiva dei file comandi;
  - possibilità di ridirezione e di collegamento tra programmi (pipe).

### Differenza fondamentale

UNIX è un sistema multi-utente e multi-programmato, mentre MS-DOS è **mono-utente** e **mono-programmato**.

## MS-DOS: interfaccia utente

L'interazione utente-“macchina astratta” può avvenire attraverso:

1. comandi DOS (**Job Control Language** o JCL)
2. programmi “applicativi”
3. programmi di utente

### Esempi:

- comando MS-DOS

```
C:> DIR
```

- programma “applicativo” *TurboC*:

```
C:> tc
```

- compilatore C:

```
C:> gcc
```

- programma applicativo realizzato dall'utente (ad esempio, “prog\_1.exe” è il file prodotto dopo compilazione e linking):

```
C:> prog_1
```

## Linguaggio comandi

Le utilità del sistema operativo sono disponibili attraverso un linguaggio comandi direttamente interpretato da un **processore dei comandi**.

I processori dei comandi sono evoluti da semplici riconoscitori di comandi a supporti per ambienti più complessi, quasi veri e propri linguaggi di programmazione (vedi il caso degli shell di UNIX).

Nel caso di MS-DOS, l'ambiente comandi diventa attivo dopo il caricamento (**bootstrap**) del sistema operativo che risiede su un disco (detto **disco di sistema**).

### Bootstrap

Dopo l'accensione della macchina viene caricato in memoria centrale ed eseguito il codice di inizializzazione residente su ROM:

- 1) Attivazione e test dei componenti hardware del sistema.
- 2) Caricamento da disco ed esecuzione (di alcune parti) del sistema operativo (file IO.SYS e MSDOS.SYS, e COMMAND.COM)

L'attivazione del sistema operativo determina, in sequenza:

1. Lettura degli eventuali parametri di configurazione (file **"CONFIG.SYS"**).
2. Esecuzione della sequenza di comandi eventualmente contenuta nel file **"AUTOEXEC.BAT"**.
3. Attivazione del processore comandi (file **"COMMAND.COM"**).

☞ Alcune parti del sistema operativo sono residenti in memoria, mentre altre (alcuni comandi, detti *esterni*) vengono caricate all'occorrenza.

## Il processore comandi

Il processore comandi costituisce una macchina virtuale basata su:

- un gestore della console;
- un interprete dei comandi da eseguire (o parser)
- un caricatore (o *loader*) dei comandi.

I comandi sono suddivisi in due classi:

- comandi interni
- comandi **esterni**

### Comandi interni

- fanno parte di quella parte di sistema operativo caricata inizialmente in memoria centrale (sono integrati nel file **COMMAND.COM**) e quindi presentano un tempo di latenza minimo. Per questo motivo, sono interni i comandi più comunemente usati (ad es., DIR, DEL, COPY, ...)

### Comandi esterni

- vengono caricati all'occorrenza dal disco, prima di potere essere messi in esecuzione (corrispondono a file con estensione .COM ed .EXE). Vengono introdotti per non occupare stabilmente memoria con comandi raramente utilizzati (ad es., PRINT, FORMAT, ...).

☞ I programmi utente sono equiparati a comandi esterni. (file con estensione .EXE o .BAT).

## Schema di funzionamento del processore comandi di MS-DOS

```
repeat
{fase di bootstrap}
if <disco di sistema in A:> then
    <carica il S.O. dal disco in A:>
else
    if <disco rigido> then <carica da quello, default C:>
    else <azioni alternative>;
{ se c'è un disco di sistema presente nella drive A: lo si carica; altrimenti
in caso di disco rigido si carica da quello, oppure azioni alternative: per
esempio errore o esecuzione del BASIC caricato da ROM}
{ambiente comandi JCL}
repeat
<attesa di un comando dalla console>
if <comando interno> then
    case <comando> of
        DIR: <esegui l'immagine di DIR>;
        COPY: <esegui l'immagine di COPY>;
        ...
    end
else
    <ricerca del comando nel direttorio di default>
if <trovato file eseguibile> then
{ si ricerca un file di estensione .COM o .EXE come estensione di default}
    <caricamento ed esecuzione del comando>
else
    if <trovato file comandi> then
        {altrimenti si ricerca un file comandi con estensione
di default .BAT}
    else <segnala errore>;
{un <CTRL><BREAK> o un <CTRL>-C causa la terminazione del
comando correntemente in esecuzione ed un ritorno al command
processor}
until <re-inizializzazione: tasti <CTRL><ALT><DEL> >
until false {cioè si ripete per sempre}
```

## I comandi di MS-DOS

I comandi del sistema operativo MS-DOS possono essere classificati in:

- comandi di gestione dei **dischi** e dei **file**;
- comandi di interazione con le **risorse** del sistema;
- comandi di utilità e ausilio per lo **sviluppo** di un programma.

### Sintassi

<comando> [<opzioni>] [<argomenti>]

## Concetti base: drive, prompt

### Disk drive

È il dispositivo di interfaccia per un disco (sia dischetti floppy, che disco rigido)

- I disk drive sono individuati univocamente attraverso singole lettere dell'alfabeto. Di solito:
  - A e B per i floppy disk drive
  - C per il drive che agisce sul disco rigido.

### Drive di default

- È quello riferito preferenzialmente dal sistema nella fase di bootstrap (solitamente un floppy).

### Drive corrente

- Ad ogni istante è definito un **drive corrente**: è il drive relativo al disco che viene considerato in tutte le operazioni, se non se ne specifica esplicitamente un altro.

☞ Per **cambiare il drive corrente**, basta nominare il suo nome seguito dal simbolo di due punti (:).

## Prompt

- In un ambiente interattivo, il **prompt** è il simbolo (ad esempio, una sequenza di caratteri) che viene mostrato all'utente per indicare che si attende l'immissione di un comando.

```
nomedrive:\>
```

- Ad esempio, se il drive corrente è C:

```
C:\>
```

- Modifica del drive corrente (da C ad A):

```
C:\> A:  
A:\>
```

## Comandi per la gestione dei dischi

### format

Affinché un disco (fisso o floppy) sia utilizzabile dal DOS è necessario che sia **formattato**.

L'operazione di "formattazione" traduce il supporto magnetico grezzo in una forma strutturata suddividendo la superficie mediante tracce e settori.

Il comando MS-DOS per la formattazione dei dischi è:

```
C:\> FORMAT nomedrive:
```

**Nota bene:** Questa operazione ha un effetto distruttivo: tutte le informazioni eventualmente presenti nel disco (prima dell'esecuzione di **format**) vengono perse.

**Esempio:** se si desidera formattare un dischetto inserito nel drive A

```
C:\> FORMAT A:
```

## format

### Opzioni

```
FORMAT nomedrive: /S
```

crea un disco di sistema, copiando i file IO.SYS e MSDOS.SYS, e COMMAND.COM (il processore dei comandi MS-DOS).

```
FORMAT nomedrive: /V
```

attribuisce un nome (etichetta) al disco da formattare.

```
FORMAT nomedrive: /F:b
```

b specifica la capacità del disco da formattare (ad esempio, 360, 720, ...).

☞ Dalla versione 5.0 esiste il comando:

```
UNFORMAT nomedrive:
```

che consente il recupero dei dati sull'unità formattata. Va eseguito prima di qualunque operazione di scrittura. `FORMAT nomedrive: /U` rende inefficace tale comando.

## Il file

Per l'utente, il file è l'unità logica di memorizzazione sui dispositivi di memoria di massa.

- Ogni file è caratterizzato da un insieme di attributi:
  - un nome
  - una dimensione (in byte)
  - una modalità di accesso
  - una data

### Nome del file

- è costituito da: <prefisso>.<estensione>

TC . EXE  
P . PAS  
DATI  
WIN . COM

- prefisso è una stringa con lunghezza limitata a 8 caratteri;
- estensione è una stringa di lunghezza al massimo di 3 caratteri.

⇒ Esistono estensioni “notevoli”, che in DOS hanno un significato standard:

- “.exe” il file contiene codice eseguibile
- “.bat” il file contiene una sequenza di comandi (*file comandi*, o *batch file*)
- “.com” il file contiene codice eseguibile (da mantenere residente in memoria centrale)

## Contenuto di un file

Ogni file è costituito da una sequenza di caratteri organizzati in linee.

Es: **FILE DI TESTO:**

Lettera.txt

```
Spettabile Ditta,  
si comunica che  
il giorno 19/11...  
.....
```

Es: **FILE DI DATI:**

Stipendio.dat

```
ROSSI PIERO  6/2/93  2.000.000  
BIANCHI PAOLA  6/3/93  2.500.000  
.....
```

Es: **FILE DI PROGRAMMA ESEGUIBILE**

Mio prog. exe

```
***@ @ @ \ \ ^ ^
```

## Nomi di file: wildcards

Esistono alcuni caratteri “speciali” (metacaratteri) che non possono essere usati all’interno di nomi di file:

\ , | , / < , > , \* , , ? ...

### In particolare

- \* e ? vengono interpretati dal DOS come “WILDCARDS”: se inseriti all’interno di stringhe, vengono considerati come caratteri “JOLLY”:
  - \* rappresenta una qualsiasi sequenza (eventualmente nulla) di caratteri permessi.
  - ? rappresenta un qualsiasi carattere singolo.

### Esempi

- |                  |  |
|------------------|--|
| <b>PAOL?.DOC</b> | identifica tutti i file che iniziano con PAOL e che sono seguiti da un qualunque altro carattere |
| <b>PAOL*.DOC</b> | identifica tutti i file che iniziano con PAOL e che sono seguiti da una qualunque stringa        |
| <b>*.COM</b>     | identifica tutti i comandi con estensione.com: (format.com, backup.com, etc.)                    |
| <b>*.*</b>       | identifica QUALSIASI FILE!!!!  |



## File system

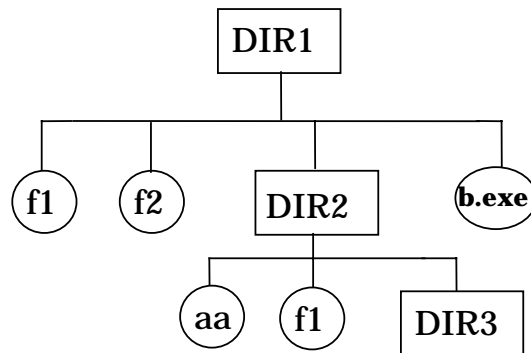
È la struttura logica all'interno della quale vengono organizzati i file nella memoria secondaria.

Il file system del sistema operativo DOS ha una struttura **gerarchica** ad albero.

La strutturazione del file system è ottenuta mediante la suddivisione in **direttori**.

### Direttorio:

- insieme di files;
- insieme di files e direttori (*sottodirettori*).



## Il File System

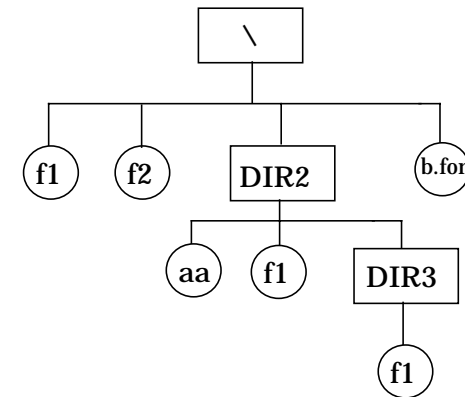
### Struttura gerarchica ad albero

- **nodi**, direttori
- **foglie**, files
- **rami**, appartenenza

☞ Non si pongono limiti al grado di innestamento della struttura gerarchica che si può costruire.

### Direttorio radice: \

È la “radice” dell’albero:



## Direttorio Corrente

- Ogni utente può “navigare” attraverso il file system, spostandosi di direttorio in direttorio. Ad ogni istante, quindi, è univocamente definito il **direttorio corrente**, che rappresenta la posizione attuale dell’utente all’interno della gerarchia.

## Identificazione di file (e direttori)

- In generale ogni file (ed ogni direttorio) è identificato univocamente da un cammino (**pathname**) più il suo nome, separati tra di loro dal carattere “\”.

Il cammino è:

- assoluto, se riferito alla radice;
- relativo, se riferito al direttorio “corrente”.

Inoltre, per indicare a quale file system (disco) si fa riferimento, si premette il nome del drive.

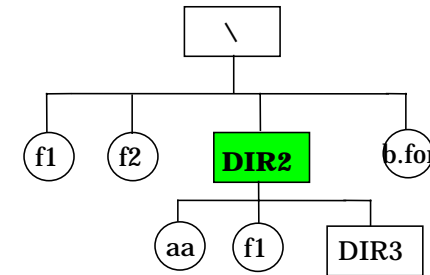
**[nome drive:][path]prefisso.estensione**

Le parentesi quadre indicano parametri opzionali

C:\f1  
C:\DIR2\f1  
C:\DIR2\DIR3\f1

## Esempio:

Consideriamo il file **aa** che si trova nel seguente albero del disco nella drive A:



## Ipotesi: DIR2 è il direttorio corrente

- nome relativo: **aa**
- nome assoluto: **A:\DIR2\aa**

## Notazioni convenzionali

- il direttorio corrente è identificato dal punto (“.”);
- il direttorio padre di quello corrente dal doppio punto (“..”);
- il direttorio radice viene indicato con “\”.

## Ad esempio:

consideriamo ancora aa, con direttorio corrente DIR3

- nome assoluto: **A:\DIR2\aa**
- nome relativo: **..\aa**

☞ Il nome assoluto è indipendente dal direttorio corrente!

## Il nome del file

Quindi

`[nomedrive:][path]prefisso.estensione`

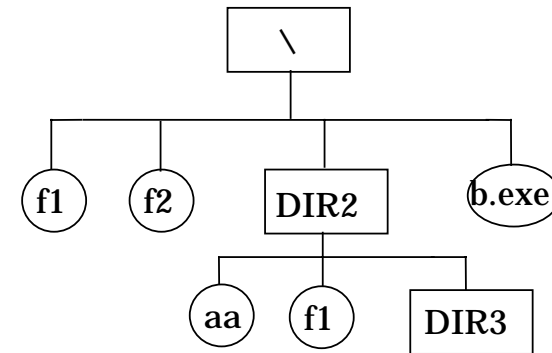
Significato dei singoli elementi:

- **nomedrive** è la lettera che individua il drive (è opzionale se va bene quello di default);
- **path** è il cammino a partire dalla radice dell'albero dei sottodirettori (nel caso di nome assoluto) o a partire dal direttorio corrente (nel caso di nome relativo). Anche path è un elemento opzionale;

**File (o direttori) distinti:**

- possono avere nomi relativi uguali
- non possono avere nomi assoluti uguali

Ad esempio, i file **f1**:



**A: \f1**

**A: \DIR2\f1**

## Comandi per la gestione del file system

### DIR [nomedrive:][path] [opzioni]

Visualizza la lista dei file e dei direttori contenuti in un direttorio.

#### Opzioni:

- /P la lista delle informazioni è fornita a videate
- /W vengono visualizzati solo i nomi dei file e dei sottodirettori

☞ DIR accetta le wildcards.

### CD [nomedrive:][path]

Cambia il direttorio corrente. Senza parametri, visualizza il direttorio corrente.

### MKDIR [nomedrive:]path

Crea un nuovo sottodirettorio (abbreviazione MD).

### RMDIR [nomedrive:]path

Distrugge un sottodirettorio, solo se è vuoto (abbreviazione RD).

### TREE [nomedrive:][path]

Visualizza la struttura gerarchica del file system.

## Esempi DIR:

### C:\> DIR

Il volume nell'unità C è DOS400  
Il numero di serie del volume è 1666-13D7  
Directory C:\

COMMAND	COM	38523	19/04/89	12:00
WS4	<DIR>		12/03/93	19:11
WIN31	<DIR>		15/03/93	11:12
DOS	<DIR>		26/10/90	12:17
AUTOEXEC	BAT	439	20/09/93	10:37
WINW	BAT	128	12/05/93	16:19
SETHOST	DAT	690	18/05/93	16:04
TP6	<DIR>		14/09/93	18:36
APPLE	BAT	896	21/05/93	11:00
APPLE	<DIR>		06/04/93	13:29
CONFIG	SYS	338	27/04/93	12:00
		12 file	9037824 byte disponibili	

### C:\> DIR a:

Il volume nell'unità A non ha etichetta  
Il numero di serie del volume è 1D48-1CCF  
Directory A:\

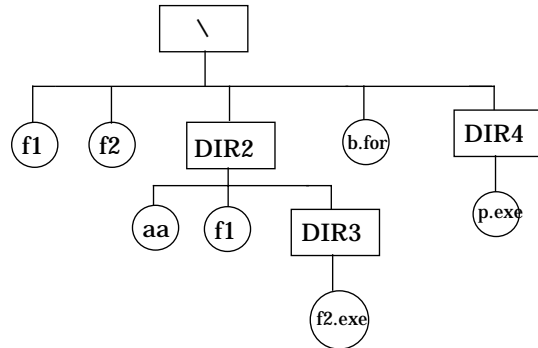
DIR		1951	30/09/93	11:02
PASCAL2	<DIR>		28/09/93	19:32
SORT	DOC	19126	24/11/92	11:49
FONDAEXB	DOC	263353	14/12/92	20:47
PASCAL1	<DIR>		30/09/93	9:45
ORDINA	<DIR>		30/09/93	9:51
		6 file	379392 byte disponibili	

### C:\> DIR A:\PASCAL1\\*.PAS

Il volume nell'unità A non ha etichetta  
Il numero di serie del volume è 1D48-1CCF  
Directory A:\PASCAL1

CICLO	PAS	222	28/09/93	19:29
ESEMPIO	PAS	572	28/09/93	19:30
FATT1	PAS	182	28/09/93	19:30
MAGGIORE	PAS	336	28/09/93	19:30
		9 file	377856 byte disponibili	

**Esempi:** consideriamo il file system sul disco A:



```

C:\> A:
A:\> CD DIR2
A:\DIR2> CD DIR3
A:\DIR2\DIR3> DIR
F2 EXE 34899 02/09/1992 16:01
  
```

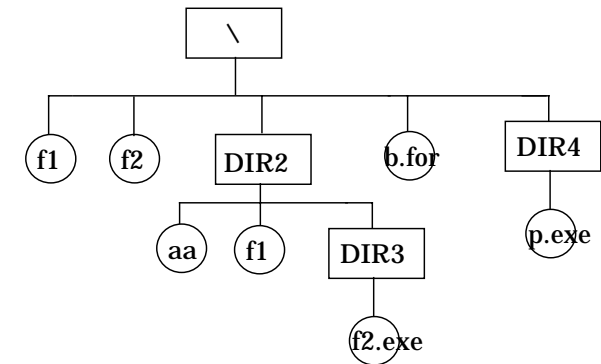
questa serie di comandi equivale a:

```
C:\> DIR A:\DIR2\DIR3
```

oppure a:

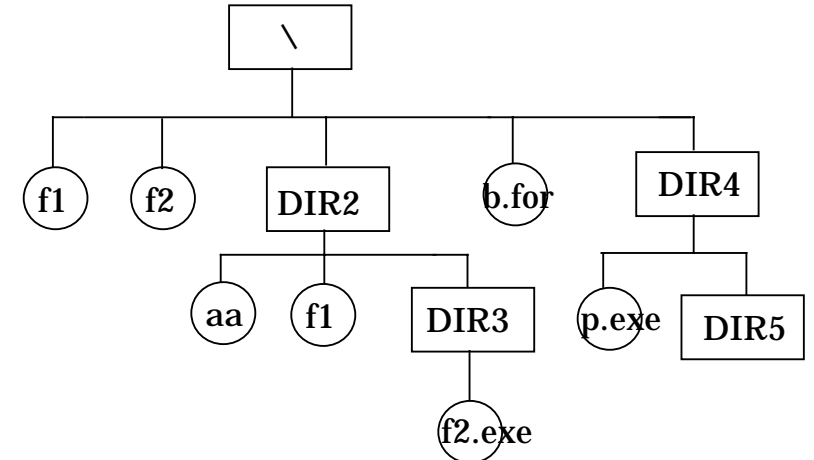
```

C:\> CD A:\DIR2\DIR3
C:\> DIR A:
  
```



```

A:\> CD \DIR2\DIR3
A:\DIR2\DIR3> CD ..
A:\DIR2> CD ..\DIR4
A:\DIR4> MD DIR5
  
```



## Comandi di gestione dei file

### TYPE [nomedrive:][path]nomefile

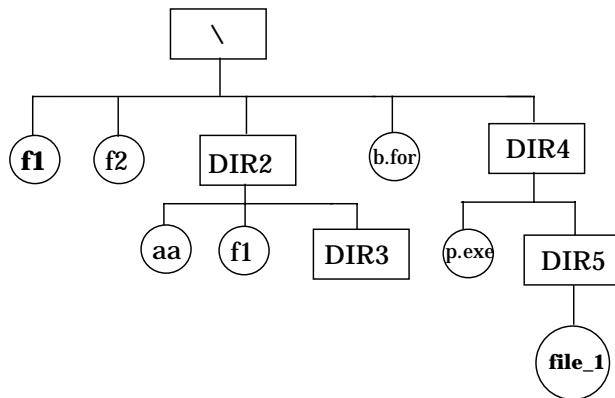
Visualizza il contenuto di un file.

```
A:\> TYPE DIR2\f1
```

### COPY <file\_sorgente> <file\_destinazione>

Consente di copiare un file in un altro. I nomi dei file (sorgente e destinazione) possono essere specificati in modo relativo o assoluto.

```
A:\> COPY f1 A:\DIR4\DIR5\file_1
```



```
A:\> COPY *.* C:\
```

### COPY <file1> + <file2> [ + ...] [<file\_dest>]

Concatenamento di file. Per copiare più file in uno singolo (file\_dest).

```
A:\> COPY f1 + f2 + b.for tutto
```

☞ Se <file-dest> è omissso, allora si copia nel primo file, che viene così modificato.

```
A:\> COPY f1 + f2 + b.for
```

### XCOPY <path-sorg> <path-dest> [<opz>]

Copia file e sottodirettori (opzioni) appartenenti alla gerarchia specificata.

#### Opzioni:

/S file + sub-directory non vuoti

/E file + sub-directory

/D: data copia i file modificati dopo la data specificata.

#### Ad esempio:

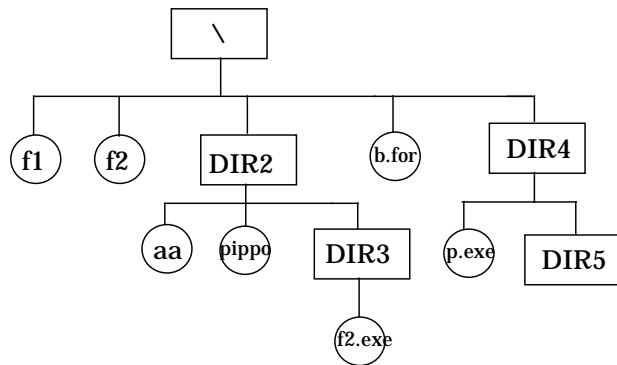
```
A:\> MD \temp
```

```
A:\> XCOPY \DIR2 \temp /E
```

## REN [nomedrive1:][path]nomefile1 nomefile2

Consente il cambiamento del nome di un file.

```
A:\> REN \DIR2\f1 pippo
```



## DEL [nomedrive:][path]nomefile

Permette la cancellazione di uno o più file.

☞ DEL accetta le Wildcards:

```
C:\> DEL *.BAK
```

Elimina tutti i file nella Directory corrente con estensione “.BAK”.

```
C:\> DEL *.*
```

Elimina **tutti** i file contenuti nel direttorio corrente, ma non elimina i sottodirettori.

**(Da utilizzare con cautela!!)**

## Confronto di file:

### FC <nomefile1> <nomefile2>

Confronta il contenuto di due file e ne visualizza le differenze.

### COMP <nomefile1> <nomefile2>

Confronta il contenuto di due file e fornisce (in caso di differenze) la posizione ed i valori (esadecimali) dei caratteri diversi.

## Stampa di file:

### PRINT [<pathname>] [<opzioni>]

Stampa il contenuto di uno o più file. Accetta le wildcards. La stampa avviene in *background*: questo significa che, durante la stampa, il sistema MS-DOS accetta altri comandi.

### Opzioni:

/D:periferica	Specifica la Stampante
/C	Sospende la stampa
/T	Termina la stampa (svuota la coda)
/P	Aggiunta di file nella coda di stampa

### Ad esempio:

```
C:\> PRINT autoexec.*
```

```
C:\autoexec.bat in stampa
```

```
C:\autoexec.old in coda
```

```
...
```

```
C:\> PRINT autoexec.old /C
```

☞ Se non viene specificato il nome del file, il comando PRINT mostra la situazione della coda di stampa.

### In alternativa a PRINT:

```
C:\> TYPE AUTOEXEC.* >> LPT1
```

## Modalità di accesso ai file

È possibile specializzare il tipo di accesso a file (sola lettura, archiviazione).

### ATTRIB [<opzione> <pathname>

Modifica o imposta le modalità di accesso ai file.

### Opzioni:

- |           |   |
|-----------|---|
| <b>+R</b> | imposta l'attributo di <b>sola lettura</b>  |
| <b>-R</b> | cancella l'attributo di <b>sola lettura</b>   |
| <b>+A</b> | imposta l'attributo di <b>archiviazione</b> che rende possibile la memorizzazione di file in formati più compatti (vedi BACKUP) |
| <b>-A</b> | cancella l'attributo di <b>archiviazione</b>  |

☞ **ATTRIB <file>** visualizza gli attributi del file specificato.



## Gestione delle periferiche

### Dispositivi come file

Ad alcuni dispositivi sono associati dei nomi particolari (CON: per la console, PRN: per la stampante) e operazioni simili a quelle su file (omogeneità con i file).

### Ad esempio:

- È possibile copiare il contenuto di un file sul video:  
  
`COPY FILE.DAT CON:`
- Copiare le informazioni date in ingresso dalla console in un file:  
  
`COPY CON: FILE.DAT`
- La terminazione dello standard input viene ottenuta tramite la combinazione dei tasti <CTRL>-Z.
- Inviare (copiare) il contenuto di un file alla stampante.

`COPY FILE.DAT PRN:`

## Comandi di gestione dei dischi

### DISKCOPY [nomedrive1:] [nomedrive2:]

Esegue una copia FISICA di dischi UGUALI, (es.: 3,5" con 3,5"). Se si vogliono copiare file da dischi diversi usare il comando COPY od XCOPY.

Se il disco destinazione contenuto in nomedrive2 non è 'formattato', esso viene inizializzato allo stesso modo del disco sorgente.

`C:\> DISKCOPY a: b:`

### CHKDSK <Pathname>

Restituisce informazioni relative allo stato di allocazione di un disco (o frammentazione di un file)

`CHKDSK <nome unità>`  
`CHKDSK <nome file>`

### Opzioni:

`/F`                      ricompatta in file i byte frammentati sul disco e recupera le allocazioni perse.

### DISKCOMP <sorgente> <destinazione>

Confronta il contenuto di due floppy disk.

```
C:\> DISKCOMP a: b:
```

### BACKUP <path-sorg> <drive-dest> [<opzioni>]

Viene utilizzato per il salvataggio di file o insiemi di file e sotto directory su **dischetto** (o su una serie di dischetti).

Differisce da COPY e XCOPY perché:

- a) la direzione della copia è fissa: da DISCO RIGIDO a DISCHETTO.
- b) I file vengono salvati in un formato **non leggibile**.
- c) Un comando di backup può coinvolgere più dischetti (se la quantità di byte da copiare eccede la capacità di un dischetto); i file che eccedono lo spazio disponibile sul disco vengono spezzati e copiati in parte sul disco “corrente” ed in parte sul successivo.

L'utilizzo tipico di backup è quello di effettuare il salvataggio di file di grandi dimensioni, o di effettuare il salvataggio del contenuto dell'intero disco rigido.

#### Opzioni:

- |    |  |
|----|--|
| /S | copia anche le sotto-directory   |
| /A | aggiunge sul dischetto di destinazione senza cancellare gli eventuali file pre-esistenti |
| /M | copia solo i file modificati dopo l'ultimo backup.                                       |

#### Esempio:

```
C:\> BACKUP c: a: /S
```

### RESTORE <drive-sorg.> <dest> [<opzioni>]

Il comando RESTORE è complementare al comando BACKUP: ripristina i file (e/o i direttori) salvati con un backup.

#### Opzioni:

- |    |   |
|----|---|
| /S | ripristino di eventuali sotto directory                           |
| /M | ripristino dei soli file modificati dopo il backup                |
| /N | ripristino dei soli file non più esistenti sul disco destinazione |

#### Esempio:

```
C:\> RESTORE a: C:\DIR2 /S/M
```

## Comandi di interazione con il sistema

### Comandi di stato del sistema

Sono i comandi che consentono di ottenere informazioni sull'attuale configurazione del sistema, ed, eventualmente, di modificarla.

#### ECHO

Consente di attivare (ECHO ON) o disattivare (ECHO OFF) la visualizzazione dei caratteri in ingresso che vengono forniti da console.

#### CLS

Effettua la pulizia dello schermo, lasciando solo il prompt di MS-DOS.

#### DATE, TIME

Consentono la visualizzazione o la modifica della data e dell'ora del sistema.

```
C:\>DATE
```

La data attuale è 28/2/01

```
C:\>DATE 30-3-96
```

(impostazione nuova data)

#### KEYB

Consente il caricamento di un particolare programma di gestione della tastiera.

```
C:\>KEYB IT
```

### Comandi di stato dei dispositivi:

#### MODE

Il comando MODE, a seconda dei suoi parametri, determina lo stato e la modalità di gestione dei vari dispositivi.

```
C:\> MODE LPT1: 80, 6, B
```

Specifica che la stampante collegata alla porta parallela 1 deve considerare una lunghezza di riga di 80, con spaziatura tra le righe di 6 pollici; l'ultimo parametro specifica di ritentare le operazioni di scrittura in caso di non risposta.

## Gestione di variabili di sistema

### Variabili di sistema

In MS-DOS è possibile dichiarare ed utilizzare variabili (“di sistema”).

Le variabili di sistema possono essere riferite solo all’interno di file eseguibili:

- COMANDI
- FILE COMANDI
- PROGRAMMI

Il comando MS-DOS per dichiarare ed utilizzare variabili “di sistema” è:

### **SET [<variabile> = <valore>]**

- L’argomento esprime quale valore assegnare alla variabile specificata. Il valore è una sequenza di caratteri (stringa).
- Senza argomenti visualizza i valori correnti delle variabili dichiarate.

```
C:\>SET MIODIR = A:\DIR2
```

☞ In MS-DOS esistono variabili di sistema predefinite:

- **PATH** rappresenta i cammini preferenziali di ricerca dei file eseguibili.
- **APPEND** rappresenta i cammini preferenziali di ricerca dei file di dati;
- **PROMPT** rappresenta il prompt di sistema operativo
- ...

```
C:\>SET PROMPT=Anna $p $g
```

```
Anna C:\>
```

## Path

Se il nome di un programma (o di un comando esterno) viene dato in modo relativo, esso viene ricercato nel direttorio corrente, a meno che non sia definito un cammino di ricerca (variabile di sistema PATH). In tal caso, la ricerca viene estesa anche ai direttori specificati nel path.

### PATH [<dir 1>;<dir 2>;...;<dir n>]

È un comando che permette di indirizzare automaticamente la ricerca di:

- COMANDI
- FILE ESEGUIBILI
- FILE COMANDI (.BAT)

verso cammini ("path") preferenziali.

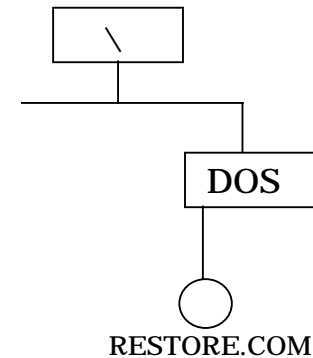
Il comando PATH modifica il valore della variabile di sistema PATH.

- Se non viene fornito nessun argomento visualizza la impostazione corrente (cammini di ricerca correntemente in uso).
- Se non viene impostato alcun cammino preferenziale:

PATH = <Directory corrente>

☞ Successivi richiami di PATH determinano la perdita delle impostazioni precedenti.

### Esempio PATH:



```
C:\>PATH  
PATH=\  

```

```
C:\> RESTORE a: c:  
"comando o nome file non valido"
```

```
C:\> PATH C:\Dos  
C:\> RESTORE a: c:
```

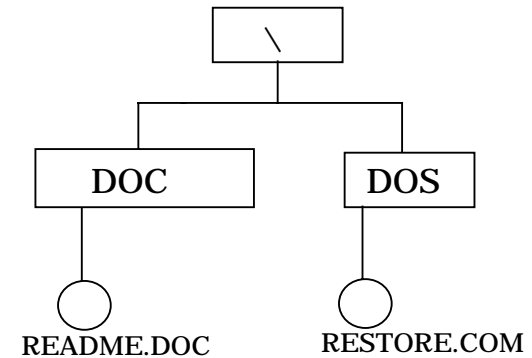
## APPEND [<dir 1>;<dir 2>;...;<dir n>]

- Permette di indirizzare automaticamente la ricerca di FILE DI DATI verso cammini preferenziali.
- Aggiorna il valore della variabile di sistema APPEND.
- Se non viene fornito nessun argomento, visualizza la impostazione corrente.
- Se non viene impostato alcun cammino preferenziale:

APPEND = <Directory corrente>

☞ Successivi richiami di APPEND determinano la perdita delle impostazioni precedenti.

## Esempio APPEND:



```
C:\>APPEND  
Append=C:\
```

```
C:\> TYPE README.DOC  
"file non trovato"
```

```
C:\> APPEND C:\DOS; C:\DOC;  
(impostazione nuovo cammino)
```

```
C:\> TYPE README.DOC  
"..." (visualizza il file readme.doc)
```

### Esempio:

```
C:\> SET PATH=C:\DOS
```

[equivale a C:\> PATH C:\DOS]

```
C:\> SET APPEND =C:\DOC
```

[equivale a C:\> APPEND C:\DOC]

```
C:\> SET PROMPT=$P$G
C:\> SET MIODIR=A:\DIR2
C:\> SET
PATH=C:\DOS
APPEND =C:\DOC
PROMPT=$P$G
MIODIR=A:\DIR2
```

## Ridirezione dei comandi

### Comandi come filtri

In generale, ogni comando accetta dati in ingresso *input* e produce risultati in *output*.

- *Standard Input*: la tastiera
- *Standard Output*: il video



### Ridirezione

È possibile sostituire i canali standard di input e di output con file o dispositivi.

### In particolare

Sono stati introdotti alcuni *metacaratteri* per esprimere la ridirezione dei comandi:

- < esprime la ridirezione in ingresso
- > esprime la ridirezione in uscita.
- >> esprime la ridirezione in uscita, in modo append: l'uscita avviene su un file preesistente, senza alterarne il precedente contenuto, aggiungendo le nuove informazioni di seguito all'ultimo byte del file.

## Esempi di ridirezione in uscita:

```
C:\> DIR B: > FILE.DAT
```

il contenuto del direttorio è inserito nel file specificato.

```
C:\> TYPE PIPPO > LPT1
```

l'uscita del comando "TYPE PIPPO" (il contenuto di PIPPO) è ridiretta verso la stampante LPT1.

Equivale ad un comando di stampa in "foreground" (attesa fino al completamento della stampa).

```
C:\> TYPE PLUTO >> PIPPO
```

concatena il file PLUTO con PIPPO.

```
C:\> COMP FILE1.PAS FILE2.PAS >> COMP.DAT  
C:\> COMP FILE3.PAS FILE4.PAS >> COMP.DAT
```

nel file COMP.DAT sono via via aggiunte le uscite dei due comandi di confronto.

## Ridirezione in ingresso: comandi filtro

Un filtro è un comando che accetta input da:

- tastiera
- file
- dispositivi
- etc.

I filtri MS-DOS sono SORT, FIND, MORE.

### **SORT** < <input>

Ordina alfabeticamente le linee di un file ASCII avuto per input.

```
C:\> SORT < PIPPO
```

### **FIND** "<stringa>" < <input>

Ricerca una <stringa> in un file di input; se la <stringa> è presente, viene visualizzata la linea che la contiene.

```
C:\> FIND "PATH" < AUTOEXEC.BAT  
"PATH = C:\DOS;"
```

ricerca la parola PATH all'interno del file AUTOEXEC.BAT

### **MORE** < <input>

Esegue una visualizzazione a pagine dell'input.

```
C:\> MORE < PIPPO
```



## Esempi:

```
C:\> SORT /R < FILE.SRT > INVFILE.SRT
```

il file in ingresso è ordinato con ordinamento inverso (grazie all'opzione /R) e posto nel file specificato in uscita.

```
C:\> FIND "ciao" < FILE.DAT
```

cerca la stringa “ciao” nel file FILE.DAT: riporta le righe che la contengono.

```
C:\> MORE < FILE1.PAS
```

visualizza (una pagina alla volta) il contenuto del file specificato.

## Piping di comandi

È possibile concatenare sequenzialmente comandi in modo che l'output del comando che precede coincida con l'input del comando che segue.

- Il concatenamento si indica con il carattere: “|”
- I comandi concatenati in successione devono essere filtri (tranne il primo).
- Il collegamento viene realizzato attraverso file temporanei gestiti dal sistema operativo.
- È una particolare forma di ridirezione.

## Esempio:

```
C:\> DIR | SORT | MORE
```

Fornisce una lista di file e direttori del direttorio corrente su C, in ordine alfabetico e per videate successive.

## Esempi:

```
C:\> DIR A: | FIND ".COM" | SORT | MORE
```

visualizza in ordine alfabetico, fermandosi ad ogni videata, i file di estensione .COM del direttorio corrente sul disco A.

### Alternativa:

usando la sola ridirezione, avremmo dovuto introdurre una serie di file temporanei:

```
C:\> DIR A: > temp.dat
C:\> FIND ".COM" < temp.dat >
temp1.dat
C:\> SORT < temp1.dat > temp2.dat
C:\> MORE < temp2.dat
C:\> DEL temp*.dat
```

☞ Ridirezione e piping possono essere utilizzati insieme:

```
C:\> TYPE elenco.dat | FIND "Rossi" | SORT > lpt1
```

- Stampa le linee del file elenco.dat in cui compare la stringa “Rossi” in ordine alfabetico.

```
C:\> FIND "1993" < iscrizioni.dat |
SORT > iscritti.93
```

- Produce il file iscritti.93 (ordinato alfabeticamente) che contiene le linee del file iscrizioni.dat in cui compare la stringa “1993”.

## File comandi

In MS-DOS è possibile mandare in esecuzione sequenze di comandi memorizzate su file (“BATCH FILE” o **file comandi**).

- I comandi da eseguire vengono memorizzati in sequenza in file di nome:

<prefisso>.BAT

- L’estensione “.BAT” rende noto al sistema operativo che il contenuto del file è da sottoporre all’interprete comandi.

Es.: TC.BAT per chiamare l’ambiente TurboC

```
cls
echo off
cd \TC
TC
cd \
echo on
```

## Invocazione del file comandi

C:\> TC

oppure

C:\> TC.bat

- L'invocazione di un file comandi produce la esecuzione sequenziale di tutti i comandi costituenti, nell'ordine testuale.
- I comandi possono contenere ridirezione e piping.

## File di Configurazione del Sistema

La configurazione standard del sistema può essere modificata mediante i due file:

- AUTOEXEC.BAT
- CONFIG.SYS

### Il file AUTOEXEC.BAT

Il file AUTOEXEC.BAT è un **file comandi** che contiene la sequenza di comandi da eseguire al termine della fase di bootstrap (tipicamente, sono comandi di configurazione del sistema).

### Ad esempio:

AUTOEXEC.BAT

```
echo off
path=c:\dos;c:\TC
append c:\data
prompt $p$g
date
time
print /d:lpt1
keyb It
cls
echo on
```

## Il file CONFIG.SYS

Prima di cercare il file AUTOEXEC.BAT, MS-DOS esegue un gruppo di comandi che caricano i driver installabili e riservano dello spazio nella memoria del sistema per l'elaborazione delle informazioni.

L'elenco dei driver da caricare è contenuto nel file CONFIG.SYS:

- Il file viene letto all'avviamento del sistema: se si modifica questo file, è necessario riavviare il sistema affinché le modifiche abbiano effetto.
- Per modificare la configurazione, è possibile aggiungere nuove direttive a CONFIG.SYS o modificare quelle esistenti.

Il file CONFIG.SYS varie direttive, tra cui:

<b>Device</b>	carica un driver installabile nel sistema operativo.
<b>Buffers</b>	imposta la quantità di memoria che MS-DOS riserva per il trasferimento delle informazioni tra dischi e memoria centrale.
<b>File</b>	Imposta il numero di file che MS-DOS consente di aprire contemporaneamente.

## Esempio:

```
Device = c:/mouse/mouse.sys
Buffers = 30
Files = 30
```

## Comandi di utilità e sviluppo di un programma

Per ottenere un file eseguibile (.EXE o .COM) bisogna passare attraverso le fasi di sviluppo di un programma:

- editing
- compilazione
- linking
- loading (loader, RTS)

Se sono stati rilevati errori:

- **debugging** (si utilizza un debugger per verificare la correttezza del programma).

MS-DOS fornisce programmi di utilità per la prima e le ultime due fasi.

### EDLIN

È un editor di linea (dalla versione 5.0 è disponibile un editor full-screen molto più facile da usare e comodo)

### DEBUG

È un debugger non simbolico.

Il caricamento di file eseguibili è fornito dal run-time system.

### Esercizio:

Supponendo che il disco A contenga la seguente struttura di file:

```
A:\
|__PIPP0: a.bat, b.bat
|__PLUTO: c.bat, d.bat
```

indicare la situazione risultante dopo i seguenti comandi eseguiti in sequenza:

```
C:\> MKDIR A:\PIPP0\PAPERINO
C:\> CHDIR A:\PIPP0\PAPERINO
C:\> COPY A:\PLUTO A:
```

### Soluzione

Il primo comando crea la subdirectory PAPERINO entro il direttorio PIPPO.

Il secondo comando lo rende il direttorio corrente *sull'unità A* (e per questo ciò non è visibile nel prompt, che resta settato sull'unità C, non avendo noi esplicitamente mai cambiato unità).

Il terzo comando copia tutti i file della directory PLUTO nella directory corrente dell'unità A:, che è \PIPP0\PAPERINO.

La situazione finale è la seguente:

```
A:\
|__PIPP0: a.bat, b.bat
|    |__PAPERINO: c.bat d.bat
|__PLUTO: c.bat, d.bat
```

### Esercizio:

Supponendo che il disco C contenga la seguente struttura di file:

```
C:\
|__LUCIA: es1a.txt, es1b.txt es2a.txt es2b.txt
|__LAURA: ex.txt
```

indicare la situazione risultante dopo i seguenti comandi (eseguiti in sequenza):

```
C:\> COPY LUCIA\ES?A.TXT LAURA
C:\> REN LAURA\ES*.TXT LAURA\*.LUC
```

Indicare anche, se esiste, un modo più compatto per ottenere lo stesso risultato.

### Soluzione

Il primo comando copia i due file ES1A.TXT e ES2A.TXT nel direttorio LAURA, il secondo comando li rinomina cambiandone l'estensione in .LUC. Ciò non si applica al file EX.TXT già esistente nel direttorio LAURA, in quanto non compreso nell'insieme di file referenziati dalla sintassi "ES\*.TXT" del comando RENAME.

La situazione finale è la seguente:

```
C:\
|__LUCIA: es1a.txt, es1b.txt es2a.txt es2b.txt
|__LAURA: ex.txt es1a.luc es2a.luc
```

### Comando equivalente:

```
C:\> COPY LUCIA\ES?A.TXT LAURA\*.LUC
```

### Esercizio:

Supponendo che il disco C contenga la seguente struttura di file:

```
C:\
|__LUCIA: win.com
|
|__LAURA: win.exe won.com
```

indicare come dev'essere predisposto il PATH affinché, dando il comando "WIN", venga eseguito il programma situato nel direttorio LUCIA (e non quello, quasi omonimo, del direttorio LAURA), mentre battendo "WON" venga regolarmente trovato ed eseguito il programma situato nel direttorio LAURA.

### Soluzione

È chiaro che, per trovare sia WIN nel direttorio LUCIA sia WON nel direttorio LAURA, entrambi questi direttori dovranno essere elencati nel PATH; tuttavia, la necessità di selezionare, dei due quasi-omonimi, quello del direttorio LUCIA, fa sì che quest'ultimo direttorio debba essere elencato *prima dell'altro* all'interno dell'elenco del comando PATH.

Dunque, potrà essere:

```
C:\> PATH C:\LUCIA;C\LAURA
```