

Il sistema Operativo

È un insieme di programmi che operano direttamente al di sopra della macchina fisica, mascherandone le caratteristiche e fornendo agli utenti un insieme di funzionalità di alto livello.

È un sistema complesso organizzato a strati (**macchine virtuali**).

PROGRAMMI UTENTE

INTERPRETE COMANDI

FILE SYSTEM

GESTORE DELLE PERIFERICHE

GESTORE DELLA MEMORIA

GESTIONE DEI PROCESSI (nucleo)

MACCHINA FISICA

Il Sistema Operativo MS-DOS (Micro-Soft Disk Operating System)

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

Per **personal computer** intendiamo una architettura di computer, costituita come segue:

- una scheda centrale con microprocessore e memoria centrale oltre ad un insieme di porte per la comunicazione con dispositivi esterni;
- un insieme di unità esterne connesse:
 - **video** e **tastiera**; indicate insieme con il termine di **console**;
 - **memoria di massa**, cioè dischi che possono essere:
 - a) Unità a dischetti (floppy disk).
 - b) Unità a dischi rigidi (hard disk). Sono non rimuovibili. Hanno capacità di almeno un ordine o due di grandezza superiori a quelle dei floppy: decine/centinaia di megabyte e tempi di accesso dell'ordine dei msec.
 - Possono essere presenti **dispositivi addizionali**: stampanti, joystick, tavolette grafiche, unità per la comunicazione remota, etc.

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 (pipes).

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" puo' 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" TurboPascal:

C> Turbo

compilatore FORTRAN:

C> FOR1

programma di utente "prog_1.exe"
(prodotto dopo compilazione e linking):

c> prog_1

Word-processors, database, fogli elettronici, etc. sono programmi applicativi.

Ambiente 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**).

In particolare, 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 ed esecuzione del sistema operativo (da disco).

L'attivazione del sistema operativo determina, in sequenza:

- 1 Lettura di 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")

Il processore dei comandi

Comincia ad agire dopo la inizializzazione (o bootstrap) del sistema.

Il command processor costituisce una macchina virtuale basata su:

- un gestore della console;
- un elemento che analizza i comandi da eseguire cioè il "parser" dei comandi dati da console;
- un caricatore (o "loader") dei comandi.

I comandi sono diversificati in **interni** ed **esterni**.

I **comandi interni** fanno parte del sistema operativo caricato inizialmente in memoria centrale (sono contenuti tutti nel file **COMMAND.COM**) e quindi presentano un tempo di latenza minimo: sono interni i comandi più comunemente usati.

Ad esempio: DIR, DEL, COPY, ...

I **comandi esterni** devono essere caricati da disco prima di potere essere messi in esecuzione (corrispondono a file con estensione .COM ed .EXE). La ragione di questa scelta è essenzialmente di non occupare stabilmente memoria con comandi raramente utilizzati.

Ad esempio: PRINT, FORMAT, ...

I **programmi utente** sono considerati dal processore alla stregua di comandi esterni. Corrispondono a file con estensione .EXE, .COM o .BAT.

Schema di funzionamento del processore comandi di MS-DOS

{ *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 directory 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> >

I comandi di MS-DOS possono essere classificati in:

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

Un comando è composto da non più di 3 parti:

1. **Nome**
2. **Parametri**
3. **Opzioni**

Il **Nome**, che identifica il comando, è sempre presente ed occupa il primo posto nella sintassi del comando.

I **Parametri**, se richiesti, definiscono gli oggetti su cui il comando deve agire. Alcuni comandi non richiedono parametri; altri ne richiedono uno o più. I parametri debbono essere separati tra loro e dal *nome* mediante spazi bianchi.

Le **Opzioni** sono composte da una barra (/) seguita da uno o più caratteri, ed hanno lo scopo di alterare la modalità di esecuzione del comando. Sono facoltative. Possono essere espresse una dopo l'altra anche senza separarle con spazi bianchi.

Di norma le *Opzioni* seguono i *Parametri*.

N.B. I comandi possono essere immessi indifferentemente in caratteri minuscoli o maiuscoli.

Concetti base : drive, prompt

Il termine **drive** è una abbreviazione di *disk drive* e significa *unità*.

MS-DOS definisce «**drive corrente**» il drive che verrà considerato in tutte le operazioni facenti riferimento a un *drive* quando non se ne indica esplicitamente un altro.

I *drive* sono identificati da una lettera: tipicamente “A” e “B” per i *drive* che possono ospitare dischetti flessibili (*floppy disk*); “C”, “D”, ecc. per i *drive* che indicano i dischi rigidi (se presenti).

Per cambiare il *drive* corrente basta indicare (come se si trattasse di un comando) il suo identificatore seguito dal simbolo *due punti* (esempio **A:**).

Ad ogni disco magnetico (sia *floppy* che *hard disk*) è associato un nome detto *etichetta di volume*.

In un ambiente interattivo, il **prompt** (*invito*) è il carattere, o il gruppo di caratteri, che il sistema operativo presenta sul video per segnalare che è in attesa di un comando.

In MS-DOS il *prompt* predefinito è costituito dalla lettera del *drive* corrente seguita dal segno “>”; il prompt può essere “personalizzato” da AUTOEXEC.BAT. Ad esempio, se il *drive* corrente è “A” e la *directory* corrente è la *directory* «*radice*», il prompt predefinito e quello modificato sono rispettivamente: **A>** e **A:\>**

La Guida in linea per i comandi (*help*)

Per ricevere direttamente sullo schermo informazioni su un comando, la sua sintassi, i parametri e le opzioni basta immettere da tastiera, subito dopo il *prompt*, il nome del comando stesso seguito dall'opzione */?*, oppure immettere «*help*» seguito dal nome del comando.

Per esempio, volendo informazioni sul comando **FORMAT** si può utilizzare uno dei due comandi:

FORMAT /? oppure **HELP FORMAT**

Il primo comando porta sullo schermo, riga dopo riga, una descrizione concisa delle operazioni svolte dal comando, dei parametri e delle opzioni.

Il secondo invece, che opera a schermo pieno, presenta la prima (o unica) pagina relativa alla sintassi da cui, con semplici operazioni, si può passare alle pagine successive, oppure alle note o agli esempi. La "navigazione" tra una pagina e le successive o le precedenti, come pure tra sintassi, note ed esempi, si effettua semplicemente mediante il mouse o la tastiera.

Volendo un elenco dei comandi disponibili, listati insieme con vari argomenti dell'MS-DOS, usare il semplice comando:

HELP

seguito dall'opzione */H* se si desidera sfruttare al massimo la capienza dello schermo.

Giunti a questo punto, per avere istruzioni su come procedere, premere il tasto **F1**.

Memorizzazione di informazioni: il file

Dal punto di vista dell'utente, il file è l'unità logica di memorizzazione delle informazioni.

Un file è caratterizzato da:

- un nome
- una dimensione (in byte)
- degli attributi
- una data

Il nome di un file è costituito da:

nomefile.estensione

nomefile è una stringa con lunghezza limitata a 8 caratteri;

estensione è una stringa con lunghezza limitata a 3 caratteri. È una sorta di identificazione del tipo di informazioni contenute nel file. Per esempio, un file che contenga un programma sorgente scritto in linguaggio FORTRAN è consigliabile che abbia estensione "FOR".

Il punto fra *nomefile* e *estensione* è richiesto della sintassi.

L'estensione è facoltativa. Quando ci si deve riferire ad un file senza estensione il punto può essere omesso.

Attenzione: la stringa *nomefile.estensione* non deve contenere spazi bianchi.

Nome di un file: caratteri ammessi

Lettere alfabetiche, senza distinzione tra minuscole e maiuscole: a-z A-Z		
Cifre decimali: 0-9		
Caratteri speciali:	@ a commerciale	& e commerciale
\$ dollaro	# diesis	% per cento
- trattino	– sottolineatura	~ tilde
' apice	' accento grave	^ accento circonflesso
! punto esclamativo	() parentesi tonde	{ } parentesi graffe

Estensioni notevoli

I comandi *esterni* corrispondono a *programmi eseguibili* che, a seconda delle caratteristiche, risiedono in files con *estensione* COM, EXE o BAT:

COM il contenuto del file verrà caricato in memoria per la esecuzione esattamente com'è (file *immagine*)

EXE il contenuto del file, al momento del caricamento in memoria per la esecuzione, deve subire un processo di finitura in funzione dell'indirizzo di memoria scelto.

BAT non è propriamente un comando, bensì un file di testo in cui ogni linea è un comando che si vuole eseguito.

Il Nome di un comando esterno è il Nome del file che ne contiene il programma eseguibile, privato della estensione.

TXT file di testo
DOC file documento

Esempi di nomi di files

COMMAND.COM: file che contiene il codice del processore dei comandi MS-DOS.

MIOPROG.EXE: esempio di file eseguibile

MIOPROG.EXE

```
ß\_\^¥†©÷ÓµÆ@å¢‰|•‡f±¥†©÷™åÆ@µ¢‰|•‡f¥^¥®
ö*œ_µ¢_ß¬_^¥®•‡‰™å§ª<¿µ¢‰|•‡f±¤†£÷™å
.....
```

PROVA.PAS: file che contiene un programma di prova scritto in linguaggio Pascal (“PROVA” è la parte *nomefile* e “PAS” è la *estensione*).

BUBBLE.FOR: file contenente un programma scritto in linguaggio Fortran (“BUBBLE” è la parte *nomefile* e “FOR” è la *estensione*).

LETTERA.TXT: file contenente un testo ASCII

lettera.txt

```
Spettabile Ditta,
dal vostro catalogo datato Ottobre 1994
```

STIPENDI.DAT: file di dati

stipendi.dat

AVANTI LUIGI	1/2/93 L. 2.145.000
BOSSI PIERO	6/2/93 L. 2.000.000

Nomi di file: caratteri “jolly” (wildcards)

Alcuni caratteri non possono essere usati per costruire il nome di un file. Per esempio, nessuno dei seguenti caratteri, disponibili sulla tastiera, può essere usato:

\ | / [] < > : ; . , + = " * ?

Tuttavia i due ultimi caratteri della riga precedente, ossia * e ?, pur non potendo far parte del “vero” nome di un file, possono a volte essere usati nei comandi come caratteri “jolly” per specificare gruppi di file:

- * rappresenta una qualsiasi sequenza (eventualmente nulla) di caratteri permessi.
- ? rappresenta un qualsiasi carattere singolo.

Esempi:

PAOL?.DOC	identifica tutti i file il cui nome è costituito dalle 4 lettere PAOL seguite da un solo qualunque carattere e dalla estensione DOC
PAOL*.DOC	identifica tutti i file il cui nome è costituito dalle 4 lettere PAOL seguite da non più di 4 caratteri qualunque e dalla estensione DOC
*.COM	identifica tutti i files con estensione COM (comandi): format.com, backup.com, etc.
.	identifica QUALSIASI FILE !!!!!

Uso dei caratteri jolly

Solo il punto può essere indicato dopo un asterisco; altri caratteri **non** vengono presi in considerazione.

Per esempio, la stringa

`*A.DOC`

è equivalente alla stringa

`*.DOC`

e **non** individua (come può sembrare) tutti i files con estensione DOC il cui nome termina con A, ma tutti i files con estensione DOC.

Un altro uso dei caratteri jolly riguarda l'abbinamento di files o gruppi di files. Si riportano due esempi di questo uso con il comando REN (usato per cambiare nome e/o estensione ai files):

`REN *.DAT *.OLD`

`REN Z*.* K*??$`

Nel primo caso tutti i files che hanno *estensione* DAT la cambiano in OLD, conservando intatto il *nomefile*.

Nel secondo caso tutti i files il cui *nomefile* inizia con Z cambiano la iniziale in K, e cambiano il terzo carattere della *estensione* (qualunque essa sia) in \$.

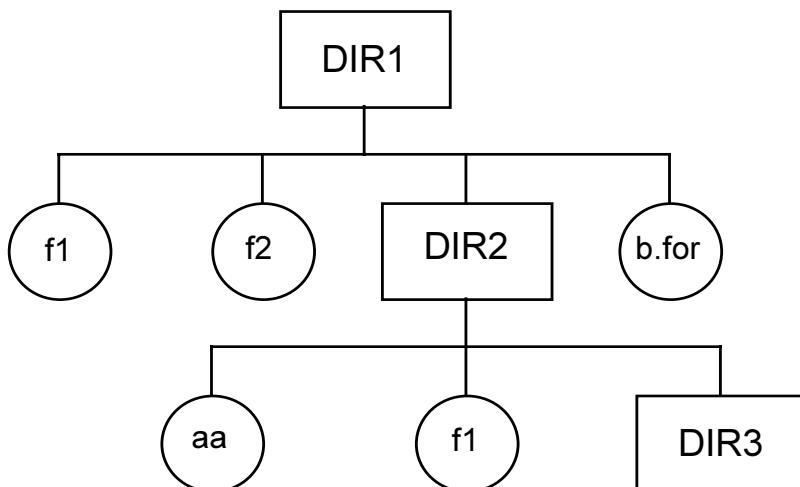
Attenzione: non tutti i comandi sono abilitati a riconoscere i caratteri jolly.

Il File system

È l'organizzazione dei file su disco, come vengono strutturati e memorizzati.

Ogni disco contiene almeno una *directory*, ossia un «indirizzario» in cui registrare nome, posizione e altre informazioni di ogni file residente sul disco stesso.

La *directory* principale (*directory* radice o *root*) viene creato al momento della *formattazione* del disco. Altre *directory* possono essere create dagli utenti secondo una struttura gerarchica: ogni *directory* «figlio» viene registrato nella *directory* «padre» come se si trattasse di un file speciale.

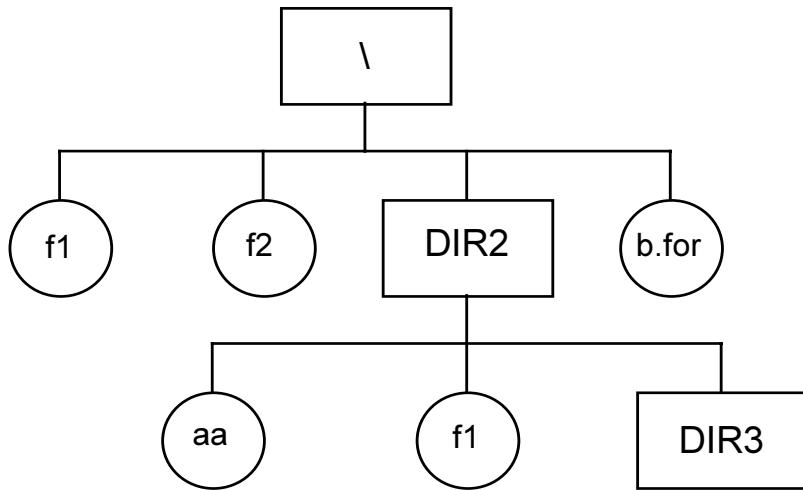


Struttura gerarchica:

nodi	=	<i>directory</i>
foglie	=	files
rami	=	appartenenza

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

La *directory radice* (\) è l'apice della struttura gerarchica:



Identificazione di files e *directory*:

Ogni file o *directory* è identificato da un percorso (*path*) seguito dal suo nome.

Il percorso è **assoluto** se è riferito alla radice: è costituito dal simbolo della radice (\) seguito dai nomi dei nodi che si incontrano per passare dalla radice alla foglia voluta (file o *directory*) separati tra loro da barre rovesciate.

Il percorso è **relativo** se è riferito al *directory* “corrente”: è costituito dai nomi dei nodi che si incontrano per passare dal *directory* corrente alla foglia voluta (file o *directory*) separati tra loro da barre rovesciate: il nome del *directory* corrente non deve essere indicato.

Per indicare su quale *drive* si trova un file, se ne premette l'identificatore, seguito dal carattere “due punti”:

[drive:] [path] nomefile.estensione

Usiamo le parentesi quadre solo per indicare elementi facoltativi: **esse non vanno per nessun motivo riportate nei parametri o nei comandi.**

La forma generale con cui si fa riferimento ad un file è dunque:

[drive:] [path] nomefile.estensione

ove:

drive È la lettera che individua il *drive*; è facoltativo se il *drive* è quello corrente.

path È il percorso sull’albero dei *directory* per raggiungere il file: a partire dalla radice, radice inclusa (*path* assoluto) o a partire dal *directory* corrente, questo escluso (*path* relativo). Il *path* è obbligatorio, ma è libera la scelta tra assoluto e relativo. Se il file risiede nel *directory* corrente il suo *path* relativo è nullo.

File (o *directory*) distinti possono avere nomi o percorsi relativi uguali, ma non nomi o percorsi assoluti uguali (file **f1** nella figura seguente).

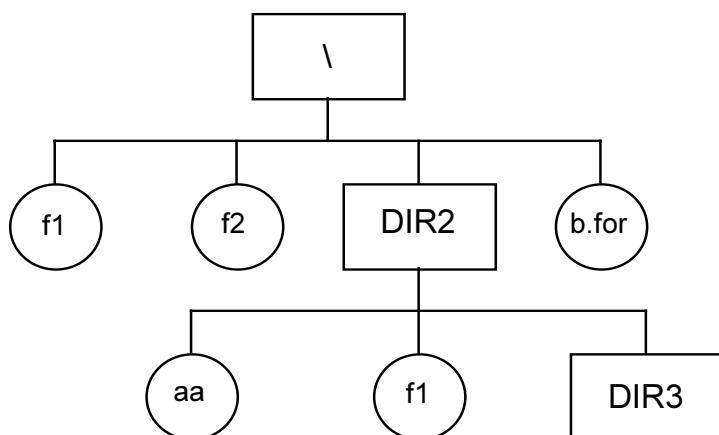
N.B. se i nomi o i percorsi relativi sono uguali si tratta evidentemente di nomi o percorsi relativi a *drive* o *directory* correnti diversi.

Notazioni convenzionali:

- la *directory* corrente è identificata dal punto (.)
- la *directory* padre di quella corrente è identificata dal doppio punto (..)
- la *directory* radice è indicata con una barra rovesciata (\)

Esempio

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



Il suo nome dato in forma assoluta è:

A:\DIR2\aa

Nel caso in cui A:\DIR2 sia la *directory corrente*, il nome in forma relativa è:

aa

Nel caso invece in cui la *directory corrente* sia A:\DIR2\DIR3, il nome in forma relativa è:

..\\aa

Comandi per l'esplorazione del file system

Per cambiare drive scrivere la lettera che identifica il drive seguita dai due punti (esempio A:).

DIR [drive:][path] [opzioni] visualizza il contenuto di una *directory* (quella corrente del *drive* corrente se non se ne specifica una differente). DIR accetta i caratteri jolly.

Tra le opzioni: **/P** la lista delle informazioni è fornita una schermata alla volta

/W vengono visualizzati solo i nomi dei file e delle *directory*, su 5 colonne, senza altre informazioni

CHDIR [drive:][path] cambia *directory* corrente. Il comando può essere abbreviato in CD. Senza parametri, visualizza la *directory* corrente.

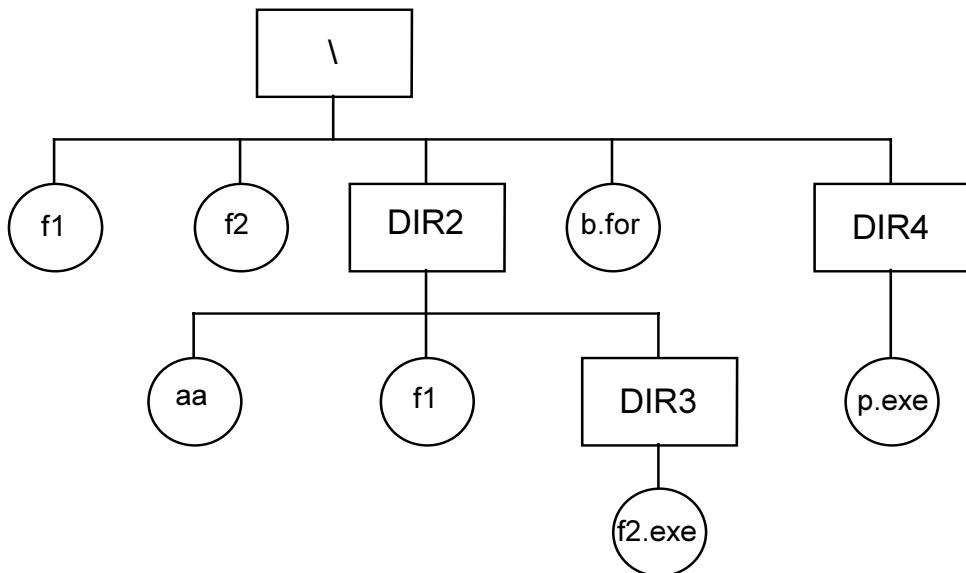
MKDIR [drive:][path] crea una nuova *directory*. Il comando può essere abbreviato in MD.

RMDIR [drive:][path] cancella una *directory*, solo se è vuota. Il comando può essere abbreviato in RD.

DELTREE [drive:][path] cancella una *directory*, compresi tutti files e le sotto-*directory* che contiene.

TREE [drive:][path] consente di esaminare la struttura gerarchica dei *directory*.

Esempio: File system del disco A



C:\>a:

A:\>cd dir2

A:\DIR2>cd dir3

A:\DIR2\DIR3>dir

Il volume nell'unità A è FONDAMENTI

Il numero di serie del volume è 427B-1482

Directory di A:\DIR2\DIR3

.	<DIR>	03/12/93	19.36
..	<DIR>	03/12/93	19.36
F2	EXE	2207 24/09/91	2.00
		3 file	2.207 byte
			1.410.048 byte disponibili

questa serie di comandi equivale a:

C:\>dir a:\dir2\dir3

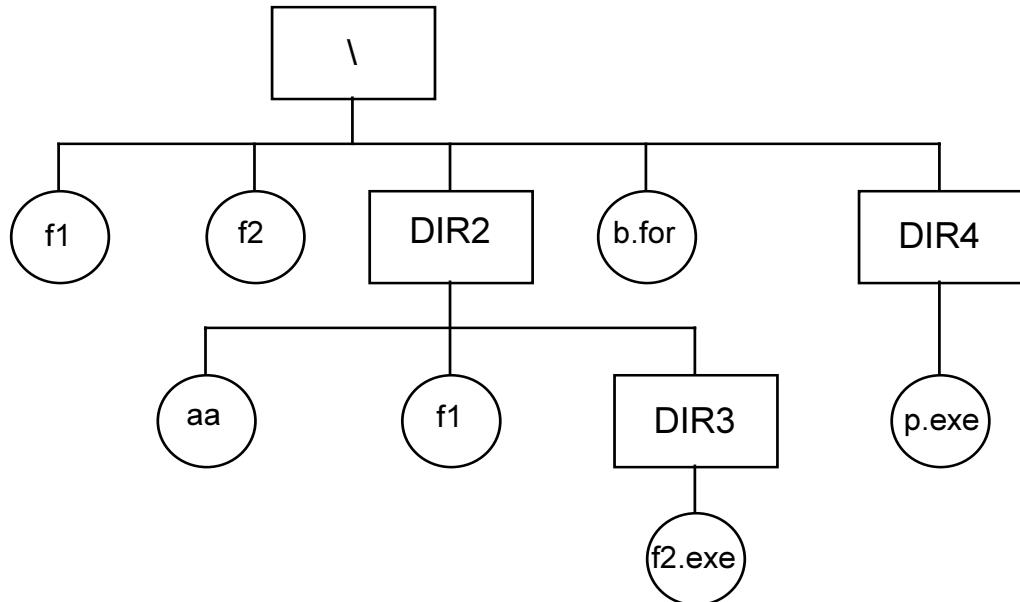
oppure a:

C:\>cd a:\dir2\dir3

C:\>dir a:

Esempio: creazione di una directory

Situazione iniziale:



Se la directory corrente è la directory radice:

A : \ >mkdir dir4\dir5

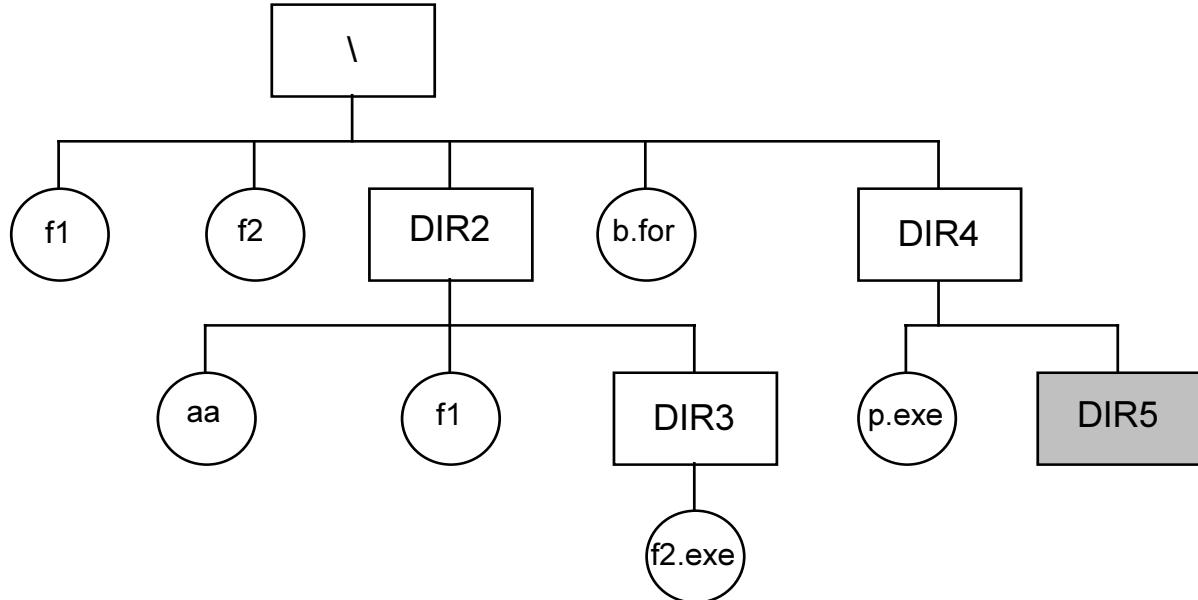
Se la directory corrente è DIR3:

A : \DIR2\DIR3 >mkdir ... \.. \dir4\dir5

oppure, indipendentemente dalla directory corrente:

A : ?? >mkdir \dir4\dir5

Situazione finale:



Comandi di gestione dei file

TYPE [drive:][path]nomefile

visualizza sullo schermo il contenuto di un file.

```
A:\>type dir2\f1
```

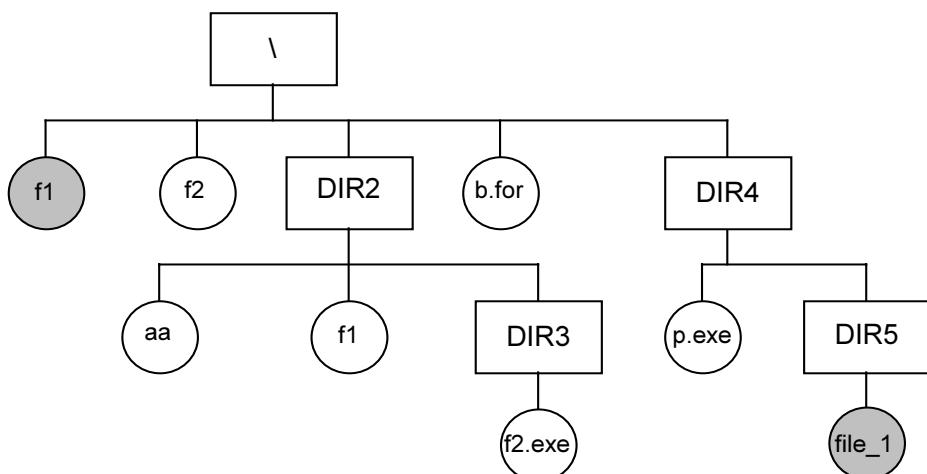
COPY origine [destinazione]

consente di copiare uno o più file in altra posizione. *Origine* può rappresentare un gruppo di file mediante caratteri jolly o indicando (anche implicitamente) un *directory*.

Se manca la *destinazione*, s'intende trattarsi di file con nomi ed estensioni uguali a quelli di origine e residenza nel *directory* corrente; in questo caso i file di origine **non** devono risiedere nel *directory* corrente.

Esempio 1

```
A:\DIR2>copy ..\f1 \dir4\dir5\file_1
```



Esempio 2

```
A:\DIR2>copy \*.* c:\
```

oppure, più semplicemente:

```
A:\DIR2>copy \ c:\
```

Copia i files **f1**, **f2** e **b.for** nella radice del disco **C**.

Il comando **COPY** permette anche di copiare più file in uno singolo (concatenazione di file):

COPY file1+file2[+...][file-dest]

A: \>copy f1+f2+b.for tutto.dat

Se il parametro *file-dest* è assente il file di destinazione è il primo file, che viene quindi modificato:

A: \>copy f1+f2+b.for

XCOPY path-sorg path-dest [opz]

copia file e *directory* secondo le modalità espresse dalle opzioni.

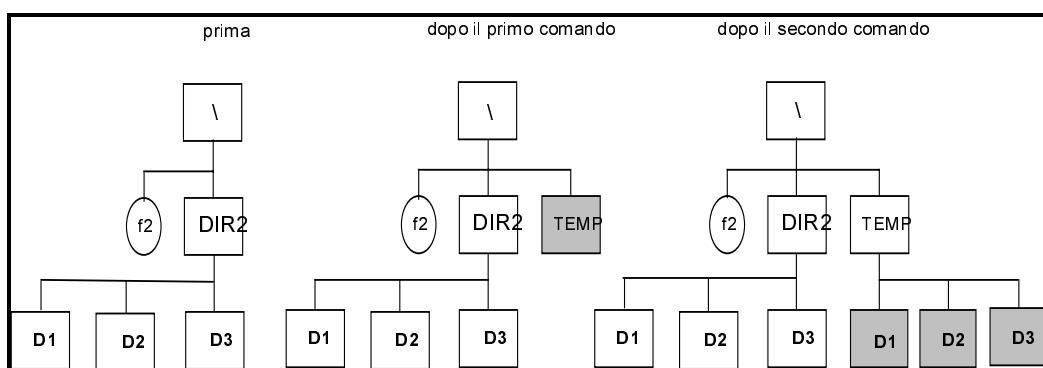
Possibili opzioni:

/S	file + <i>directory</i> non vuoti
/E	file + <i>directory</i> (da usare con /S)
/D:gg/mm/aa specificata.	copia i file modificati dopo la data

Esempio

A: \>mkdir \temp

A: \>xcopy \dir2 \temp /s /e



Il comando **MOVE** ha due diverse funzioni:

1. **Muove da una *directory* ad un'altra uno o più file;**

Il formato è:

MOVE [drv1:][pth1]nf1[,drv2:][pth2]nf2[...]] [drv:][pth][nf]

ove il nome del file di destinazione (*nf*) può essere indicato solamente quando il file da trasportare è uno solo e gli si vuole anche cambiare nome. Nel caso di trasporto di più file la destinazione va indicata solamente con *drive* e *path* (se il path è quello corrente lo si può indicare con un punto)

2. **Cambia nome ad una *directory*.**

Il formato è:

MOVE [drv1:][pth1] [drv1:][pth2]

ove i due percorsi *pth1* e *pth2* debbono differire solo nell'ultimo elemento, che consiste nel nome vecchio e nuovo del *directory*.

Esempi

Trasporto di due file sul drive C, *directory* UTI (*sotto-directory* della radice): uno dal *drive* A, l'altro dal C stesso.

C: \>**MOVE A: \ZED\AHA.K, DOS\POI.COM UTI**

Trasporto di un file con cambio di nome:

C: \>**MOVE MIO.TXT A: \VARIA\SUO.TXT**

Cambio del nome di un *directory*:

C: \>**MOVE LAV\ATT\OGGI LAV\ATT\IERI**

Per cambiare nome ad un file si usa il comando **REN** (o **RENAME**):

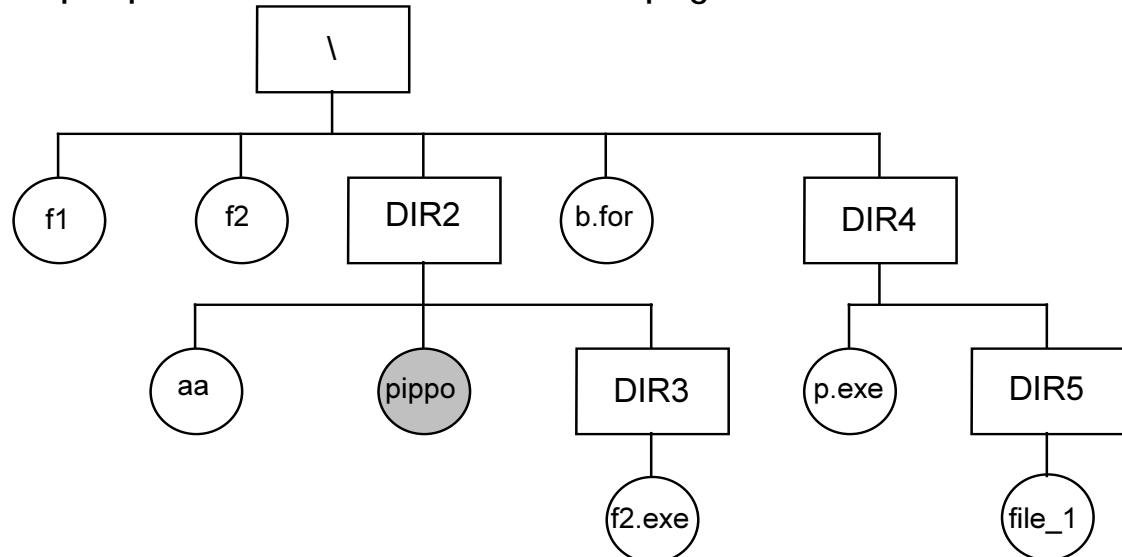
REN [drive:][path]nomefile1 nomefile2

Usando i caratteri jolly, si può cambiare nome anche a gruppi di files
I caratteri jolly possono essere usati sia in *nomefile1* che in *nomefile2*. Quelli usati in *nomefile2* rappresentano caratteri identici a quelli di *nomefile1*.

Esempi

A:\>ren a:\dir2\f1 pippo

Dopo questo comando lo schema di pag. 23 diviene:



C:\CONT\CORR>ren *.txt *.scr

Tutti i file del *directory* corrente che hanno estensione TXT la cambiano in SCR.

D:\ALT>ren ESAM*.K* LZ*.?P?

Tutti i file della *directory* corrente il cui nome inizia con ESAM e la cui estensione inizia con K avranno le prime due lettere del nome cambiate in LZ e la lettera intermedia della estensione cambiata in P.

DEL o **ERASE** cancella un file o un gruppo di file:

DEL [drive:][path]nomefile

Per eliminare un gruppo di file si usano i caratteri jolly.

Esempi

C:\>del *.bak

elimina dalla *directory* corrente i file con estensione BAK.

C:\>del *.*

elimina dalla *directory* corrente tutti i file, ma non elimina le *sotto-directory*. **Da utilizzare con cautela!**

FC (File Compare) confronta tra loro due files:

FC [drive1:][path1]fn1 [drive2:][path2]fn2

Le differenze vengono presentate sullo schermo. Varie *opzioni* consentono di “guidare” il confronto, considerando o no uguali caratteri alfabetici maiuscoli e minuscoli, limitando la dimensione dei messaggi emessi sullo schermo, ecc.

I caratteri jolly sono interpretati come nel comando REN.

Esempi

C:\LETTE\>fc /C ASH1.FOR ASH2.FOR

I due files vengono confrontati ignorando le differenze dovute a caratteri maiuscoli e minuscoli.

C:\CNTB\>fc SUB*.FOR \SAF*.*.FOL

Ogni file nel *directory* corrente che ha estensione FOR e nome che inizia con SUB viene confrontato con il file che ha stesso nome ed estensione FOL nel *directory*\SAF.

La stampa del contenuto di un file si può ottenere con il comando **PRINT**:

PRINT [opzioni] [drive][path]nfile [...] [/C] [/P]

Per richiedere la stampa di un gruppo di files possono essere utilizzati i caratteri jolly.

La stampa avviene in *background*: ciò significa che, mentre stampa, l'MS-DOS accetta ed esegue altri comandi.

Il comando PRINT imposta una "coda" di stampa, a cui è possibile aggiungere altri elementi con successivi comandi PRINT.

Opzioni:

/D:perif	specifica la Stampante ¹
/P	aggiunge file alla coda di stampa
/C	toglie file dalla coda di stampa
/T	termina la stampa (svuota la coda)

N.B. per l'uso di /P e /C è bene consultare la Guida in linea.

Esempio

```
C:\>print autoexec.*  
La stampa di C:\AUTOEXEC.BAT è in corso  
C:\AUTOEXEC.OLD è in coda  
C:\>print autoexec.old /c
```

Se non vengono specificati nomi di file, il comando PRINT mostra la situazione della coda di stampa.

¹**perif** può essere: PRN (valore standard), LPTn o COMn

Gli «attributi» dei file

A ciascun file sono associati 4 attributi:

- (a) Archivio²
- (r) Sola lettura
- (h) File nascosto
- (s) File di sistema

ATTRIB [opzioni] [[drive][path]file]

modifica o imposta le modalità di accesso ai file.

Opzioni:

- +A|**A** attiva o disattiva l'attributo di **archivio** sul file specificato
- +R|**R** attiva o disattiva l'attributo di **sola lettura** sul file specificato
- +H|**H** attiva o disattiva l'attributo di **file nascosto** sul file specificato
- +S|**S** attiva o disattiva l'attributo di **file di sistema** sul file specificato

Senza opzioni, visualizza gli attributi del file specificato.

Senza opzioni nè parametri visualizza gli attributi di tutti i file del *directory* corrente.

² Usato per contrassegnare i file che hanno subito modifiche dopo l'ultimo *backup*. È utilizzato dai comandi XCOPY e MSBACKUP (e da BACKUP e RESTORE, nella versione 5 di MS DOS).

Dispositivi come file

Ad alcuni dispositivi sono associati nomi particolari, per esempio CON alla console, PRN alla stampante. Usando questi nomi, sui dispositivi corrispondenti sono ammesse operazioni analoghe a quelle su file (dispositivi trattati come file).

Ad esempio, è possibile copiare il contenuto di un file sul video:

```
copy file.dat con
```

o copiare in un file le informazioni battute alla tastiera:

```
copy con file.dat
```

in questo caso la fine dell'input viene definita mediante la combinazione dei tasti **CTRL-Z** (editor molto primitivo); o copiare un file sulla stampante:

```
copy file.dat prn
```

I nomi particolari, predefiniti, sono riservati, e **non** possono essere assegnati a file, anche se accompagnati da una estensione. Essi sono i seguenti:

CLOCK\$	COM1	LPT1
CON	COM2	LPT2
AUX	COM3	LPT3
NUL	COM4	PRN

Comandi di gestione dei dischi

Formattazione dei dischi

Tutti i dischi nuovi devono essere formattati prima dell'uso, per creare la directory principale e una tabella di assegnazione dei file per il disco.

(Attenzione: la formattazione cancella tutte le informazioni presenti sul disco).

FORMAT *drive*:

Esempio: si desidera formattare un dischetto inserito nel drive A (*si noti il «prompt» prima del comando*):

C:\> **FORMAT A:**

Opzioni del FORMAT

FORMAT *drive*: /S

crea un “disco di sistema”, copiandovi alcuni file fondamentali, tra i quali COMMAND.COM, che contiene il processore dei comandi.

FORMAT *drive*: /V:*label*

attribuisce direttamente un nome (*label*) al disco; senza questa opzione il nome viene richiesto alla fine della operazione.

FORMAT *drive*: /F:*num*

“*num*” specifica la capacità del disco da formattare (ad esempio: 360, 720, 1440, ...).

Dalla versione 5.0 esiste il comando:

UNFORMAT *drive*:

che consente il recupero dei dati da un’unità appena formattata. Va eseguito prima di qualunque operazione di scrittura.

Il comando FORMAT con l’opzione /U:

FORMAT *drive*: /U

rende inefficace il successivo uso di unformat.

Comandi di gestione dei dischi

DISKCOPY [*drive1:*] [*drive2:*]

Copia il contenuto di un dischetto su un altro dischetto dello stesso tipo. Il precedente contenuto del dischetto di destinazione viene perduto.

Se si specifica lo stesso *drive* per l'origine e la destinazione, viene usato il disco rigido come supporto temporaneo dei dati da copiare, e l'utente è invitato a scambiare una o più volte i dischetti nell'unico *drive*.

Se il dischetto in *drive2* non è "formattato", esso viene "formattato" dallo stesso DISKCOPY prima della copia.

Esempio: **diskcopy a: b:**

DISKCOMP [*drive1:*] [*drive2:*]

confronta il contenuto di due dischetti dello stesso tipo.

Esempio: **diskcomp a: b:**

SCANDISK

Esamina la superficie del disco o dischetto specificato e ne rileva eventuali difetti o errori, aprendo sullo schermo una finestra di dialogo per fare scegliere all'utente l'azione da intraprendere per superare il problema.

Nuovo con la versione 6.2 di MS DOS, sostituisce il precedente comando CHKDSK, peraltro ancora disponibile.

Comandi di configurazione del sistema

PATH [[*drive:*]*path*[;....]]

imposta uno o più percorsi per la ricerca dei file *eseguibili*, ossia:

<i>file con estensione</i> COM	=	COMANDI
<i>file con estensione</i> EXE	=	PROGRAMMI
<i>file con estensione</i> BAT	=	FILE di COMANDI

Finché non viene impostato un percorso di ricerca, i file eseguibili vengono cercati solo nella *directory* corrente.

Dal momento in cui uno o più percorsi sono impostati, i file eseguibili vengono cercati prima nella *directory* corrente, poi nelle *directory* specificate nei "percorsi di ricerca".

Se nella stessa *directory* sono presenti file eseguibili con lo stesso nome, ma con estensione diversa, la ricerca avviene nell'ordine: COM, EXE, BAT.

Il comando PATH senza alcun parametro visualizza la impostazione corrente (percorsi di ricerca in uso).

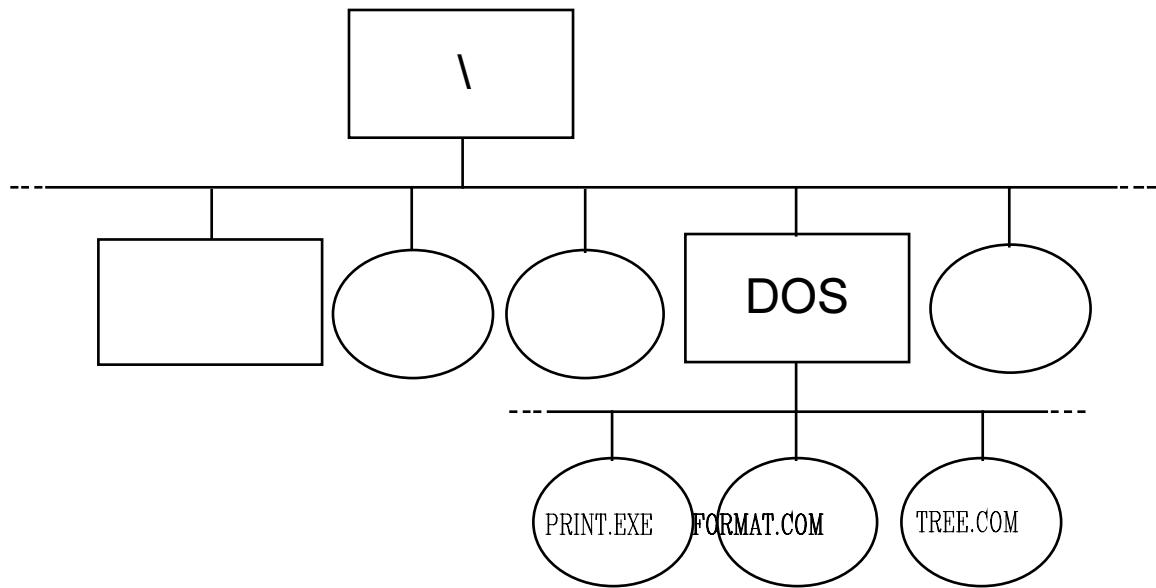
Successivi richiami di PATH (con parametri) determinano la perdita dei percorsi precedentemente impostati.

Se, come parametro, è presente solo un "*punto e virgola*", vengono eliminati tutti i percorsi precedentemente impostati:

PATH ;

N.B. I parametri debbono essere separati tra loro solo dal *punto e virgola*, senza alcuno spazio bianco.

Esempio 1 di PATH



C:\>**path**

Nessun percorso

C:\>**format a:**

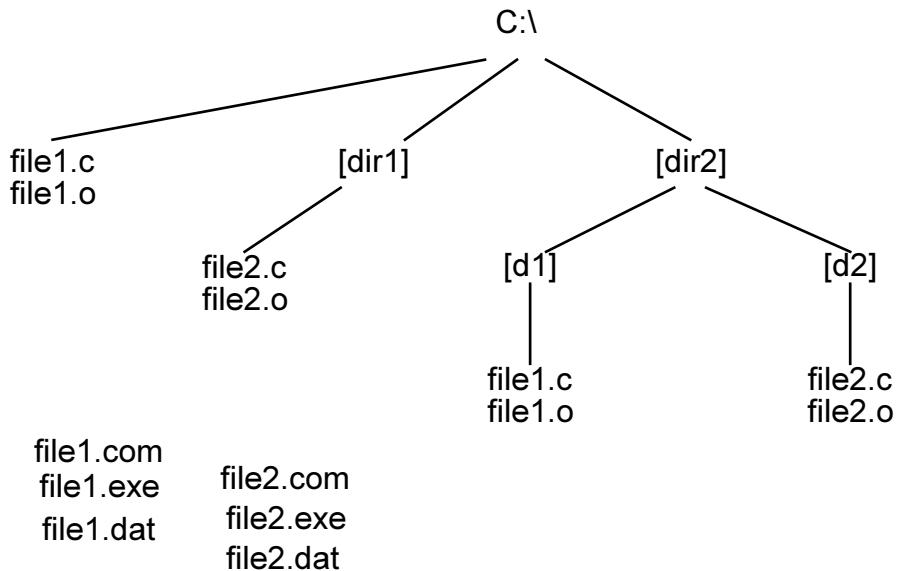
Comando o nome di file errato

C:\>**path c:\dos** *(impostazione del percorso di ricerca)*

C:\>**format a:**

Inserire un nuovo disco ...

Esempio 2 di PATH



Facendo riferimento all'albero gerarchico tracciato qui sopra, il comando:

```
PATH c:\;c:\dir2;c:\dir2\d2
```

fa ricercare i file eseguibili, oltre che nella *directory* corrente, nell'ordine, nelle *directory*:

- **c:** (radice)
- **c:\dir2**
- **c:\dir2\d2**

Attenzione: i percorsi di ricerca debbono essere espressi in forma assoluta, in modo da restare fissi al variare della *directory* corrente.

APPEND [[*drive:】path[;....]]*

Imposta uno o più percorsi di ricerca per i file di dati.

Finché non viene impostato un percorso di ricerca, i file di dati (es. .doc e .dat) vengono cercati solo nella *directory* corrente.

Dal momento in cui uno o più percorsi sono impostati, i file di dati vengono cercati prima nella *directory* corrente, poi nelle *directory* specificate dai percorsi di ricerca (*directory* "aggiunte").

Se non viene fornito nessun argomento, visualizza la impostazione corrente (percorsi di ricerca in uso).

Successivi richiami di "APPEND" (con parametri) determinano la perdita dei percorsi precedentemente impostati.

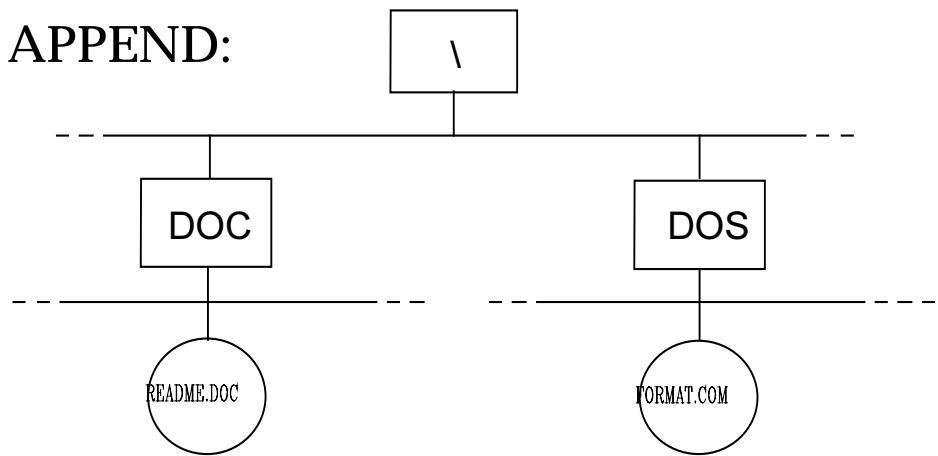
Se, come argomento, è presente solo un "*punto e virgola*", vengono eliminati tutti i percorsi precedentemente impostati:

APPEND ;

Se non viene impostato alcun percorso preferenziale, il percorso di ricerca è quello standard, ossia il "*directory corrente*".

N.B. I parametri debbono essere separati tra loro solamente dal *punto e virgola*, senza alcuno spazio bianco.

Esempio di APPEND:



C:\>**append** (*visualizza i directory "aggiunti"*)
APPEND=C:\DOS

C:\>**type readme.doc**
File non trovato - README.DOC

C:\>**append c:\dos;c:\doc** (*imposta nuovi percorsi*)

C:\>**type readme.doc**
... (*visualizza il file readme.doc*)

PROMPT [*testo*]

Modifica l'aspetto del *prompt* dei comandi.

Senza parametro ripristina il *prompt* predefinito, costituito dalla lettera del *drive* corrente seguita dal segno ">".

Il parametro *testo* è formato dalla stringa di caratteri (che può comprendere anche spazi bianchi) che si vuole che appaia come *prompt*.

Nella stringa non possono comparire alcuni caratteri speciali, che all'interprete dei comandi parrebbero ambigui. Per includerli nel *prompt* bisogna codificarli in modo particolare.

Questa codifica permette di inserire nel *prompt* anche informazioni destinate a variare durante la sessione, come la data, l'ora, il *drive* e il percorso correnti.

Ognuno di questi elementi deve essere rappresentato nel parametro *testo* da una coppia di caratteri, di cui il primo è sempre il dollaro (\$).

Ad esempio, la forma usuale del *prompt*, che comprende il *drive* e il percorso correnti seguiti dal segno ">" si ottiene con il comando:

PROMPT \$P\$G

Infatti **\$P** è la codifica del *drive* e percorso correnti, e **\$G** è quella del segno ">".

ECHO

È utilizzato quasi esclusivamente nei *file di comandi* (file con estensione BAT). Consente di attivare (ECHO ON) o disattivare (ECHO OFF) la visualizzazione dei comandi man mano che vengono eseguiti.

Accompagnato, invece che da ON/OFF, da un testo qualsiasi, lo fa comparire sullo schermo.

Direttamente da tastiera è di scarsa utilità se accompagnato da un messaggio; accompagnato da OFF o ON permette di fare scomparire o riapparire il *prompt*.

CLS

Cancella ogni scritta dallo schermo, e pone il cursore nella posizione in alto a sinistra (posizione *home*) ove compare il *prompt* se non è stato disabilitato dal comando ECHO OFF.

DATE e TIME

Consentono la visualizzazione e/o la modifica della data e dell'ora impostate nel sistema.

Esempi:

- **C:\>date**
La data corrente è Lun 19/12/1994
Digitare la nuova data (gg/mm/aa):
(premere solo "Invio" se non si vuole modificare la data)

- **C:\>date 9/12/93** *(impostazione di nuova data)*

- **C:\>time**
L'ora corrente è 10.51.45,96
Digitare la nuova ora:
(premere solo "Invio" se non si vuole modificare l'ora)

- **C:\>time 17.15** *(impostazione di nuova ora)*

Variabili di ambiente

In MS-DOS è possibile dichiarare ed utilizzare variabili (*variabili "di ambiente" o "di sistema"*).

Il comando MS-DOS per dichiarare, impostare o visualizzare variabili di ambiente è:

SET [variabile = valore]

Senza parametri visualizza i valori correnti di tutte le variabili dichiarate fino a quel momento.

Se manca *valore* la *variabile* viene cancellata.

SET offre una soluzione alternativa ai comandi PATH e PROMPT, in quanto questi comandi non fanno che impostare le due variabili:

PATH e PROMPT

Le variabili di ambiente possono essere interrogate solo all'interno di file eseguibili:

COMANDI	<i>file con estensione COM</i>
FILE di COMANDI	<i>file con estensione BAT</i>
PROGRAMMI	<i>file con estensione EXE</i>

Esempio

C:\>**set path=c:\dos** (path c:\dos)

C:\>**set prompt=\$p\$g** (prompt \$p\$g)

C:\>**set miadir=a:\dir2**

C:\>**set**
PATH=c:\dos
PROMPT=\$p\$g
MIODIR=a:\dir2

Ridirezione

In generale, i comandi del DOS presentano dati sullo schermo (uscita) e ne richiedono dalla tastiera (ingresso).

In effetti schermo e tastiera sono dispositivi particolari per il DOS:

tastiera	=	dispositivo standard di ingresso
schermo	=	dispositivo standard di uscita
schermo	=	dispositivo standard per messaggi d'errore

È possibile fare in modo che l'uscita di un comando (ma non i messaggi d'errore) non vada allo schermo, ma sia *ridiretto* verso un file o altro dispositivo.

È anche possibile *ridirigere* l'ingresso e fare in modo che non venga richiesto alla tastiera, ma ad un file o altro dispositivo.

I due simboli “<” e “>” sono usati per indicare rispettivamente la *ridirezione* in ingresso e quella in uscita.

Il simbolo “>>” consente di *ridirigere* l'uscita ad un file preesistente, senza alterarne il precedente contenuto, ma accodando le nuove informazioni a quelle già esistenti.

Ciascuno dei simboli visti si usa ponendolo alla destra di un comando (completo di parametri e di opzioni) e facendolo seguire dal nome di un file o di un dispositivo.

La *ridirezione* si può usare anche nel richiamo di programmi applicativi.

Esempi di ridirezione in uscita

```
C:\>dir b: > file.dat
```

Il listato della *directory* del disco **B:** è ridiretto direttamente nel file **FILE.DAT**. Se questo file non esiste già, viene creato automaticamente. Se il file esiste, viene sovrascritto.

```
C:\>type pippo.dat > lpt1
```

l'uscita del comando **TYPE PIPPO.DAT** (il contenuto del file **PIPPO.DAT**) è ridiretta verso la stampante.

Equivale ad un comando di stampa in *foreground* (l'utente non potrà dare altri comandi fino al completamento della stampa).

```
C:\>type varia.dat >> \barra\tempo.inf
```

copia il file **VARIA.DAT** nel file **TEMPO.INF** (che risiede nella *directory* \barra) lasciandone intatto il contenuto ed aggiungendo le nuove informazioni in coda a quelle preesistenti. Se **TEMPO.INF** non esiste già, esso viene creato, ed il simbolo **>>** assume la stessa funzione di **>**.

```
C:\>fc file1.pas file2.pas >> comp.dat
```

```
C:\>fc file3.pas file4.pas >> comp.dat
```

nel file **COMP.DAT** sono via via aggiunte le uscite dei due comandi di confronto. Se **COMP.DAT** non esiste prima della esecuzione del primo comando, esso viene creato, ed il simbolo **>>** assume, solo nel primo comando, la stessa funzione di **>**.

Ridirezione in ingresso: i comandi filtro

La *ridirezione* in ingresso fa in modo che l'ingresso al comando venga preso non da tastiera ma da un file.

E' usata in pratica solo con i comandi *filtro*.

I comandi *filtro* sono tre: **SORT**, **FIND** e **MORE**, e sono così chiamati perché agiscono da *filtro* tra il dispositivo di ingresso e quello di uscita.

In assenza di segni di *ridirezione*, i comandi *filtro* richiedono l'ingresso da tastiera e mandano l'output al video. Tuttavia questo uso è di scarsa utilità, e i comandi filtro sono solitamente utilizzati con la *ridirezione* (o il *piping*).

SORT [/r] [/+n] < *file.inp*

Ordina alfabeticamente (secondo la sequenza dei caratteri ASCII) e visualizza sullo schermo le linee del file *file.inp* ricevuto in ingresso. Si noti che SORT non distingue le lettere minuscole dalle maiuscole.

La opzione */r* inverte la modalità di ordinamento (da Z ad A e da 9 a 0)

La opzione */+n* esegue l'ordinamento in base ai caratteri presenti nella posizione *n*-esima di ogni linea, e nelle posizioni alla destra di quella. Se questa opzione è assente, viene considerato *n* = 1.

Esempio:

Disponi in ordine alfabetico il file **lista.rec** e ridirigilo nel file **lista.ord**:

```
C:\>sort < lista.rec > lista.ord
```

FIND [/v] [/c] [/n] [/i] "stringa" < file.inp

Ricerca **stringa** (la stringa di caratteri espressa tra virgolette) nel file **file.inp**; se la stringa è presente vengono mostrate sul video tutte le linee che la contengono.

Si noti che FIND, a differenza di SORT, distingue le lettere minuscole dalle maiuscole.

Le opzioni, da inserire tra il comando e il parametro, consentono variazioni alla funzione di FIND:

- /v** Mostra le righe che **non** contengono la stringa
- /c** Mostra solo il **numero** di righe che contengono la stringa
- /n** Mostra, davanti ad ogni riga, il suo numero progressivo
- /i** Considera alla stessa stregua minuscole e maiuscole

Esempio:

Ricerca la parola **path** (in caratteri minuscoli o maiuscoli) all'interno del file **AUTOEXEC.BAT**:

```
C:\>find /i "path" < autoexec.bat
PATH=C:\DOS
```

MORE < file.inp

Presenta sul video, una schermata alla volta, i dati in ingresso.

Esempio:

Presenta sul video il contenuto del file **capito3.txt**, con arresto ogni volta che lo schermo è pieno:

```
C:\>more < capito3.txt
```

ogni volta che lo schermo è pieno l'ultima riga porta la scritta «-- **Continua** --» e il processo si arresta, per riprendere non appena si batte un tasto.

Esempi d'uso dei comandi filtro

C:\>**sort /r < file.srt > invfile.srt**

il file in ingresso, **FILE.SRT**, viene ordinato con ordinamento inverso (grazie all'opzione **/r**) e posto nel file **INVFILE.SRT**.

C:\>**find /v /i "roma" < file.dat**

cerca la stringa "roma", in qualunque combinazione di caratteri maiuscoli e minuscoli (grazie all'opzione **/i**), nel file **FILE.DAT** e presenta sullo schermo tutte le righe che **non** la contengono (opzione **/v**).

C:\>**more < file1.pas**

visualizza (una pagina alla volta) il contenuto del file **FILE1.PAS**.

Piping

Mentre con la *ridirezione* si dirotta l'input o l'output di un comando ad un file (o dispositivo), con il *piping* si dirotta l'input o l'output di un comando ad un altro comando.

Il *piping* consiste dunque nella concatenazione di due o più comandi; viene rappresentato con il simbolo “ | ”.

I comandi concatenati in successione **devono** essere *comandi filtro* (tranne il primo).

Il collegamento viene realizzato attraverso file temporanei gestiti dal sistema operativo.

Esempi di piping e ridirezione

```
C:\>dir | sort | more
```

Lista file e *sotto-directory* della *directory* corrente su C, in ordine alfabetico e per schermate successive.

```
C:\>dir a: | find " COM" | sort | more
```

Presenta, fermandosi ad ogni schermata, l'elenco ordinato dei files con estensione COM della *directory* corrente sul disco A.

Si osservi che il comando FIND non può distinguere il campo *estensione* dagli altri campi presenti sulla lista prodotta da DIR; tuttavia, dato il formato della lista, la stringa " COM" (si noti lo spazio bianco iniziale) non può presentarsi altro che nel campo *estensione*.

L'equivalente, usando la sola ridirezione, avrebbe comportato la creazione di una serie di file temporanei:

```
C:\>dir a: > temp0.dat
C:\>find " COM" < temp0.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" <iscrizio.dat | sort>iscritti.a93
Produce il file ISCRITTI.A93 (ordinato alfabeticamente) che contiene tutte le linee del file ISCRIZIO.DAT in cui compare la stringa "1993".
```

File di comandi

In MS-DOS è possibile mandare in esecuzione sequenze di comandi memorizzate su file (*file batch*, *programma batch* o *file di comandi*). I comandi da eseguire si trovano in file di nome:

nome.BAT

L'estensione "BAT" informa il sistema operativo che il file è da sottoporre all'interprete dei comandi.

L'invocazione di un file di comandi provoca l'esecuzione dei comandi che lo compongono nell'ordine in cui sono stati registrati, come se fossero battuti alla tastiera.

Qualunque comando che può essere dato da tastiera può far parte di un file *batch*, quindi anche il nome di un altro file *batch* o di un programma applicativo.

Vi sono alcuni comandi tipici dei file *batch*, che non vengono praticamente mai dati da tastiera; essi sono:

CALL Esegue un altro programma batch, poi torna al primo

ECHO

FOR Esegue un comando per ogni elemento di un insieme indicato

GOTO Salta alla riga contrassegnata da una *label*

IF Dispone l'esecuzione su condizione di un comando

PAUSE Sospende temporaneamente l'esecuzione

REM Identifica una riga di commento

SHIFT Cambia la posizione dei parametri sostituibili

I *parametri sostituibili* sono simboli (%0, %1, %2, ecc.) includibili nel file *batch*, che durante l'esecuzione sono sostituiti dai valori forniti all'invocazione.

Il file AUTOEXEC.BAT

Il file AUTOEXEC.BAT contiene la sequenza di comandi (tipicamente di configurazione e inizializzazione) con cui si predisponde la macchina all'utilizzo.

Esso deve trovarsi nella *directory* principale del disco da cui viene avviato MS-DOS. Ogni volta che il Sistema Operativo viene avviato, vengono automaticamente eseguiti i comandi contenuti in AUTOEXEC.BAT.

Se si modifica questo file, affinché le modifiche abbiano effetto è sufficiente mandarlo in esecuzione inserendo da tastiera il comando:
AUTOEXEC

Esempio di AUTOEXEC.BAT

```
@ECHO OFF

KEYB IT
PATH C:\DOS;C:\TP
APPEND C:\DATA
PROMPT $P$G
SET TEMP=C:\TEMP
DOSKEY
CLS
```

Il comando KEYB che si trova in questo esempio ha la funzione di caricare in memoria il programma di gestione per una tastiera nazionale, in questo caso per quella italiana.

Il file CONFIG.SYS

Prima di cercare il file AUTOEXEC.BAT, MS-DOS esegue un gruppo di **comandi speciali** che caricano i *driver*³ installabili e riservano dello spazio nella memoria centrale del sistema per l'uso dei dischi.

Il file che contiene questi comandi è il CONFIG.SYS. Esso deve trovarsi nella *directory* principale del disco da cui viene avviato MS-DOS.

Per configurare il sistema in base alle proprie esigenze, è possibile aggiungere nuovi comandi al CONFIG.SYS o modificare quelli esistenti.

Dal momento che il file CONFIG.SYS presiede alla impostazione di MS-DOS, esso viene letto solo all'avvio del sistema. Se lo si modifica, affinché le modifiche abbiano effetto, è necessario riavviare il sistema.

Può contenere 15 tipi diversi di comandi, **non accettati direttamente da tastiera**. Importanti sono:

DEVICE carica un *driver* installabile.

BUFFERS specifica la memoria da riservare per il trasferimento delle informazioni ai/dai dischi.

FILES Imposta il numero di files che MS-DOS consente di aprire contemporaneamente.

Esempio:

```
DEVICE=C:\MOUSE\MOUSE.SYS
BUFFERS=30
FILES=30
```

³*Driver* o, più propriamente, *Device Driver* è un programma che gestisce e controlla un certo dispositivo (per es. una unità a disco o a nastro, o una particolare unità video) estendendo in tal modo le capacità del Sistema Operativo.

La SHELL di MS-DOS

Molte operazioni per la gestione dei file e la manutenzione dei dischi, oltre che mediante comandi eseguiti dalla riga di comando (dopo il *prompt*), possono essere eseguite dalla *SHELL*.

La *SHELL* fornisce un'alternativa grafica al *prompt* dei comandi. In una sola schermata permette di visualizzare i *drive* (unità), le *directory*, i *file* e le applicazioni pronte per essere utilizzate.

La *SHELL* viene avviata mediante il comando
DOSSHELL

che provoca la suddivisione dello schermo in varie zone, ciascuna riservata ad un tipo di informazioni.

La scelta delle aree da visualizzare e delle operazioni da eseguire può essere effettuata dalla tastiera o con l'uso del *mouse*.

La *SHELL* è dotata di una Guida in linea.

I Virus

I virus sono programmi che si diffondono, ad insaputa degli utenti, copiando se stessi, quindi «infettando», altri programmi o dischi.

Gli effetti prodotti dai virus sono vari. Mentre alcuni si limitano alla emissione occasionale di suoni o messaggi, altri possono alterare il contenuto dello schermo (dati e/o colori). Nei casi più gravi si può arrivare alla distruzione di files o del sistema operativo stesso.

I virus, nati come scherzi innocenti in ambiente universitario negli anni '60, hanno iniziato ad assumere carattere doloso intorno al 1987, forse ad opera di programmatori che, per rancore verso i concorrenti, tentavano in questo modo di sabotarli.

Esistono programmi **Anti-Virus**, ossia programmi di protezione capaci di riconoscere e rimuovere molti virus conosciuti. Poiché vengono continuamente scoperti nuovi virus, questi prodotti vengono periodicamente aggiornato.

Poichè tutti i programmi anti-virus sono in grado di individuare solamente virus già conosciuti, il loro uso non consente di realizzare una protezione totale.

La migliore protezione si ottiene procurandosi il proprio *software* sempre da fonti sicure, creando regolarmente copie di riserva dei propri files di lavoro e proteggendo dalla scrittura tutti i dischi destinati ad essere solo letti.

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	si utilizza un <i>editor</i> per scrivere un programma in forma sorgente;
assemblaggio	si utilizza un <i>assemblatore</i> o un <i>compilatore</i> per averne la forma oggetto;
compilazione	
linking	si utilizza un <i>linker</i> per ottenerne la forma caricabile ed eseguibile;
loading	si utilizza un <i>loader</i> per caricarlo in memoria e mandarlo in esecuzione;

Se sono stati rilevati errori:

debugging	si utilizza un <i>debugger</i> per verificare la correttezza del programma.
------------------	-----------------------------------------------------------------------------

MS-DOS **non** fornisce programmi per assemblaggio o compilazione.

Fino alla versione di MS-DOS 4.1 l'unico editor di testo disponibile era **EDLIN**, che è un editor *di linea*.

Dalla versione 5.0 è disponibile, oltre al precedente, anche **MS-DOS EDITOR**, un editor *a schermo intero*, molto più comodo e facile da usare.

MS-DOS include un debugger non simbolico chiamato **DEBUG**.

Altri strumenti sono forniti all'interno di ambienti di programmazione integrati, come il Programmer's WorkBench del Fortran Professional Development System della MicroSoft.

ESERCIZIO 1

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

```
A: \
  | -CANE:    a.bat b.bat
  | -GATTO:    c.bat d.bat
```

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

```
C: \>mkdir a:\cane\lupo
C: \>chdir a:\cane\lupo
C: \>copy a:\gatto a:
```

Soluzione

Il primo comando crea la *directory LUPO*, figlio della *directory CANE*. Il secondo comando la rende la *directory* corrente **sull'unità A** (ma ciò non è visibile nel *prompt*, che presenta l'unità C, unità corrente). Il terzo comando copia tutti i file della *directory GATTO* nella *directory* corrente dell'unità A, che è \CANE\LUPO.

La situazione finale è la seguente:

```
A: \
  | -CANE:          a.bat    b.bat
  |   | -LUPO:       c.bat    d.bat
  |   |
  | -GATTO:         c.bat    d.bat
```

ESERCIZIO 2

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

C:\

-TERRA:	es1a.txt	es1b.txt	es2a.txt	es2b.txt
-MARE:	ex.txt			

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

```
C:\>copy terra\es?a.txt mare
C:\>ren mare\es*.txt mare\*.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** nella *directory* **MARE**; il secondo li rinomina cambiandone l'estensione in **LUC**. Il comando **REN** non si applica al file **EX.TXT** già esistente nella *directory* **MARE**, in quanto non compreso nell'insieme di file individuati dalla sintassi "**ES*.TXT**" del comando **REN**.

La situazione finale è la seguente:

C:\

-TERRA:	es1a.txt	es1b.txt	es2a.txt	es2b.txt
-MARE:	ex.txt	es1a.luc	es2a.luc	

Comando equivalente:

```
C:\>copy terra\es?a.txt mare\*.luc
```

ESERCIZIO 3

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

```
C: \
  | -ARGO:      win.com
  | -ELIO:      win.exe    won.com
```

indicare come dev'essere predisposto il **PATH** affinché, qualunque sia la *directory* corrente, dando il comando "**WIN**", venga eseguito il programma **WIN.COM**, situato nella *directory* **ARGO**, e non il programma **WIN.EXE** situato nella *directory* **ELIO**, mentre battendo "**WON**" venga regolarmente eseguito il programma **WON.COM** situato nella *directory* **ELIO**.

Soluzione

E' chiaro che, per trovare sia **WIN** nella *directory* **ARGO**, sia **WON** nella *directory* **ELIO**, entrambe queste *directory* dovranno essere elencate nel **PATH**; tuttavia, la necessità di selezionare, dei due **WIN**, quello della *directory* **ARGO**, fa sì che quest'ultima *directory* debba essere elencata *prima dell'altra* all'interno della lista del comando **PATH**.

Dunque, il comando richiesto dovrà essere:

```
C: \>PATH C: \ARGO ; C: \ELIO
```