



Dos: introduzione	4653
Avvio del sistema	4654
Dispositivi secondo il Dos	4656
Directory, file ed eseguibili	4657
Comandi e ambiente di avvio	4660
Caratteri jolly	4663
Invito dell'interprete dei comandi	4664
Comandi interni principali	4665
Flussi standard	4675
Accesso diretto ai dispositivi	4679
Riferimenti	4680
Dos: dischi, file system, directory e file	4681
Suddivisione in partizioni	4682
Inizializzazione di un'unità di memorizzazione	4683
Etichetta di un'unità di memorizzazione	4685
Analisi e correzione del file system	4686
Copia	4687
Trasferimento del sistema	4688
Modifica delle unità	4689
Altre particolarità	4692
Dos: configurazione	4699

File «CONFIG.SYS»	4700
File «AUTOEXEC.BAT»	4708
Comandi ridondanti	4708
Localizzazione	4710
Orologio	4712
Dos: script dell'interprete dei comandi	4715
Parametri, variabili ed espansione	4715
Chiamate di altri script	4717
Strutture di controllo	4717
Comandi utili negli script	4722
Dos: gestione della memoria centrale	4727
Gestione particolare	4727
Comandi appositi	4728
Verifica	4729
FreeDOS	4731
Installazione	4731
Impostazione e configurazione	4734
RxDOS	4735
Riferimenti	4736
Progetto GNUish	4737
Programmi di servizio vari	4738
Gnuplot	4738
Spreadsheet Calculator	4739

Ispell	4739
Perl	4740
Riferimenti	4740
The valuable DOS Freeware page	4741
Introduction	4743
OS and GUI	4744
Utility	4744
Network	4747
Compilers	4748
Typesetting	4749
More Dos software sources	4750
Search engines	4750
Introduzione a ReactOS	4753
Installazione	4753
Riferimenti	4755
DOSEMU: l'emulatore di hardware DOS compatibile	4757
Predisporre un ambiente adatto al Dos all'interno di DOSEMU	4758
La configurazione di DOSEMU	4760
Installare e utilizzare il Dos	4763

Dos: introduzione



Avvio del sistema	4654
Dispositivi secondo il Dos	4656
Directory, file ed eseguibili	4657
Comandi e ambiente di avvio	4660
Caratteri jolly	4663
Invito dell'interprete dei comandi	4664
Comandi interni principali	4665
CH, CHDIR	4666
X:	4667
MD, MKDIR	4667
RD, RMDIR	4668
DIR	4669
COPY	4670
DEL, ERASE	4672
REN, RENAME	4673
SET	4674
TYPE	4675
Flussi standard	4675
SORT	4677
MORE	4678
Accesso diretto ai dispositivi	4679

DOS è acronimo di *Disk Operating System* e sta a indicare il nome di un sistema operativo per micro elaboratori basati su microprocessori i86, successore del vecchio CP/M. Probabilmente, data la sua estrema limitatezza, è un po' azzardato voler parlare di «sistema operativo», tanto che qualcuno lo appella: «gestore di interruzioni» (*interrupt*).

Questo sistema operativo nasce come software proprietario; tuttavia, attualmente il progetto più attivo attorno a questo tipo di sistema è FreeDOS, il cui scopo è quello di realizzarne un'edizione libera e completa.

Avvio del sistema

«

Un sistema Dos è composto essenzialmente da un kernel, un interprete dei comandi e da una serie di programmi di servizio. Questo concetto è analogo ai sistemi Unix, con la differenza che il kernel offre funzionalità molto scarse e solo per mezzo di interruzioni software (IRQ).

Nelle versioni proprietarie del Dos, il kernel è suddiviso in due file, che raccoglievano funzionalità distinte in base all'importanza relativa. I nomi usati sono stati differenti e nel caso di FreeDOS il kernel è contenuto tutto in un solo file (tabella u179.1).

Tabella u179.1. Comparazione tra i nomi dei file che compongono il kernel di un sistema Dos.

Microsoft	IBM	Novell, Cal- dera	RxDOS	FreeDOS
IO.SYS	IBM- BIO.COM	IBM- BIO.COM	RXDO- SBIO.SYS	KER- NEL.SYS
MSDOS.SYS	IBM- DOS.COM	IBM- DOS.COM	RX- DOS.SYS	--

I file del kernel devono trovarsi nella directory radice della partizione o del dischetto per poter essere avviati. Per la precisione, l'avvio del kernel viene gestito direttamente dal codice inserito nel settore di avvio della partizione o del dischetto (512 Kibyte), che a sua volta viene avviato dal firmware (il BIOS, secondo la terminologia specifica dell'architettura i86 e successiva).

Il kernel, dopo essere stato avviato, non attiva una procedura di avvio, ma si limita a interpretare uno script speciale, '**CONFIG.SYS**', e subito dopo avvia l'interprete dei comandi, ovvero la shell. Tradizionalmente, il programma in questione è '**COMMAND.COM**'. Secondo la tradizione, l'interprete dei comandi che viene avviato dal kernel si occupa subito di eseguire lo script '**AUTOEXEC.BAT**'. Gli script '**CONFIG.SYS**' e '**AUTOEXEC.BAT**' devono trovarsi nella directory radice del disco o della partizione da cui si avvia il sistema, ovvero quella in cui si trova già il kernel che viene avviato.

L'interprete dei comandi, '**COMMAND.COM**', è in grado di eseguire direttamente alcune funzionalità, attraverso comandi interni che non si traducono in programmi di servizio veri e propri. Tradizionalmente '**COMMAND.COM**' si colloca nella directory radice del disco o della partizione in cui si trova il kernel stesso. Ciò non è propriamente in-

dispensabile, ma conviene attenersi a questa linea per evitare fastidi inutili.

Dispositivi secondo il Dos

«

I dispositivi secondo il Dos hanno un nome, composto da lettere e cifre numeriche, terminato da due punti opzionali:

nome_dispositivo [:]

Il nome in questione può essere indicato utilizzando lettere maiuscole o minuscole, senza che la cosa faccia differenza. I nomi più comuni sono elencati nella tabella u179.2. È il caso di osservare che i due punti che concludono il nome, vanno usati necessariamente quando questo viene abbinato ad altre informazioni da cui non potrebbe essere distinto (per esempio un percorso).

Tabella u179.2. Nomi dei dispositivi più comuni in Dos.

Dispositivo	Descrizione
'A:'	Disco nella prima unità a dischetti.
'B:'	Disco nella seconda unità a dischetti.
'C:'	Prima partizione Dos nel primo disco fisso.
'D:', 'E:', ... 'Z:'	Partizione Dos o altro tipo di disco.
'CON:'	Console: tastiera e schermo.
'PRN:'	Porta stampante principale.
'LPT1:', 'LPT2:', ...	Porte parallele.
'COM1:', 'COM2:', ...	Porte seriali.

Il Dos mantiene distinti i dischi e le partizioni, nel senso che questi non devono creare una struttura unica come avviene nei sistemi Unix. Pertanto, quando si fa riferimento a un percorso di un file o

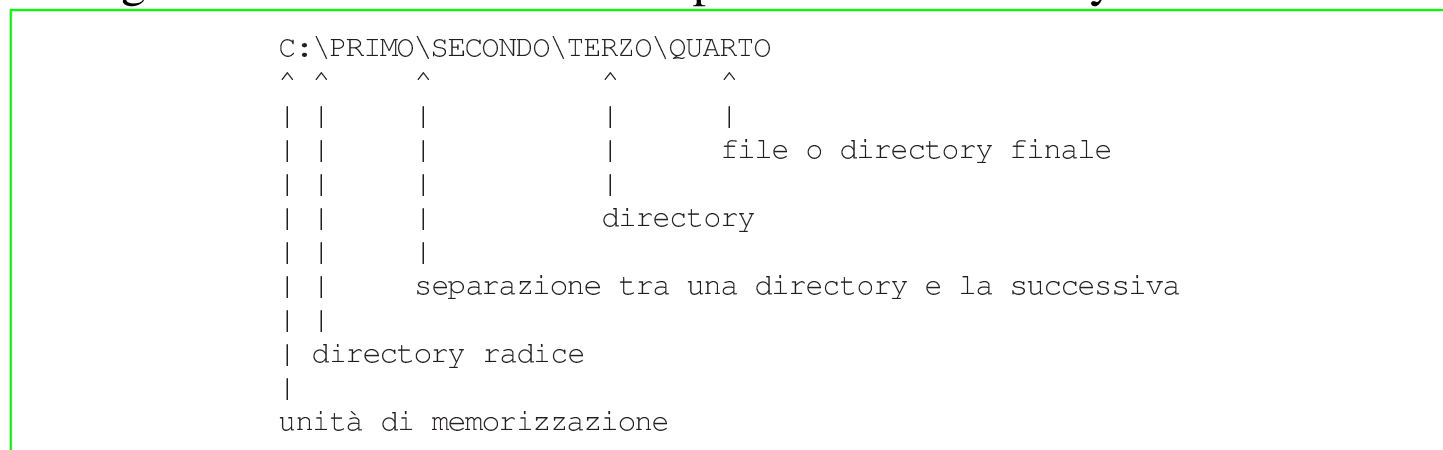
di una directory, si deve tenere in considerazione anche il disco o la partizione in cui si trova.

Il modo utilizzato dal Dos per identificare i dischi e le partizioni, di fatto impedisce di accedere a questi dispositivi in modo indipendente dal file system sottostante. Per intenderci, l'«unità» 'X:' può essere una partizione Dos di un disco non meglio identificato; mentre non esiste un modo univoco per poter raggiungere il dispositivo fisico in cui si trova questo disco.

Directory, file ed eseguibili

Il Dos è nato dopo Unix e da questo sistema ha ereditato alcuni concetti elementari (forse troppo pochi). I percorsi di file e directory si separano con una barra obliqua, che però è inversa rispetto allo Unix. Anche con il Dos c'è una directory radice; tuttavia si aggiunge l'indicazione dell'unità di memorizzazione (il disco o la partizione). Si può osservare a questo proposito la figura u179.3.

Figura u179.3. Struttura di un percorso in un file system Dos.



I nomi di file e directory possono essere indicati utilizzando lettere maiuscole o minuscole, senza che la cosa possa fare differenza. Questi nomi possono essere composti utilizzando anche cifre numeriche e altri simboli (che comunque è bene usare con parsimonia).

Per la precisione, sono esclusi i simboli: ‘/’, ‘\’, ‘[’, ‘]’, ‘<’, ‘>’, ‘+’, ‘=’, ‘;’, ‘:’, ‘,’’, ‘?’’, ‘*’, ‘{’, ‘}’ e il punto che va usato esattamente come descritto nel seguito.

Tradizionalmente, il Dos utilizza un tipo di file system elementare, denominato FAT (Dos-FAT), in cui i nomi dei file e delle directory possono essere composti utilizzando al massimo 11 caratteri, di cui otto compongono un prefisso e tre un suffisso. Il prefisso e il suffisso di questi nomi appaiono uniti attraverso un punto. Per esempio: ‘CIAO.COM’, ‘LETTERA.TXT’, ‘PIPPO.NB’,... Questa conformazione dei nomi è una caratteristica fondamentale del Dos, da cui deriva una serie di consuetudini e di limitazioni molto importanti.

È importante osservare che non è opportuno che i nomi dei file coincidano con quelli dei dispositivi (senza i due punti finali). In pratica, non conviene creare file del tipo ‘CON:’, ‘PRN:’, ecc. Tutto dipende dal contesto, ma in generale è bene fare attenzione a questo particolare.

Come nei sistemi Unix il Dos annovera il concetto di directory corrente, a cui si aggiunge il concetto di unità di memorizzazione corrente. Infatti, la directory va collocata in un disco o in una partizione. In base a questo principio, si possono indicare dei percorsi relativi, che fanno riferimento alla posizione corrente (nell’unità di memorizzazione corrente). Tuttavia, in più, ogni unità di memorizzazione ha una sua directory corrente. Per esempio, fare riferimento a un file in una certa unità di memorizzazione ‘x:’, senza specificare il

percorso, significa indicare implicitamente la directory corrente di quella unità.

Per esempio, supponendo che la directory corrente dell'unità 'X:' sia 'X:\PRIMO\SECONDO\'', facendo riferimento al file 'X:CIAO', si intende indicare implicitamente il file 'X:\PRIMO\SECONDO\CIAO'.

In un percorso si possono usare anche i simboli '.' e '..', con lo stesso significato che hanno in un sistema Unix: la directory stessa e la directory genitrice.

Il file system tradizionale del Dos consente di annotare solo poche informazioni per i file e le directory: la data di modifica e quattro indicatori booleani, rappresentati da altrettante lettere:

- H file o directory nascosti;
- S file o directory di sistema;
- R file o directory in sola lettura e non cancellabile;
- A file o directory da archiviare (i dati sono stati modificati).

Si tratta di attributi completamente differenti da quelli di Unix. Si può osservare in particolare la mancanza di un attributo che specifichi la possibilità di eseguire un programma o di attraversare una directory. Secondo la tradizione Dos, gli attributi vanno considerati nel modo seguente:

- A viene attivato ogni volta che il file viene scritto o modificato e serve per automatizzare i sistemi di copia periodica;
- R se attivo, il Dos non consente la scrittura o la rimozione;
- S se attivo si tratta di un file di «sistema», ma in pratica si comporta come l'attributo H;

- H se attivo si tratta di un file «nascosto», che così non dovrebbe apparire nelle liste di file e directory.

In generale, file e directory nascosti o di sistema non dovrebbero essere spostati fisicamente, nemmeno nell'ambito della stessa unità di memorizzazione. Questa esigenza nasce in particolare per i file del kernel, che non possono essere spostati se si vuole poter riavviare il sistema operativo.

Dal momento che il file system non permette di determinare se un file è un eseguibile, l'unico modo per permettere al sistema di conoscere questa caratteristica sta nell'uso di suffissi convenzionali nei nomi: i file che terminano con l'estensione '`.COM`' e '`.EXE`' sono programmi binari (la differenza tra i due tipi di estensione riguarda il formato del binario); quelli che terminano per '`.BAT`' sono script dell'interprete dei comandi ('`COMMAND.COM`').

La prima stranezza che deriva da questa caratteristica del Dos sta nel fatto che per avviare un eseguibile di questi, è sufficiente indicare il nome del file senza l'estensione, che diventa così un componente opzionale agli occhi dell'utilizzatore.

Comandi e ambiente di avvio

«

L'interprete dei comandi tradizionale dei sistemi Dos è il programma '`COMMAND.COM`', che viene avviato direttamente dal kernel. '`COMMAND.COM`' può essere avviato più volte successive, anche se di solito ciò è di scarsa utilità, dal momento che il Dos non è

un sistema operativo in multiprogrammazione. In ogni caso, quando viene avviato dal kernel, si occupa di interpretare ed eseguire lo script '**AUTOEXEC.BAT**' che si trova nella directory radice dell'unità di avvio.

'**COMMAND.COM**' mostra un invito simile idealmente a quello delle shell Unix, dopo il quale possono essere inseriti i comandi. A loro volta, questi possono essere riferiti a *comandi interni* corrispondenti a funzionalità offerte direttamente dall'interprete, oppure possono rappresentare la richiesta di avvio di un programma esterno.

Figura u179.4. Riga di comando.

```
C:\>DIR A: /W
^      ^      ^      ^
|      |      |      |
|      |      |      | opzione
|      |      |      |
|      |      |      | argomento
|      |      |      |
|      |      |      | comando
|
invito
```

Il Dos ha ereditato da Unix anche il concetto di variabile di ambiente. Il meccanismo è lo stesso ed è fondamentale la variabile di ambiente '**PATH**', con la quale si possono indicare i percorsi di ricerca degli eseguibili. Tuttavia, il Dos ha delle caratteristiche speciali, per cui, è il caso di fare alcuni esempi di comandi:

- C:\>C:\PRIMO\SECONDO.EXE [Invio]

questo comando avvia l'esecuzione del file 'C:\PRIMO\SECONDO.EXE';

- C:\>C:\PRIMO\SECONDO [Invio]

questo comando potrebbe avviare l'esecuzione del primo dei file seguenti che riesce a trovare;

- 'C:\PRIMO\SECONDO.COM'
- 'C:\PRIMO\SECONDO.EXE'
- 'C:\PRIMO\SECONDO.BAT'

• C:\>**SECONDO** [*Invio*]

questo comando potrebbe avviare l'esecuzione del primo dei file seguenti che dovesse riuscire a trovare, ma in mancanza può continuare la ricerca nei percorsi indicati nella variabile di ambiente '**PATH**'.

- 'C:.\SECONDO.COM'
- 'C:.\SECONDO.EXE'
- 'C:.\SECONDO.BAT'

I percorsi indicati nella variabile di ambiente '**PATH**' sono separati da un punto e virgola; per esempio:

```
C:\;C:\DOS;C:\FDOS\BIN
```

Di solito, il Dos dà per scontato che si cerchino gli eseguibili a cominciare dalla directory corrente. Per questo, occorre considerare che è sempre come se la variabile di ambiente '**PATH**' contenesse questa indicazione prima delle altre: '**.;C:\;C:\DOS;C:\FDOS\BIN**'. È da osservare che FreeDOS si comporta in maniera differente, in quanto richiede espressamente questa indicazione della directory corrente.

Caratteri jolly

Il Dos imita l'utilizzo dei caratteri jolly come avviene nei sistemi Unix per opera delle shell. Tuttavia, nel Dos non si tratta di un'espansione che avviene per opera della shell, ma vi deve provvedere ogni programma per conto proprio. Questo rappresenta una gravissima deficienza del Dos, che però è irrimediabile.

Su questa base, i comandi tendono a richiedere l'indicazione di un argomento che rappresenta il nome di uno o più file prima delle opzioni eventuali.

Ma c'è un altro problema. Il punto che divide in due i nomi dei file e delle directory è un muro insuperabile per i caratteri jolly.

I simboli che si possono utilizzare sono solo l'asterisco e il punto interrogativo. L'asterisco vale per una sequenza qualunque di caratteri, escluso il punto; il punto interrogativo vale per un carattere qualunque.¹

Tabella u179.6. Alcuni esempi.

Modello	Corrispondenza
.	Corrisponde a un nome qualunque.
*.COM	Un nome che termina con l'estensione '.COM'.
CIAO.X?X	Tutti i nomi che iniziano per 'CIAO' e hanno un'estensione composta da un lettera «X» iniziale e finale, senza specificare cosa ci sia al secondo posto.
*	Tutti i nomi che non hanno estensione (che non contengono il punto).

Invito dell'interprete dei comandi



Esiste un'altra variabile di ambiente fondamentale per il Dos. Si tratta di **'PROMPT'**, che consente di modificare l'aspetto dell'invito dell'interprete dei comandi. La cosa funziona un po' come nelle shell Unix, per cui si assegna una stringa che può contenere dei simboli speciali, praticamente delle sequenze di escape che vengono espansse prima della visualizzazione. La tabella u179.7 riepiloga questi simboli particolari. In origine, il Dos mostrava in modo predefinito un invito simile all'esempio seguente, in cui appare solo l'unità di memorizzazione corrente:

```
C:>
```

Questo tipo di impostazione corrisponderebbe alla stringa **'\$N\$G'**. In seguito, si è passati a un invito simile al prossimo esempio, in cui si aggiunge anche l'informazione della directory corrente:

```
C:\BIN\>
```

Questo corrisponde alla stringa **'\$P\$G'**.

Tabella u179.7. Sequenze di escape per definire dei componenti speciali all'interno di una stringa di invito.

Simbolo	Corrispondenza
\$Q	=
\$\$	\$
\$T	Ora corrente.
\$D	Data corrente.

Simbolo	Corrispondenza
\$V	Numero della versione.
\$N	Lettera dell'unità corrente.
\$G	>
\$L	<
\$B	
\$H	<BS> (cancella il carattere precedente)
\$E	<ESC> (1B ₁₆)
\$_	Codice di interruzione di riga.

Cancellando il contenuto della variabile di ambiente '**PROMPT**' si ripristina la stringa di invito predefinita.

Comandi interni principali

I comandi interni sono quelli che non corrispondono a programmi di servizio veri e propri, ma sono funzionalità svolte direttamente dall'interprete dei comandi. Nelle sezioni seguenti ne vengono descritti brevemente alcuni.

CH, CHDIR



```
CH [percorso]
```

```
CHDIR [percorso]
```

‘**CH**’, o ‘**CHDIR**’, è un comando interno dell’interprete dei comandi, che consente di visualizzare o di cambiare la directory corrente. È indifferente l’uso di ‘**CD**’ o di ‘**CHDIR**’; se il comando non è seguito dal percorso, si ottiene solo la visualizzazione della directory corrente. Si osservi che se si indica un percorso assoluto di unità di memorizzazione, se questa non corrisponde a quella attuale, si cambia la directory corrente di quella unità.

Segue la descrizione di alcuni esempi.

- `C:\>CD [Invio]`

Visualizza la directory corrente.

- `C:\>CD \TMP\LAVORO [Invio]`

Sposta la directory corrente in ‘\TMP\LAVORO\’.

- `C:\TMP\LAVORO>CD DATI\LETTERE [Invio]`

Sposta la directory corrente in ‘DATI\LETTERE\’ che a sua volta discende dalla posizione iniziale precedente.

- `C:\TMP\LAVORO\DATI\LETTERE>CD .. [Invio]`

Sposta la directory corrente nella posizione della directory genitrice di quella iniziale.

- `C:\TMP\LAVORO\DATI>CD F:\TMP` [Invio]

Cambia la directory corrente dell'unità 'F:', senza intervenire nell'unità corrente.

X:



{A | B | ... | Z} :

Il Dos gestisce le unità di memorizzazione in modo speciale. Per cambiare l'unità di memorizzazione corrente, non esiste un comando analogo a 'CD': si deve indicare il nome dell'unità a cui si vuole accedere.

Segue la descrizione di alcuni esempi.

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

Cambia l'unità di memorizzazione attuale, facendola diventare 'A:'.

- `A:\>F:` [Invio]

Cambia l'unità di memorizzazione attuale, facendola diventare 'F:'.

MD, MKDIR



MD *directory*

MKDIR *directory*

‘**MD**’, o ‘**MKDIR**’, è un comando interno dell’interprete dei comandi, che consente di creare una directory vuota.

Segue la descrizione di alcuni esempi.

- `C:\>MD LAVORO [Invio]`

Crea la directory ‘LAVORO\’ a partire da quella corrente.

- `C:\>MD \TMP\DATA [Invio]`

Crea la directory ‘\TMP\DATA\’ nell’unità corrente.

- `C:\>MD F:\TMP\DATA [Invio]`

Crea la directory ‘\TMP\DATA\’ nell’unità ‘F:’.

RD, RMDIR

«

```
RM directory
```

```
RMDIR directory
```

‘**RD**’, o ‘**RMDIR**’, è un comando interno dell’interprete dei comandi, che consente di cancellare una directory vuota.

Segue la descrizione di alcuni esempi.

- `C:\>RD LAVORO [Invio]`

Cancella la directory ‘LAVORO\’ a partire da quella corrente.

- `C:\>RD \TMP\DATA [Invio]`

Cancella la directory ‘\TMP\DATA\’ nell’unità corrente.

- C:\>RD F:\TMP\DATA [Invio]

Cancella la directory ‘\TMP\DATA\’ nell’unità ‘F:’.

DIR



```
DIR [directory | file] [/P] [/W]
```

‘**DIR**’ è un comando interno dell’interprete dei comandi, che consente di visualizzare l’elenco del contenuto di una directory o l’elenco di un gruppo di file. L’argomento del comando può essere composto utilizzando caratteri jolly, secondo lo standard del Dos, ovvero i simboli ‘*’ e ‘?’.

Tabella u179.8. Alcune opzioni.

Opzione	Descrizione
/P	Blocca lo scorrimento dell’elenco in attesa della pressione di un tasto quando questo è più lungo del numero di righe che possono apparire sullo schermo.
/W	Visualizza solo i nomi dei file e delle directory, senza altre informazioni, permettendo così di vedere più nomi assieme in un’unica schermata.

Segue la descrizione di alcuni esempi.

- C:\>DIR *.* [Invio]

Visualizza l’elenco di tutti i file contenuti nella directory corrente.

- C:\>DIR ESEMPIO.* [Invio]

Visualizza l'elenco di tutti i file il cui nome inizia per 'ESEMPIO' e continua con un'estensione qualunque.

- `C:\>DIR *.DOC [Invio]`

Visualizza l'elenco di tutti i file il cui nome termina con l'estensione '.DOC'.

- `C:\>DIR F:\DOC*.* [Invio]`

Visualizza l'elenco di tutti i file contenuti nella directory '\DOC\' dell'unità 'F:'.

- `C:\>DIR F: [Invio]`

Visualizza l'elenco di tutti i file contenuti nella directory corrente dell'unità 'F:'.

COPY



```
COPY file_origine [file_destinazione] [opzioni]
```

```
COPY file_1 + file_2 [+ ...] [file_destinazione] [opzioni]
```

'COPY' è un comando interno dell'interprete dei comandi, che consente di copiare uno o più file (sono escluse le directory). Anche qui è consentito l'uso di caratteri jolly, ma al contrario dei sistemi Unix, i caratteri jolly possono essere usati anche nella destinazione. Il 'COPY' del Dos consente anche di unire assieme più file.

Tabella u179.9. Alcune opzioni.

Opzione	Descrizione
/V	Fa in modo che venga verificato il risultato della copia.
/B	Assicura che la copia avvenga in modo «binario». Questa opzione può servire quando si copia un file su un dispositivo e si vuole evitare che alcuni codici vengano interpretati in modo speciale.
/Y	Non chiede conferma prima di sovrascrivere i file, se questi esistono già nella destinazione.

Segue la descrizione di alcuni esempi.

- `C:\>COPY ESEMPIO PROVA [Invio]`

Copia il file 'ESEMPIO' nella directory corrente ottenendo il file 'PROVA', sempre nella directory corrente.

- `C:\>COPY C:\DOS*.* C:\TMP [Invio]`

Copia tutto il contenuto della directory '\DOS\' dell'unità 'C:' nella directory '\TMP\' nella stessa unità 'C:', mantenendo gli stessi nomi.

- `C:\>COPY TESTA+CORPO+CODA LETTERA [Invio]`

Copia, unendoli, i file 'TESTA', 'CORPO' e 'CODA', ottenendo il file 'LETTERA'.

- `C:\>COPY *.DOC *.TXT [Invio]`

Copia tutti i file che nella directory corrente hanno un nome che termina con l'estensione '.DOC', generando altrettanti file, con lo stesso prefisso, ma con l'estensione '.TXT'.

- C:\>**COPY PROVA.PRN PRN: /B** [Invio]

Copia il file 'PROVA.PRN' nel dispositivo 'PRN:', ovvero sulla stampante, assicurandosi che la copia avvenga senza alterare alcunché.

DEL, ERASE

<<

DEL *file*

ERASE *file*

'**DEL**', o '**ERASE**', è un comando interno dell'interprete dei comandi, che consente di cancellare uno o più file (sono escluse le directory). È da considerare che i file che hanno l'attributo di sola lettura attivo, non possono essere modificati e nemmeno cancellati.

Segue la descrizione di alcuni esempi.

- C:\TMP>**DEL *.*** [Invio]

Cancella tutti i file nella directory corrente.

- C:\TMP>**DEL ESEMPIO.*** [Invio]

Cancella tutti i file contenuti nella directory corrente, il cui nome inizia per 'ESEMPIO' e termina con qualunque estensione.

- C:\TMP>**DEL *.BAK** [Invio]

Cancella tutti i file contenuti nella directory corrente, il cui nome termina con l'estensione '.BAK'.

REN, RENAME

```
REN file_origine nome_nuovo
```

```
RENAME file_origine nome_nuovo
```

'**REN**', o '**RENAME**', è un comando interno dell'interprete dei comandi, che consente di cambiare il nome di uno o più file (sono escluse le directory). Il primo argomento può essere un percorso relativo o assoluto, completo anche dell'indicazione dell'unità, mentre il secondo argomento è il nuovo nome, che implicitamente non può essere collocato altrove.

Segue la descrizione di alcuni esempi.

- C:\>**REN ESEMPIO PROVA** [*Invio*]

Cambia il nome del file 'ESEMPIO', che si trova nella directory corrente, in 'PROVA'.

- C:\>**REN *.TXT *.DOC** [*Invio*]

Cambia il nome di tutti i file che, nella directory corrente, hanno l'estensione '.TXT', trasformandoli in modo tale da avere un'estensione '.DOC'.

SET



```
SET [variabile_di_ambiente=stringa]
```

‘**SET**’ è un comando interno dell’interprete dei comandi che ha lo scopo di assegnare un valore a una variabile di ambiente, oppure di leggere lo stato di tutte le variabili di ambiente esistenti. Quando si assegna un valore a una variabile, questa viene creata simultaneamente; quando non si assegna nulla a una variabile, la si elimina.

Segue la descrizione di alcuni esempi.

- `C:\>SET [Invio]`

Elenca le variabili di ambiente esistenti assieme al loro valore.

- `C:\>SET PROMPT=PG$G [Invio]`

Assegna alla variabile di ambiente ‘**PROMPT**’ la stringa ‘**\$P\$G\$G**’. Questo si traduce nella modifica dell’aspetto dell’invito dell’interprete dei comandi.

- `C:\>SET PATH=.;C:\BIN;D:\BIN [Invio]`

Assegna alla variabile di ambiente ‘**PATH**’ la stringa ‘**.;C:\BIN;D:\BIN**’.

- `C:\>SET PROMPT= [Invio]`

Elimina la variabile di ambiente ‘**PROMPT**’, assegnandole la stringa nulla.

TYPE

TYPE *file*

‘**TYPE**’ è un comando interno dell’interprete dei comandi, che consente di leggere ed emettere il contenuto di un file attraverso lo standard output. Questo si traduce in pratica nella visualizzazione del file in questione.

Segue la descrizione di alcuni esempi.

- `C:\>TYPE LETTERA` [Invio]

Emette il contenuto del file ‘LETTERA’ che si trova nella directory e nell’unità corrente.

- `C:\>TYPE C:\DOC\MANUALE` [Invio]

Emette il contenuto del file ‘MANUALE’ che si trova nella directory ‘\DOC\’ dell’unità ‘C:’.

Flussi standard

Il Dos ha ereditato da Unix anche i concetti legati ai flussi standard. In pratica, i programmi hanno a disposizione tre flussi predefiniti: uno in lettura rappresentato dallo standard input, due in scrittura rappresentati dallo standard output e dallo standard error. Il meccanismo è lo stesso di Unix, anche se non funziona altrettanto bene; infatti, non è possibile ridirigere lo standard error attraverso l’interprete dei comandi.

Secondo la tradizione delle shell Unix, la ridirezione dello standard output si ottiene con il simbolo ‘>’ posto alla fine del comando inte-

ressato, seguito poi dal nome del file che si vuole generare in questo modo. Per esempio,

```
C:\>TYPE LETTERA > PRN: [Invio]
```

invece di visualizzare il contenuto del file 'LETTERA', lo invia al dispositivo di stampa corrispondente al nome 'PRN: '; inoltre,

```
C:\>DIR *.* > ELENCO [Invio]
```

invece di visualizzare l'elenco dei file che si trovano nella directory corrente, crea il file 'ELENCO' con questi dati.

La ridirezione dello standard output fatta in questo modo, va a cancellare completamente il contenuto del file di destinazione, se questo esiste già; al contrario, si può utilizzare anche '>>', con il quale, il file di destinazione viene creato se non esiste, oppure viene solo esteso.

Lo standard input viene ridiretto utilizzando il simbolo '<', con il quale è possibile inviare un file a un comando utilizzando il flusso dello standard input.

Alcuni comandi hanno la caratteristica di utilizzare esclusivamente i flussi standard. Si parla in questi casi di programmi filtro. Il programma di servizio tipico che si comporta in questo modo è '**SORT**', il quale riceve un file di testo dallo standard input e lo riordina restituendolo attraverso lo standard output. Si osservi l'esempio seguente:

```
C:\>SORT < ELENCO > ORDINATO [Invio]
```

In questo modo, '**SORT**' riceve dallo standard input il file 'ELENCO' e genera attraverso la ridirezione dello standard output il file

‘ORDINATO’.

Per mettere in contatto lo standard output di un comando con lo standard input del successivo, si utilizza il simbolo ‘|’. L’esempio seguente mostra un modo alternativo di ottenere l’ordinamento di un file:

```
C:\>TYPE ELENCO | SORT > ORDINATO [Invio]
```

In generale, tutti i comandi che generano un risultato visuale che scorre sullo schermo, utilizzano semplicemente lo standard output, che può essere ridiretto in questo modo. Si osservi ancora l’esempio seguente che riordina il risultato del comando ‘DIR’, mostrandolo comunque sullo schermo:

```
C:\>DIR *.DOC | SORT [Invio]
```

Nelle sezioni seguenti vengono mostrati alcuni comandi filtro.

SORT



```
SORT [opzioni] < file_da_ordinare > file_ordinato
```

Il comando ‘**SORT**’, che dovrebbe corrispondere a un programma di servizio vero e proprio, riordina il file di testo che ottiene dallo standard input, generando un risultato che emette attraverso lo standard output.

Tabella u179.10. Alcune opzioni.

Opzione	Descrizione
/R	Riordina in modo decrescente

Opzione	Descrizione
<i>/+n_colonna</i>	Riordina in base al testo che inizia a partire dalla colonna indicata come argomento (si tratta di un numero a partire da uno, per indicare la prima colonna).

L'esempio seguente emette l'elenco della directory corrente riordinato in base all'estensione, che è un'informazione collocata a partire dalla decima colonna:

```
C:\>DIR *.DOC | SORT /+10 [Invio]
```

MORE

«

```
MORE < file_da_leggere
```

```
MORE file_da_leggere
```

Il comando '**MORE**' legge un file, fornito come argomento o attraverso lo standard input, mostrandolo poi sullo schermo una pagina dopo l'altra. In questo modo, è possibile leggere il contenuto dei file più lunghi delle righe a disposizione sullo schermo.

Per passare alla pagina successiva, basta premere un tasto qualunque, oppure ciò che viene indicato espressamente.

Segue la descrizione di alcuni esempi.

- C:\>DIR | MORE [Invio]

Permette di controllare lo scorrimento a video del risultato del comando '**DIR**'.

- `C:\>MORE LETTERA.TXT` [*Invio*]

Permette di controllare lo scorrimento a video del contenuto del file 'LETTERA.TXT'.

- `C:\>TYPE LETTERA.TXT | MORE` [*Invio*]

Si ottiene lo stesso risultato dell'esempio precedente, attraverso l'uso di un condotto.

Accesso diretto ai dispositivi

Il Dos offre poche occasioni per accedere direttamente ai dispositivi. Si tratta generalmente solo della console e della porta parallela. L'esempio seguente mostra come «copiare» un file sul dispositivo di stampa, per ottenere così la sua stampa diretta: <<

```
C:\>COPY LETTERA PRN: [Invio]
```

La stessa cosa avrebbe potuto essere ottenuta con la ridirezione dei flussi standard:

```
C:\>TYPE LETTERA > PRN: [Invio]
```

Può essere interessante la possibilità di copiare il flusso di ingresso della console in un file:

```
C:\>COPY CON: LETTERA [Invio]
```

In questo caso, l'inserimento nel file 'LETTERA' prosegue fino a quando viene ricevuto un codice EOF, che si ottiene qui con la combinazione di tasti [*Ctrl z*] seguita da [*Invio*].

È bene ricordare che la console, ovvero il dispositivo 'CON:', riceve dati in ingresso attraverso la tastiera ed emette dati in uscita utilizzando lo schermo. In pratica, quando un programma attende dati

dallo standard input non ridiretto, li riceve dalla console, cioè dalla tastiera; nello stesso modo, quando un programma emette dati attraverso lo standard output non ridiretto, li invia alla console, cioè sullo schermo.

Riferimenti

«

- *FreeDOS*

<http://www.freedos.org>

- *OpenDOS Unofficial Home Page*

<http://www.deltasoft.com/opendos.htm>

¹ Ci sono programmi di origine Unix, portati in Dos, che non hanno questa limitazione riferita al punto che separa l'estensione.

Dos: dischi, file system, directory e file



Suddivisione in partizioni	4682
Inizializzazione di un'unità di memorizzazione	4683
Etichetta di un'unità di memorizzazione	4685
Analisi e correzione del file system	4686
Copia	4687
DISKCOPY	4687
XCOPY	4688
Trasferimento del sistema	4688
Modifica delle unità	4689
ASSIGN	4689
JOIN	4690
SUBST	4691
Altre particolarità	4692
VERIFY	4692
APPEND	4693
ATTRIB	4693
DELTREE	4694
FIND	4694
MOVE	4695
TREE	4696
COMP e FC	4697

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 di-

sco, 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.
/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**’, quan-

do 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 non 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.
/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 al 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 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.

Dos: configurazione



File «CONFIG.SYS»	4700
BREAK	4700
BUFFERS	4701
COUNTRY	4701
DEVICE, DEVICEHIGH	4703
DOS	4704
DRIVEPARM	4704
FCBS	4706
FILES	4706
INSTALL	4706
LASTDRIVE	4706
SHELL	4707
STACK	4708
File «AUTOEXEC.BAT»	4708
Comandi ridondanti	4708
Localizzazione	4710
CHCP	4710
KEYB	4710
GRAFTABL	4712
Orologio	4712

Nel Dos è un po' difficile scindere i concetti di configurazione e script, perché per configurare il sistema, occorre predisporre degli script. Si tratta dei file '`\CONFIG.SYS`' e '`\AUTOEXEC.BAT`', collocati nell'unità di avvio. Questo fatto è già stato accennato nel capitolo introduttivo; in questo si vuole approfondire un po' la cosa.

File «CONFIG.SYS»

«

Il file '`CONFIG.SYS`', collocato nella directory radice dell'unità di avvio, è uno script speciale avviato dal kernel prima dell'interprete dei comandi. In linea di massima, si tratta di una sequenza di direttive che occupano ognuna una riga; alcune versioni recenti del Dos consentono di suddividere le direttive in sezioni da scegliere in base a un menù iniziale.

Le direttive di '`CONFIG.SYS`' hanno la forma seguente:

```
nome=valore
```

In pratica, si assegna una stringa (senza delimitatori espliciti) a un nome che ha un significato particolare.

In questo file, vengono ignorate le righe vuote, quelle bianche e quelle che iniziano con la parola chiave '**REM**':

```
REM annotazione
```

È importante osservare che i nomi delle direttive non fanno differenza tra lettere maiuscole e minuscole. In generale, questo vale anche per le stringhe che vengono assegnate a questi nomi.

BREAK



```
BREAK={ ON | OFF }
```

Teoricamente, questa istruzione consente di attivare o di disattivare la funzionalità abbinata alla combinazione di tasti [*Ctrl c*]. In condizioni normali, quando si assegna la parola chiave ‘ON’, si attiva il funzionamento della combinazione [*Ctrl c*].

BUFFERS



```
BUFFERS=n_buffer [ , n_buffer_secondari ]
```

Questa istruzione consente di definire la quantità di memoria tampone per gli accessi ai dischi. Si assegnano uno o due valori numerici, separati da una virgola. Il primo valore va da 1 a 99 ed esprime il numero di aree da usare come memoria tampone; il secondo valore, facoltativo, indica delle memorie tampone secondarie, con valori che vanno da uno a otto.

COUNTRY



```
COUNTRY=n_codice_paese [ , [ n_codifica ] [ , file_informazioni_nazionali ] ]
```

Questa istruzione, attraverso quanto contenuto in un file che tradizionalmente si chiama ‘COUNTRY.SYS’, permette di configurare il sistema in base alla nazionalità. Per la precisione, si può specificare un codice riferito alla nazionalità, attraverso il quale si ottiene una forma particolare per le date e gli orari, con l’aggiunta eventua-

le di un altro codice che specifica la codifica dei caratteri prescelta (*codepage*). La tabella u181.1 riepiloga questi codici che fanno riferimento tradizionalmente anche a paesi che non esistono più.

Si può osservare che la stringa assegnata alla direttiva '**COUNTRY**' può contenere l'indicazione di un file (con il percorso, completo di unità o meno). Questo file è quello che contiene poi le indicazioni relative alla nazionalità prescelta; come già accennato, di solito si tratta del file '`COUNTRY.SYS`'.

Tabella u181.1. Codici di nazionalità.

Località	Codice di nazionalità	Codifiche utili
USA	001	437, 850
Canada francese	002	863, 850
America latina	003	850, 437
Russia	007	866, 437
Olanda	031	850, 437
Belgio	032	850, 437
Francia	033	850, 437
Spagna	034	850, 437
Ungheria	036	850, 852
Jugoslavia	038	850, 852
Italia	039	850, 437
Svizzera	041	850, 437
Cecoslovacchia	042	850, 852
Regno unito	044	850, 437
Danimarca	045	850, 865
Svezia	046	850, 437
Norvegia	047	850, 865
Polonia	048	850, 852
Germania	049	850, 437
Brasile	055	850, 860

Località	Codice di nazionalità	Codifiche utili
Australia	061	850, 437
Giappone	081	932, 437, 850, 942
Corea	082	934, 437, 850, 944
Cina	088	938, 437, 850, 948
Turchia	090	857, 850
Asia (inglese)	099	850, 437
Portogallo	351	850, 860
Islanda	354	850, 861
Finlandia	358	850, 437

L'esempio seguente predispone l'impostazione nazionale per l'Italia, utilizzando la codifica 850, che ha il vantaggio di essere quella più comune dei paesi che usano l'alfabeto latino:

```
COUNTRY=039,850,C:\DOS\COUNTRY.SYS
```

DEVICE, DEVICEHIGH

```
DEVICE=programma_di_gestione_dispositivo [opzioni]
```

```
DEVICEHIGH=programma_di_gestione_dispositivo [opzioni]
```

Si tratta di un modo per avviare un programma speciale che ha lo scopo di rimanere residente in memoria. In generale, tali programmi servono per la gestione di qualche dispositivo, indispensabile prima di avviare l'interprete dei comandi.

La differenza tra le due direttive sta nel fatto che la seconda cerca di caricare il programma nella memoria «alta».

Le opzioni riguardano il programma.

L'esempio seguente avvia il programma '**MOUSE.SYS**' che presumibilmente gestisce il mouse (l'opzione '/2' serve probabilmente a utilizzare il mouse collegato alla seconda porta seriale):

```
DEVICE=C:\MOUSE\MOUSE.SYS /2
```

DOS

<<

```
DOS= { HIGH | LOW } [ , { UMB | NOUMB } ]
```

```
DOS= [ { HIGH | LOW } , ] { UMB | NOUMB }
```

Questa istruzione richiede al kernel di allocarsi nella memoria convenzionale, '**LOW**', o in quella alta, '**HIGH**'. La parola chiave '**UMB**' richiede di mantenere un collegamento tra la UMB e la memoria convenzionale; la parola chiave '**NOUMB**' fa sì che questo collegamento non abbia luogo.

DRIVEPARM

<<

```
DRIVEPARM= [ opzioni ]
```

Si tratta di una direttiva attraverso cui si possono definire i parametri relativi ai dispositivi a blocchi, per la precisione si tratta solo di dischi, se questo può essere necessario. Le opzioni assomigliano a quelle dei programmi di servizio, iniziando con una barra obliqua normale: '/x...'.

Tabella u181.4. Alcune opzioni.

Opzione	Descrizione
<i>/d:n_dispositivo_fisico</i>	Consente di indicare il dispositivo attraverso un numero, da 0 a 255. Lo zero corrisponde alla prima unità a dischetti.
<i>/c</i>	Se si utilizza questa opzione, si intende che l'unità fisica è in grado di sapere se il disco è inserito o meno.
<i>/f:n_formato</i>	Stabilisce il formato del dispositivo fisico; in pratica, fissa la geometria.
<i>/f:0</i>	Dischetto 160 Kibyte, 180 Kibyte, 320 Kibyte, 360 Kibyte.
<i>/f:1</i>	Dischetto 1 200 Kibyte.
<i>/f:2</i>	Dischetto 720 Kibyte.
<i>/f:5</i>	Disco fisso.
<i>/f:6</i>	Nastro.
<i>/f:7</i>	Dischetto 1 440 Kibyte.
<i>/f:9</i>	Dischetto 2 880 Kibyte.
<i>/h:n_testine</i>	Definisce il numero di testine.
<i>/i</i>	Indica che si tratta di un dischetto da 3,5 pollici.
<i>/n</i>	Si tratta di un disco fisso.
<i>/s:n_settori</i>	Definisce il numero di settori per traccia.

Opzione	Descrizione
<code>/t : n_cilindri</code>	Definisce il numero dei cilindri (in altri termini: il numero di tracce per faccia).

FCBS



```
FCBS=n_blocchi
```

Permette di definire il numero di blocchi di controllo dei file (*file control block*). Il valore va da 1 a 255, mentre il valore normale è di quattro blocchi.

FILES



```
FILES=n_blocchi
```

Permette di indicare il numero massimo di file aperti. Il numero che può essere assegnato va da 8 a 255. Il valore predefinito dovrebbe essere di otto file.

INSTALL



```
INSTALL=programma [opzioni]
```

Si tratta di un'istruzione con la quale si può avviare preventivamente un programma (che dovrebbe essere residente in memoria), prima dell'avvio dell'interprete dei comandi. In questo caso, a differenza della direttiva '**DEVICE**', o '**DEVICEHIGH**', si tratta di un programma normale.

Le opzioni riguardano il programma.

LASTDRIVE

```
LASTDRIVE=lettera_unità_finale
```

Consente di specificare l'ultima lettera di unità che può essere richiesta. Questo consente di risparmiare risorse, se si è consapevoli del fatto che non servono lettere oltre un certo punto. La lettera in questione può essere indifferentemente maiuscola o minuscola, senza che ciò possa fare differenza.

SHELL

```
SHELL=programma [opzioni]
```

Permette di indicare esplicitamente il programma da avviare alla fine della procedura di avvio del kernel. In generale si tratta dell'interprete dei comandi. Questa direttiva può consentire di avviare un interprete alternativo a quello normale, oppure permette di avviarlo da una collocazione insolita; inoltre permette di dare al programma in questione delle opzioni particolari.

L'esempio seguente avvia il programma '**COMMAND.COM**' che si trova nella directory 'C:\DOS\':

```
SHELL=C:\DOS\COMMAND.COM
```

STACK

```
STACK=nlivelli [ , dimensione_in_byte ]
```

Con questa istruzione è possibile fissare la dimensione dello stack, utilizzando valori da 8 a 64, oltre allo zero. Il valore dopo la virgola indica la dimensione in byte di ogni livello dello stack. In questo caso i valori vanno da 32 a 512.

File «AUTOEXEC.BAT»

Il file 'AUTOEXEC.BAT' collocato nella directory radice dell'unità di avvio, è inteso essere uno script che viene eseguito dall'interprete dei comandi, 'COMMAND.COM', dopo l'avvio del sistema.

Questo script viene realizzato normalmente in modo sequenziale, senza strutture di controllo. In generale è importante per due cose: impostare alcune variabili di ambiente fondamentali, per esempio 'PATH'; avviare dei programmi che poi restano residenti in memoria, quando questo non si ottiene già attraverso il file '\CONFIG.SYS'.

Comandi ridondanti

Anche nel Dos è molto importante l'uso delle variabili di ambiente. È già stato mostrato il comando 'SET', attraverso il quale si impostano o si annullano le variabili di ambiente:

```
SET nome_variabile=stringa_assegnata
```

Alcune variabili hanno un'importanza particolare, per cui esiste un comando interno apposito (dell'interprete dei comandi), che serve a inizializzarle senza nemmeno l'uso del comando '**SET**'.

- `PROMPT stringa_di_invito`

Il comando interno '**PROMPT**' rappresenta un modo alternativo per impostare la variabile di ambiente con lo stesso nome. Se si usa il comando senza l'argomento, si ripristina l'invito predefinito.

- `PATH [percorsi_degli_eseguibili]`

Il comando interno '**PATH**' rappresenta un modo alternativo per impostare la variabile di ambiente con lo stesso nome. Se non si indica l'argomento, si ottiene la visualizzazione dell'elenco dei percorsi attivo.

Esistono altri comandi particolari che si sovrappongono alle istruzioni del file '`CONFIG.SYS`'.

- `BREAK [ON | OFF]`

Abilita o disabilita la funzionalità abbinata alla combinazione di tasti [*Ctrl c*]. Utilizzando il comando senza argomento, si ottiene la visualizzazione dello stato attuale.

Localizzazione



La localizzazione del Dos si riduce alla configurazione della mappa della tastiera e alla definizione dell'insieme di caratteri. L'insieme di caratteri dipende dalla scelta della nazionalità, fatta nel file 'CONFIG.SYS', attraverso la direttiva '**COUNTRY**'.

Nelle sezioni seguenti vengono mostrati alcuni comandi utili per le impostazioni che riguardano la localizzazione.

CHCP



```
CHCP [n_codifica]
```

Si tratta di un comando interno dell'interprete dei comandi che interviene nella definizione della codifica utilizzata. In pratica, se si utilizza senza argomenti, mostra il numero della codifica attiva; se si indica un numero come argomento, cambia la codifica attiva, purché questa sia una di quelle ammissibili in base alla nazionalità stabilita con la direttiva '**COUNTRY**' nel file di configurazione 'CONFIG.SYS'.

L'esempio seguente fa in modo che sia attivata la codifica corrispondente al numero 850:

```
C:\>CHCP 850 [Invio]
```

KEYB



```
KEYB [sigla_nazionale [, [n_codifica] [, file_informazioni_tastiere ] ] ]
```

'**KEYB**' è un comando esterno che consente di cambiare la configurazione della tastiera secondo alcuni modelli di nazionalità predefiniti.

La sigla nazionale è un codice di due lettere che, assieme alla nazionalità, dovrebbe indicare anche la lingua utilizzata. La tabella u181.6 elenca queste sigle.

Tabella u181.6. Sigle nazionali-linguistiche per l'impostazione della mappa della tastiera.

Sigla	Corrispondenza
US	USA (predefinito)
FR	Francia
GR	Germania
IT	Italia
SP	Spagna
UK	Gran Bretagna
PO	Portogallo
SG	Svizzera tedesca
SF	Svizzera francese
DK	Danimarca
BE	Belgio
NL	Olanda (Nederland)
NO	Norvegia
LA	America latina
SV	Svezia
SU	Finlandia (Suomi)
CF	Canada francese

Segue la descrizione di alcuni esempi.

- C:\>**KEYB** [Invio]

Mostra la configurazione attuale.

- C:\>**KEYB IT** [Invio]

Predisporre la mappa dei tasti per la disposizione italiana.

- `C:\>KEYB IT, 850, C:\DOS\KEYBOARD.SYS [Invio]`

Predisporre la mappa dei tasti per la disposizione italiana, specificando l'uso della codifica 850 e del file 'C:\DOS\KEYBOARD.SYS' per trovare le impostazioni standard delle tastiere.

GRAFTABL

«

```
GRAFTABL [n_codifica]
```

```
GRAFTABL /STATUS
```

'**GRAFTABL**' è un comando esterno che consente di cambiare la codifica per i caratteri visualizzati sullo schermo. L'opzione '**/STATUS**' permette di conoscere la situazione attuale, mentre l'indicazione di un numero di codifica cambia l'impostazione.

L'esempio seguente imposta l'uso della codifica 850:

```
C:\>GRAFTABL 850 [Invio]
```

Orologio

«

Il Dos consente di accedere all'orologio dell'elaboratore, per leggere la data e l'ora, o per cambiare tali informazioni. In generale, il Dos non prevede la gestione di un orologio hardware allineato al tempo universale; pertanto, l'orologio hardware deve corrispondere necessariamente all'ora locale, lasciando all'utente il problema legato alle variazioni dell'ora estiva.

I comandi per accedere all'orologio sono **'DATE'** e **'TIME'**:

```
DATE [ data ]
```

```
TIME [ orario ]
```

Se non si indica la data o l'orario, viene mostrato quello attuale e viene richiesto all'utente di modificarlo o di confermarlo.

Il modo in cui va scritta da data o l'ora, dipende dalla localizzazione. Per conoscere quello giusto, basta osservare in che modo vengono visualizzate tali informazioni.

Dos: script dell'interprete dei comandi



Parametri, variabili ed espansione	4715
Chiamate di altri script	4717
Strutture di controllo	4717
IF	4717
FOR	4719
GOTO	4720
Emulazione di un ciclo iterativo	4721
Comandi utili negli script	4722
REM	4722
ECHO	4722
PAUSE	4723
CLS	4724
CHOICE	4724

Uno script dell'interprete dei comandi, conosciuto solitamente con il nome di file *batch*, potrebbe essere definito come un file di testo normale in cui può essere indicato un elenco di comandi da eseguire. Tuttavia, questi script consentono l'uso anche di strutture di controllo elementari, per cui si possono realizzare dei programmi molto semplici, senza troppe pretese.

È interessante osservare che questi script vengono individuati solo attraverso l'estensione che ha il nome: '.BAT'. Inoltre, non esiste la necessità di renderli «eseguibili» come si fa nei sistemi Unix.

Parametri, variabili ed espansione



Gli script dell'interprete dei comandi hanno accesso agli argomenti che vengono loro forniti. Si possono gestire solo nove di questi argomenti alla volta, attraverso i parametri posizionali relativi, da '%1' a '%9'. Come avviene nelle shell Unix, è disponibile il comando interno '**SHIFT**' per fare scorrere in avanti gli argomenti nei parametri disponibili.

Bisogna ricordare che in Dos i caratteri jolly non vengono espansi dalla shell, per cui la limitazione a soli nove parametri posizionali, non dovrebbe costituire un problema.

Nell'ambito di uno script possono essere dichiarate e utilizzate delle variabili di ambiente. È già stato mostrato in precedenza l'uso del comando '**SET**' per impostare o eliminare le variabili di ambiente. Per fare riferimento al contenuto di una variabile, si usa la notazione seguente:

```
%nome_variabile%
```

L'esempio seguente rappresenta il caso tipico di estensione di un percorso di ricerca degli eseguibili, quando si ritiene che la variabile '**PATH**' sia già stata usata:

```
SET PATH=%PATH%;C:\PIPP0
```

Chiamate di altri script

Tradizionalmente, il Dos ha un baco molto grave, ormai divenuto una caratteristica fondamentale, riguardante l'avvio di script all'interno di altri script. In generale, quando si chiama un programma che in realtà corrisponde a uno script, al termine di questo non riprende l'esecuzione di quello chiamante. Per ottenere la ripresa dell'interpretazione dello script di partenza occorre usare il comando speciale **'CALL'**.

```
CALL nome_script [argomenti_dello_script]
```

Strutture di controllo

Le strutture di controllo per la programmazione attraverso gli script dell'interprete dei comandi sono molto limitate. È disponibile una struttura condizionale semplificata e un ciclo di scansione di file, che vengono descritti brevemente.

IF

```
IF [NOT] ERRORLEVEL valore_di_uscita_ultimo_comando comando
```

```
IF [NOT] stringa_1==stringa_2 comando
```

```
IF [NOT] EXIST file comando
```

La struttura condizionale degli script dell'interprete dei comandi Dos è in pratica un comando interno dello stesso interprete. Come si

può vedere dagli schemi sintattici, viene fornita una condizione che può essere invertita con la parola chiave **'NOT'** e il risultato è solo l'esecuzione di un altro comando se la condizione risulta vera.

Nel primo caso, la condizione si riferisce alla verifica del valore di uscita dell'ultimo comando eseguito. La condizione si verifica se il numero indicato è inferiore o uguale al valore restituito effettivamente da tale comando; nel secondo, la condizione si verifica se le due stringhe (non delimitate) sono identiche; nel terzo si verifica la condizione se il file indicato esiste effettivamente.

Segue la descrizione di alcuni esempi.

- ```
IF ERRORLEVEL 1 GOTO :errore
```

Se il comando precedente ha restituito un valore maggiore o uguale a uno, salta all'etichetta **' :errore '**.

- ```
IF %1==ciao ECHO L'argomento è corretto
```

In questo caso, se l'espansione del parametro **'%1'**, corrispondente al primo argomento ricevuto all'avvio, si traduce nella stringa **'ciao'**, viene emesso un messaggio per mezzo del comando **'ECHO'**.

- ```
IF %1x==x ECHO L'argomento è mancante
```

Quello che si vede è il trucco necessario per poter verificare se un parametro contiene la stringa nulla: si aggiunge una lettera, in questo caso una «x», verificando che la corrispondenza avvenga solo con la stessa lettera.

- ```
IF NOT EXIST LETTERA.TXT ECHO Scrivi! > LETTERA.TXT
```

Qui, se non esiste il file ‘LETTERA.TXT’ nella directory corrente, questo file viene creato attraverso il comando ‘ECHO’ che invia il suo standard output verso un file con lo stesso nome.

FOR



```
FOR [%] %x IN (nome...) DO comando [argomenti_del_comando]
```

Si tratta di un comando interno che svolge un ciclo di scansione di un gruppo di nomi, generalmente file, attraverso il quale viene creato un parametro variabile speciale, il cui nome si compone di una sola lettera, a cui viene assegnato a ogni ciclo uno dei nomi contenuti tra parentesi tonde. A ogni ciclo viene eseguito il comando, che a sua volta può fare uso del parametro.¹

Quando viene usato all’interno di uno script dell’interprete dei comandi, il parametro viene indicato con due simboli di percentuale (‘%%x’); al contrario, se il comando viene impartito dalla riga di comando, se ne usa uno solo.

Segue la descrizione di alcuni esempi.

- ```
FOR %A IN (uno due tre) DO ECHO %A
```

In questo modo, si ottiene la visualizzazione delle parole ‘**uno**’, ‘**due**’ e ‘**tre**’. In pratica, è come se fosse stato fatto:

- ```
ECHO uno  
ECHO due  
ECHO tre
```

Volendo fare la stessa cosa dalla riga di comando, è necessario il raddoppio del simbolo ‘%’:

```
C:\>FOR %%A IN (uno due tre) DO ECHO %%A [Invio]
```

- ```
FOR %A IN (*.TMP *.BAD) DO DEL %A
```

Cancella, uno a uno, tutti i file che terminano con le estensioni ‘.TMP’ e ‘.BAD’.

## GOTO

«

```
GOTO etichetta
```

Gli script dell’interprete dei comandi dispongono dell’istruzione di salto incondizionato, non avendo di meglio. Anche questa istruzione può essere presa come un comando interno dell’interprete, con la differenza che non c’è modo di utilizzarlo al di fuori di uno script.

Nel corso di uno script del genere, possono apparire delle righe che contengono solo un’etichetta, nella forma:

```
:nome_etichetta
```

La posizione corrispondente a queste etichette può essere raggiunta con il comando ‘**GOTO**’, che può fare riferimento solo al nome dell’etichetta, oppure a tutta l’etichetta, includendo anche i due punti.

Segue la descrizione di alcuni esempi.

- ```
IF EXIST LETTERA.TXT GOTO riprendi  
ECHO Il file LETTERA.TXT è assente  
:riprendi
```

In questo esempio, se il file ‘LETTERA.TXT’ esiste, si salta all’etichetta ‘**:riprendi**’; altrimenti si esegue il comando ‘**ECHO**’.


```
IF EXIST LETTERA.TXT GOTO :riprendi
ECHO Il file LETTERA.TXT è assente
:riprendi
```

Esattamente come nell'esempio precedente, con la differenza che il comando '**GOTO**' indica l'etichetta con i suoi due punti iniziali.

Emulazione di un ciclo iterativo

Dal momento che non è disponibile una struttura di controllo per il ciclo iterativo, questo può essere ottenuto solo attraverso l'uso del comando '**GOTO**'. Vale la pena di mostrare in che modo si può ottenere tale risultato.

```
: etichetta_di_ingresso
IF condizione GOTO : etichetta_di_uscita
...
...
...
GOTO etichetta_di_ingresso
: etichetta_di_uscita
```

```
: etichetta_di_ingresso
...
...
...
IF condizione GOTO : etichetta_di_ingresso
: etichetta_di_uscita
```

I due modelli sintattici mostrano due esempi di cicli iterativi. Nel primo caso si verifica una condizione, in base alla quale si decide se

proseguire o se terminare il ciclo; nel secondo si esegue una volta il ciclo e quindi si verifica una condizione per decidere se ripeterlo o se uscire.

Comandi utili negli script

« Alcuni comandi sono particolarmente utili all'interno di script dell'interprete dei comandi. Vengono descritti brevemente nelle sezioni seguenti.

REM

REM *commento*

I commenti negli script dell'interprete dei comandi si indicano attraverso un comando apposito: '**REM**'. Il funzionamento è evidente: tutto quello che segue il comando, fino alla fine della riga, viene ignorato.

Alcune edizioni del Dos hanno introdotto anche l'uso del punto e virgola, come simbolo per indicare l'inizio di un commento. Segue un esempio tipico di utilizzo di questo comando:

```
REM  
REM (c) 2000 Pinco pallino  
REM
```

ECHO

ECHO [ON | OFF]

ECHO *stringa*

Il comando interno **'ECHO'** ha un significato duplice: da una parte consente di visualizzare un testo; dall'altra controlla la visualizzazione dei comandi contenuti in uno script. Infatti, si distingue il fatto che l'eco dei comandi sia attivo o meno. utilizzando il comando **'ECHO'** senza argomenti, si ottiene l'informazione sul suo stato di attivazione. Di solito si disattiva l'eco dei comandi negli script.

Per disattivare l'eco di un comando particolare, senza disattivare l'eco in generale, basta inserire inizialmente il simbolo '@'.

Segue la descrizione di alcuni esempi.

- `@ECHO OFF`

Disattiva l'eco dei comandi, facendo in modo che anche questo comando non venga visualizzato (si usa per questo il simbolo '@').

- `ECHO Premi un tasto per continuare`

Mostra un messaggio per spiegare come comportarsi.

PAUSE

PAUSE

Il comando interno **'PAUSE'** sospende l'esecuzione di uno script in attesa della pressione di un tasto. Il comando emette attraverso lo standard output un messaggio di avvertimento in tal senso. Di solito, per evitare di vedere tale messaggio, si ridirige lo standard output in un file nullo.

L'esempio seguente, prima mostra un messaggio in cui si avverte che per proseguire occorre premere un tasto, quindi si usa il comando '**PAUSE**' che sospende l'esecuzione dello script, senza però mostrare altri messaggi:

```
ECHO Premere un tasto per proseguire  
PAUSE > C:\NULL
```

CLS

<<

```
CLS
```

Il comando interno '**CLS**' ripulisce lo schermo. Si utilizza senza argomenti.

CHOICE

<<

```
CHOICE [ opzioni ] [ testo_di_invito ]
```

Il comando '**CHOICE**' serve a presentare una richiesta per l'inserimento di una lettera, tra un elenco determinato. La pressione del tasto corrispondente alla lettera scelta, da parte dell'utilizzatore, provoca la conclusione del funzionamento di '**CHOICE**' che restituisce un valore corrispondente alla scelta: zero per la prima lettera, uno per la seconda,...

Si osservi che l'ultimo argomento rappresenta un messaggio che serve all'utente per comprendere il senso della scelta che sta facendo.

Tabella u182.15. Alcune opzioni.

Opzione	Descrizione
<code>/C: lettera [lettera...]</code>	Permette di fissare l'elenco di lettere che possono essere usate nella risposta. Se non si indica questa opzione, la scelta avviene solo tra 'y' e 'n'.
<code>/N</code>	Questa opzione fa in modo di escludere la visualizzazione delle lettere che possono essere scelte. In questo modo si fa affidamento esclusivamente sul testo indicato come ultimo argomento.
<code>/S</code>	Distingue tra maiuscole e minuscole per quanto riguarda le lettere tra cui scegliere.

L'esempio seguente, salta a un punto differente dello script in base alla scelta di una lettera da «a» a «f». Si osservi che non sarebbe possibile eseguire l'analisi secondo una sequenza differente, perché **'IF ERRORLEVEL'** prende in considerazione tutti i valori di uscita maggiori o uguali a quanto indicato nella condizione.

```
CHOICE /C:abcdef Inserisci una lettera
IF ERRORLEVEL 5 GOTO :f
IF ERRORLEVEL 4 GOTO :e
IF ERRORLEVEL 3 GOTO :d
IF ERRORLEVEL 2 GOTO :c
IF ERRORLEVEL 1 GOTO :b
IF ERRORLEVEL 0 GOTO :a
...
```

¹ Questo parametro assomiglia a una variabile di ambiente, ma non si comporta allo stesso modo. Si tratta di una particolarità del comando **'FOR'**.

Dos: gestione della memoria centrale

Gestione particolare	4727
Comandi appositi	4728
Verifica	4729

Quando è nato il Dos non si prevedeva l'uso di memoria centrale oltre il singolo mebibyte (1 Mibyte). In base a questa considerazione veniva articolata l'architettura hardware degli elaboratori «XT» e poi «AT», dove si prevedeva l'uso di un massimo di 640 Kibyte di memoria centrale, riservando la parte successiva, fino alla fine di 1 Mibyte, per la memoria video e altri dispositivi fisici.

In questo senso, il Dos tradizionale può operare con un massimo di 640 Kibyte di memoria centrale; per sfruttarne di più occorrono degli accorgimenti non facili da applicare.

Gestione particolare

Per sfruttare la memoria oltre il primo mebibyte, si fa uso normalmente di due programmi, avviati attraverso 'CONFIG.SYS', prima ancora dell'interprete di comandi. Si tratta di '**HIMEM.SYS**' e di '**EMM386.EXE**'. In generale, le cose si fanno nel modo seguente:

```
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE
```

Il primo dei due programmi può essere utilizzato a partire da architetture i286, mentre il secondo si può inserire solo a partire da architetture i386.

‘**HIMEM.SYS**’ è in grado di utilizzare solo una piccola parte di memoria aggiuntiva, mentre ‘**EMM386.EXE**’ permette teoricamente di sfruttare tutto il resto.

In generale, è molto difficile la gestione ottimale della memoria centrale, perché le applicazioni si comportano in maniera differente. Di solito si possono solo fare dei tentativi.

Comandi appositi

«

Per sfruttare la memoria centrale che supera la soglia convenzionale, sono disponibili alcuni comandi specifici. In generale, si comincia dalla configurazione con il file ‘**CONFIG.SYS**’: dopo l’attivazione dei gestori speciali della memoria, è possibile indicare di collocare parte dell’interprete dei comandi e dello spazio richiesto dai programmi residenti in memoria, oltre il limite della memoria convenzionale:

```
DOS=HIGH,UMB
```

In seguito, sempre nell’ambito del file ‘**CONFIG.SYS**’, si può richiedere esplicitamente l’avvio di programmi nella memoria alta attraverso la direttiva ‘**DEVICEHIGH**’, come si vede nell’esempio seguente:

```
DEVICEHIGH=C:\MOUSE\MOUSE.SYS /2
```

Per quanto riguarda i programmi avviati attraverso l’interprete dei comandi, è disponibile il comando ‘**LH**’, ovvero ‘**LOADHIGH**’:

```
LH programma [argomenti_del_programma]
```



```
LOADHIGH programma [argomenti_del_programma]
```

Per esempio, si potrebbe tentare di avviare in questo modo il programma di gestione della tastiera:

```
C:\>LH KEYB IT [Invio]
```

Verifica

Il Dos offre un solo programma molto semplice per la verifica dell'utilizzo della memoria: **'MEM'**. «

```
MEM [opzioni]
```

Se **'MEM'** viene usato senza opzioni, visualizza brevemente la quantità di memoria utilizzata rispetto al totale disponibile. È interessante l'opzione **'/CLASSIFY'**, attraverso la quale è possibile distinguere l'utilizzo della memoria da parte dei programmi residenti; inoltre è interessante l'opzione **'/FREE'**, con cui si hanno informazioni dettagliate sulla memoria libera.

Le opzioni disponibili del comando **'MEM'** variano molto da una realizzazione all'altra. In generale conviene verificare prima di utilizzarlo, per conoscere le possibilità effettive.

FreeDOS



Installazione	4731
Impostazione e configurazione	4734
RxDOS	4735
Riferimenti	4736

FreeDOS ¹ è il nome di un progetto per la realizzazione di un sistema operativo libero compatibile con il Dos. Il Dos, per quanto limitato, ha delle caratteristiche che lo possono rendere ancora interessante per elaboratori con architettura i86 particolarmente poveri di risorse, come nel caso dei sistemi cosiddetti *embedded*.

Installazione

L'installazione della distribuzione standard di FreeDOS è abbastanza semplice. Si parte da un dischetto di avvio, con il quale si prepara la partizione e la si inizializza, quindi si prosegue con il programma di installazione che chiede l'inserimento dei dischetti successivi. La riproduzione del dischetto di avvio a partire dalla sua immagine avviene come al solito attraverso il programma '**RAWRITE.EXE**', oppure per mezzo di un sistema Unix nei modi già mostrati per GNU/Linux e altri sistemi simili.

```
C:\>RAWRITE FULL.BIN A: [Invio]
```

L'esempio mostra l'uso di '**RAWRITE.EXE**' per ottenere un dischetto dall'immagine rappresentata dal file 'FULL.BIN'.



La distribuzione standard di FreeDOS si compone di un file-immagine del dischetto di avvio, che potrebbe chiamarsi 'FULL.BIN', e da una serie di file con estensione '.ZIP' che servono per ottenere i dischetti successivi. Ognuno di questi file compressi rappresenta il contenuto di un dischetto, che quindi deve essere prima estratto:

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

```
A:\>UNZIP C:\TMP\BASE1.ZIP [Invio]
```

L'esempio mostra in breve il procedimento: ci si sposta nell'unità 'A:' e da lì si estrae il file compresso che probabilmente si trova da qualche parte nel disco fisso.

Questi file compressi rappresentano una raccolta di applicativi e hanno una struttura particolare che viene descritta nel seguito.

- *nome_raccolta . 1*

L'archivio compresso deve contenere un file che rappresenta il nome della raccolta, con un'estensione numerica. La raccolta potrebbe essere suddivisa in più archivi ed è per questo che si usa l'estensione numerica, che indica il numero di sequenza dell'archivio nell'ambito della raccolta.

Il file contiene l'elenco dei pacchetti contenuti, con l'indicazione dell'opzione di installazione predefinita o meno. Si osservi l'estratto seguente (la lettera «Y» rappresenta la conferma all'installazione predefinita):

```
asgn14x: Y
attr063x: Y
bwb210x: Y
choic20x: Y
```

- ***nome_raccolta* .END**

Si tratta di un file vuoto, che rappresenta la conclusione della raccolta, nel senso che non ci sono altri dischetti ulteriori.

- ***nome_pacchetto* .LSM**

Si tratta di un file che descrive un pacchetto applicativo. Quello che segue è l'esempio del contenuto del file 'DELTR10X.LSM':

```
Begin3
Title:          deltree
Version:        1.02b
Entered-date:   27 Jul 1999
Description:    Delete a directory and all directories under it
Keywords:       freedos delete
Author:         raster@highfiber.com
Maintained-by: raster@highfiber.com
Primary-site:   http://www.highfiber.com/~raster/freeware.htm
Alternate-site: www.freedos.org
Original-site:  http://www.highfiber.com/~raster/freeware.htm
Platforms:     dos
Copying-policy: GPL
End
```

- ***nome_pacchetto* .ZIP**

Si tratta dell'archivio compresso che contiene i file dell'applicativo. In base alla struttura standard di FreeDOS, potrebbe distribuirsi nelle directory 'BIN\'', 'DOC\' e 'HELP\'.

Dopo aver preparato i dischetti, si può procedere con l'avvio del sistema attraverso il dischetto di avvio; quindi si passa a predisporre la partizione:

```
A:\>FDISK [Invio]
```

Purtroppo, il kernel di FreeDOS non è in grado di gestire partizioni più grandi di 512 Mibyte, per cui occorre tenerne conto durante l'uso di '**FDISK**'. Dopo aver preparato la partizione la si inizializza:

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

Successivamente si trasferisce il sistema, con il comando '**SYS**':

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

Infine si avvia il programma di installazione che provvede a chiedere la sostituzione dei dischetti:

```
A:\>INSTALL [Invio]
```

Impostazione e configurazione

«

Da quanto è stato descritto sull'installazione di FreeDOS si intende che, pur trattandosi di un sistema Dos, si cerca di introdurre qualche buona idea proveniente da Unix. In particolare, è prevista una struttura per la collocazione dei file:

- '**BIN**' per contenere i file eseguibili;
- '**DOC**' per contenere la documentazione che si articola in altre sottodirectory successive, come avviene con GNU/Linux
- '**HELP**' per contenere i file della guida interna relativa.

Questa struttura potrebbe essere collocata anche a partire da un punto differente della radice dell'unità, in base alle scelte fatte in fase di installazione. In ogni caso, occorre poi predisporre coerentemente alcune variabili di ambiente: '**PAGER**' per indicare il programma da utilizzare per lo scorrimento dei file delle guide; '**HELPPATH**' per indicare la directory contenente i file delle guide; '**EMACS**' per indicare la directory contenente i file di Emacs.

In condizioni normali, gli applicativi FreeDOS vengono installati a partire dalla directory '`\FDOS\`', per cui la configurazione si traduce nelle istruzioni seguenti nel file '`AUTOEXEC.BAT`':

```
SET PAGER=MORE
SET HELPPATH=C:\FDOS\HELP
SET EMACS=C:\FDOS\EMACS\
```

In base alla documentazione originale, nel caso della variabile di ambiente '**EMACS**' deve essere indicata la barra obliqua inversa finale.

A seconda della distribuzione di FreeDOS, può darsi che il file '`CONFIG.SYS`' debba essere sostituito con uno avente un nome differente. Potrebbe trattarsi del file '`FDCONFIG.SYS`'.

RxDOS

RxDOS è un altro progetto analogo a FreeDOS, scritto in maniera indipendente. È provvisto di un proprio interprete dei comandi e non ha ancora un suo sistema di installazione. Per provare il funzionamento di RxDOS ci si può avvalere solo di un dischetto, realizzato nel modo seguente:

1. si inizializza il dischetto in qualche modo, assicurando che alla fine sia disponibile un file system Dos-FAT;²
2. si esegue lo script **‘MAKEBOOT.BAT’**, il cui scopo è la predisposizione del settore di avvio nel dischetto;
3. si copiano ordinatamente nel dischetto i file elencati qui sotto.
 - ‘RXDOSBIO.SYS’
 - ‘RXDOS.SYS’
 - ‘RXDOSCMD.EXE’
 - ‘RXDVDISK.SYS’
 - ‘AUTOEXEC.DEF’
 - ‘CONFIG.DEF’

Riferimenti

«

- *FreeDOS*

<http://www.freedos.org>

¹ **FreeDOS** GNU GPL

² Il dischetto non deve avere l’etichetta, ovvero non deve avere un nome.

Progetto GNUish



Programmi di servizio vari	4738
Gnuplot	4738
Spreadsheet Calculator	4739
Ispell	4739
Perl	4740
Riferimenti	4740

Il progetto «GNUish» è una sorta di derivazione povera del progetto GNU, con lo scopo di rendere disponibile parte del software che compone il sistema GNU anche nei sistemi Dos. Il progetto ha un'importanza molto piccola, ma viene ancora mantenuto. Evidentemente, date le peculiarità dei sistemi Dos, il software che viene adattato non può avere le stesse potenzialità che ha invece in un sistema Unix.

I siti principali da cui si può ottenere copia del materiale prodotto dal progetto GNUish sono quelli elencati all'interno dal documento seguente:

- http://www.math.utah.edu/docs/info/gnuish_1.html

In questo capitolo viene mostrato il funzionamento di alcuni programmi, nell'ambito del sistema Dos, per i quali è il caso di spendere qualche parola.

Programmi di servizio vari

«

Molti dei programmi di servizio del progetto GNU sono disponibili anche per Dos. Tuttavia, è il caso di osservare alcune particolarità che possono confondere chi è abituato a usare sistemi Dos.

La prima cosa da notare è il fatto che i percorsi si possono indicare secondo lo stile Unix, utilizzando barre oblique normali. Per esempio:

```
C:\>MV C:/PRIMO/SECONDO C:/TERZO [Invio]
```

Diversamente, utilizzando lo stesso comando, ma secondo l'indicazione tipica del Dos, la cosa può funzionare ugualmente, oppure si possono presentare delle segnalazioni di errore. Bisogna tenere presente la possibilità.

Un'altra cosa da notare è l'uso dei caratteri jolly, che con questi programmi segue la logica di Unix, dove l'asterisco indica qualunque nome, senza trattare in modo speciale il punto di separazione dell'estensione:

```
C:\>CP C:/PRIMO/SECONDO/* C:/TERZO [Invio]
```

L'esempio mostra proprio questo fatto: vengono copiati tutti i file contenuti nella directory 'C:\PRIMO\SECONDO\' , nella directory 'C:\TERZO\'.

Gnuplot

«

Per funzionare, questa edizione di Gnuplot richiede due file: 'GNU PLOT.EXE' e 'GNU PLOT.GIH'. Il primo dei due è l'eseguibile in grado di gestire la grafica VGA, mentre il secondo contiene le informazioni della guida interna.

Se si vuole accedere alla guida interna, è necessario che il file 'GNUPLOT.GIH' si trovi nella directory corrente. Forse è sufficiente utilizzare il comando '**APPEND**' del Dos per risolvere il problema.

Spreadsheet Calculator

Il funzionamento generale di SC (Spreadsheet Calculator) è descritto nel capitolo [u10](#). La versione per Dos funziona correttamente (è sufficiente disporre dell'eseguibile '**SC.EXE**'), riconoscendo anche l'uso dei tasti freccia, per cui non si è più costretti a utilizzare le lettere 'h', 'j', 'k' e 'l'.

Ispell

Questa edizione di Ispell richiede due file: 'ISPELL.EXE' e 'ISPELL.DIC'. Come si intuisce, il primo è l'eseguibile, mentre il secondo è il file del dizionario. Purtroppo, il file 'ISPELL.DIC' non è sostituibile o eliminabile; l'unica cosa che si può fare è predisporre un dizionario personalizzato che si richiama con l'opzione '-p'.

```
ISPELL [-d dizionario_standard] [-p dizionario_personale] file
```

Quella che si vede è la sintassi essenziale su cui si può contare nell'edizione di Ispell per Dos. Il file del dizionario standard, 'ISPELL.DIC', può essere collocato nella stessa directory in cui si trova il file eseguibile; altrimenti si deve usare l'opzione '-d' per indicarlo esplicitamente.

Il dizionario personale è un file di testo normale (Dos), che può anche essere creato inizialmente dallo stesso Ispell. L'esempio seguente, mostra il caso in cui si voglia analizzare il file 'LETTERA.TXT' attraverso il dizionario standard e il dizionario personale 'VOCAB.TXT'. Se il file 'VOCAB.TXT' non dovesse esistere, verrebbe creato per l'occasione.

```
C:\LETTERE>ISPELL -p VOCAB.TXT LETTERA.TXT [Invio]
```

Perl

«

L'edizione Dos dell'interprete Perl richiede due file: 'PERL.EXE' e 'PERLGLOBAL.EXE'. È sufficiente che questi siano disponibili nei percorsi degli eseguibili della variabile di ambiente 'PATH'.

Bisogna tenere a mente che si tratta di una versione molto vecchia del linguaggio, per cui alcune novità non possono essere disponibili. Inoltre, l'avvio dei programmi può avvenire solo richiamando direttamente l'interprete:

```
C:\ESERCIZI>PERL FATT.PL 5 [Invio]
```

L'esempio mostra l'avvio del programma Perl contenuto nel file 'FATT.PL', che riceve un argomento costituito dal numero cinque.

Riferimenti

«

- François Pinard, *GNUish MSDOS Project*
http://www.math.utah.edu/docs/info/MSDOS_1.html

The valuable DOS Freeware page



Introduction	4743
OS and GUI	4744
Utility	4744
Network	4747
Compilers	4748
Typesetting	4749
More Dos software sources	4750
Search engines	4750

Links to valuable free Dos programs working on low equipped computers.

This material appeared originally at ‘<http://www.geocities.com/SiliconValley/7737/>’, in 1996. Now it is incorporated inside the Italian document “a2”, and it might be reached at the URI http://a2.swlibero.org/the_valuable_dos_freeware_page.html.

Questo materiale è apparso in origine, nel 1996, presso ‘<http://www.geocities.com/SiliconValley/7737/>’. Adesso viene incorporato nel documento «a2» e può essere raggiunto attraverso l’URI http://a2.swlibero.org/the_valuable_dos_freeware_page.html. L’intento dell’autore è solo quello di continuare a curare un vecchio lavoro che potrebbe essere ancora utile, nonostante si tratti di riferimenti a software in parte libero e in parte solo gratuito, oltre che evidentemente obsoleto.

Introduction	4743
OS and GUI	4744
Utility	4744
Network	4747
Compilers	4748
Typesetting	4749
More Dos software sources	4750
Search engines	4750

Introduction



The Dos operating system meant much for many people. Today, proprietary Dos-like operating systems seem to be no more developed. In this situation, the only possible future for Dos is the “free” software, and it is not just a matter of money anymore.

Unfortunately, “free” is a word with many meanings. Today, this is still the biggest obstacle to the future of the Dos world. There is so much software for Dos, with so many different license agreements. The typical Dos user doesn't mind to it. But this problem prevents the realization of big serious projects based on it.

Today, the Dos world needs *philosophy*, and the GNU idea is still the right one (<http://www.gnu.org>).

The author of this space would like to list here only “free software” in the sense stated by the Free Software Foundation, **but it is impossible**, as there isn't enough good real free software for Dos.

The listed software is meant to work on i286 and below.

It is attempted to give some kind of classification about the legal condition of the software presented here. The definition used might be outdated, or there might be other wrong assumption. In particular, the definition “public domain” means here, in most cases, that there is the source, but there is no clear license statement.

Beside the URI links of some FTP services there is an additional “search link” that queries a FTP search engine for the same file. These additional links should be used when there are troubles with the main links.

Anyone can link this document anywhere, so, there is no need to ask

for it. Anyway, it is better to link to this document using at the file name http://a2.swlibero.org/the_valuable_dos_freeware_page.html .

In the future, many links may disappear on this page, because of more selective choices concerning software license.

OS and GUI

«

- FreeDOS ¹ <http://www.freedos.org>
- nanoDos, a FreeDOS distribution with some networking programs, ready to use/install: <ftp://na.mirror.garr.it/mirrors/AppuntiLinux/nanoDos/>
Italian documentation can be found in chapter [u62](#).
- FreeGEM ² <http://l.webring.com/hub?ring=freegem>

Utility

«

archive, backup

- Gzip ³ - ‘.GZ’ archive compressor and extractor. <ftp://ftp.simtel.net/pub/simtelnet/msdos/compress/gzip124.zip> <http://www.google.com/search?q=gzip124.zip>
- TAR ⁴ - portable TAR - DOS/UNIX backup, compressor, with hardware support. <ftp://ftp.simtel.net/pub/simtelnet/msdos/arcers/tar320g.zip> <http://www.google.com/search?q=tar320g.zip>
- Untgz ⁵ - ‘.TGZ’, ‘.TAR’, ‘.GZ’, ‘.ZIP’ file extractor. <ftp://ftp.simtel.net/pub/simtelnet/msdos/arcers/untgz095.zip> <http://www.google.com/search?q=untgz095.zip>

- Info-ZIP ⁶ - '.ZIP' compatible compression and extraction utility. <http://www.info-zip.org/pub/infozip/> <ftp://ftp.info-zip.org/pub/infozip/MSDOS/>
- Restaur ⁷ - Replacement for Dos Restore, <ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/restaur1.zip> <http://www.google.com/search?q=restaur1.zip>

communication

- DosFax ⁸ - Send a fax using Dos command line <http://www.Adr.de/speicherplatz/cs/dosfax.htm>
- Bgfax ⁹ <http://www.blkbox.com/~bgfax/>
- Rifs ¹⁰ - Disk sharing over a serial line, <ftp://ftp.simtel.net/pub/simtelnet/msdos/lan/dosrifs2.zip> <http://www.google.com/search?q=dosrifs2.zip>

directory, file

- WCD ¹¹ - Powerful chdir for Dos and Unix <http://www.xs4all.nl/~waterlan/>

disk

- Dug_ide ¹² - Report ATA (IDE) disks geometry <ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/dugide21.zip> <http://www.google.com/search?q=dugide21.zip>
- Fips ¹³ - Non-destructive splitting of hard disk partitions <ftp://ftp.simtel.net/pub/simtelnet/msdos/diskutil/fips15.zip> <http://www.google.com/search?q=fips15.zip>
- Part ¹⁴ - MBR partition manager <http://www.ranish.com/part/>

help

- NG_clone ¹⁵ - Norton Guides clone <ftp://ftp.simtel.net/pub/simtelnet/msdos/txtutl/ngclon11.zip> <http://www.google.com/search?q=ngclon11.zip>

shell

- DC ¹⁶ - The Dos Controller - A Norton Commander clone <ftp://ftp.simtel.net/pub/simtelnet/msdos/fileutil/dc-sk.zip> <http://www.google.com/search?q=dc-sk.zip>

system

- Cmos ¹⁷ - Save/Restore extended C/MOS <ftp://ftp.simtel.net/pub/simtelnet/msdos/sysutl/cmos93cd93.zip> <http://www.google.com/search?q=cmos93cd93.zip>
- Cmoser ¹⁸ - Save, restore and modify C/MOS memory <ftp://ftp.simtel.net/pub/simtelnet/msdos/sysutl/cmoser11.zip> <http://www.google.com/search?q=cmoser11.zip>
- Bios ¹⁹ - Save, restore, erase C/MOS memory; save a BIOS copy to a file <http://www.bockelkamp.de/software/discontinued/bios1351.zip>
- KGB ²⁰ - Utility to monitor some Dos functions and reporting into a log file <ftp://ftp.simtel.net/pub/simtelnet/msdos/sysutl/kgb104.zip> <http://www.google.com/search?q=kgb104.zip>

text

- Vim ²¹ - VI improved, a small text editor that can handle very big files with low RAM <ftp://ftp.simtel.net/pub/simtelnet/msdos/editor/vim53d16.zip> <http://www.google.com/search?q=vim53d16.zip>

packet driver

- PC/TCP Packet Driver Collection ²² <ftp://ftp.crynwr.com/drivers/> <ftp://ftp.crynwr.com/drivers/pktd11.zip>
- WATTCP ²³ - TCP/IP library routines <http://www.wattcp.com/>
- DOS PPPD ²⁴ - Dos port of Linux PPP packet driver <ftp://ftp.simtel.net/pub/simtelnet/msdos/pktdrvr/dosppp05.zip> <http://www.google.com/search?q=dosppp05.zip>
- Comring ²⁵ - packet driver emulating ethernet over serial link(s) <http://wizard.ae.krakow.pl/~jb/ComRing/>

TCP/IP

- WATTCP apps ²⁶ - some common client application using WATTCP library <http://www.smashco.com/wattcp/apps.zip>
- MiniTelnet ²⁷ - TELNET client <http://www.smashco.com/wattcp/mt.zip>
- Bobcat ²⁸ - Text based web browser <http://www.fdisk.com/doslynx/bobcat.htm> (derived from DosLynx, <ftp://ftp2.cc.ukans.edu/pub/WWW/DosLynx/>)
- Arachne (GPL) ²⁹ - Graphical web browser <http://home.hetnet.nl/~ba8tian/arachne/arachne.htm> , <http://home.hetnet.nl/~ba8tian/arachne/175-gpl/ar175.htm> , <http://home.hetnet.nl/~ba8tian/arachne/175-gpl/a175gp75lf.zip>
- PCroute ³⁰ - IP routing program for IBM PC <ftp://ftp.simtel.net/pub/simtelnet/msdos/network/pcrte224.zip> <http://www.google.com/search?q=pcrte224.zip>

- PPRD ³¹ - Turn a dedicated PC (XT/AT) into a LPD server <ftp://ftp.simtel.net/pub/simtelnet/msdos/lan/pprd200.zip> <http://www.google.com/search?q=pprd200.zip>
- NCSA Telnet ³² - Telnet, Ftp,... NCSA <http://archive.ncsa.uiuc.edu/SDG/Software/PCTelnet/> <ftp://ftp.ncsa.uiuc.edu/Telnet/DOS/> <ftp://ftp.simtel.net/pub/simtelnet/msdos/ncsatlnt/>
- NOS (KA9Q) ³³ - A complete mini TCP/IP system <ftp://ftp.simtel.net/pub/simtelnet/msdos/tcpip/> <http://www.google.com/search?q=e920603.zip>
To use NOS you need documentation, for example the package <ftp://ftp.simtel.net/pub/simtelnet/msdos/tcpip/intronos.zip>
- SSHDOS ³⁴ - SSH client for Dos <http://sourceforge.net/projects/sshdos>
- Talk ³⁵ - Talk client for Dos <http://www.smashco.com/wattcp/talk-13.zip>
- ABC-nslookup ³⁶ - DNS query clients for Dos <http://www.smashco.com/wattcp/nslb01a.zip>

See also:

- Marc S. Ressler, *Dos Internet Pages* <http://www.fdisk.com/doslynx/>
- Smash-Co Communications, *TCP/IP for MS-DOS* <http://www.smashco.com/wattcp.asp>
- *The U-M Software Archive* <http://www.umich.edu/~archive/msdos/communications/wattcp/>
<http://www.umich.edu/~archive/msdos/communications/packet/>

Compilers



assembler

See the FreeDOS project (<http://www.freedos.org>) for assembler compilers.

batch

- BAT2EXE ³⁷ - Compile batch files for speed <ftp://ftp.simtel.net/pub/simtelnet/msdos/batchutl/bat2ex15.zip> <http://www.google.com/search?q=bat2ex15.zip>

C/C++

See the FreeDOS project (<http://www.freedos.org>) for C and C++ compilers.

Perl

Perl ³⁸ - Practical Extraction Report Language <ftp://ftp.simtel.net/pub/simtelnet/msdos/perl/>

Rexx

- BREXX ³⁹ - Rexx interpreter for Dos/Unix <http://ftp.gwdg.de/pub/languages/rexx/brexx/html/rx.html>

xBase

- nanoBase ⁴⁰ - Mini, but nearly complete xBase http://a2.swlibero.org/nanobase_1997.html

See also:

- David Muir Shamoff, *Catalog of free compilers and interpreters* <http://www.idiom.com/free-compilers/>

Typesetting

«

- Nro ⁴¹ - A Nroff implementation for Dos <ftp://ftp.simtel.net/pub/simtelnet/msdos/txtutl/nroff1.zip> <http://www.google.com/search?q=nroff1.zip>
- Ghostscript ⁴² - “GNU” original edition - PostScript pre-viewing, conversion, and printing <ftp://mirror.cs.wisc.edu/pub/mirrors/ghost/gnu/>
- emTeX ⁴³ - TeX-LaTeX distribution for Dos <ftp://www.ctan.org/tex-archive/systems/msdos/emtex/>

More Dos software sources

«

- <ftp://ftp.simtel.net/pub/simtelnet/msdos/>
- <http://garbo.uwasa.fi/pc/>

Search engines

«

- <http://www.shareware.com/>

¹ **FreeDOS** GNU GPL

² **FreeGEM** GNU GPL

³ **Gzip** GNU GPL

⁴ **TAR (Dos)** public domain

⁵ **Untgz** GNU GPL

⁶ **Info-ZIP** free software with special license

- 7 **Restaur** cannot be sold for profit
- 8 **DosFax** public domain
- 9 **Bgfx** promised to become free software
- 10 **Rifs** cannot be sold for profit
- 11 **WCD** GNU GPL
- 12 **Dug_ide** free of charge, with sources
- 13 **Fips** GNU GPL
- 14 **Part** public domain
- 15 **NG_clone** public domain
- 16 **DC** public domain (no license at all, and no sources)
- 17 **Cmos** public domain
- 18 **Cmoser** free of charge
- 19 **Bios** free of charge
- 20 **KGB** public domain
- 21 **Vim** free software with special license
- 22 **Crynwr packet driver collection** GNU GPL
- 23 **WATTCP** free of charge library
- 24 **DOS PPPD** mixed licenses
- 25 **Comring** GNU GPL
- 26 **WATTCP apps** cannot be sold
- 27 **MiniTelnet** free software with a special license
- 28 **Bobcat** GNU GPL

- 29 **Arachne** GNU GPL
- 30 **PCroute** cannot distribute modifications
- 31 **PPRD** software non libero: licenza Artistic
- 32 **NCSA Telnet** public domain
- 33 **NOS** public domain
- 34 **SSHDOS** GNU GPL
- 35 **Talk** GNU GPL
- 36 **ABC-nslookup** UCB BSD
- 37 **BAT2EXE** public domain
- 38 **Perl** GNU GPL or Artistic
- 39 **BREXX** public domain
- 40 **nanoBase** GNU GPL
- 41 **Nro** public domain
- 42 **Ghostscript** GNU GPL
- 43 **emTeX** LPPL but some files have different conditions

Introduzione a ReactOS



Installazione	4753
Riferimenti	4755

ReactOS ¹ è un progetto per la realizzazione di un sistema operativo conforme al funzionamento di MS-Windows, a partire da NT in su. Queste annotazioni sono state fatte a proposito della versione 0.1.2, dove il suo sviluppo non ha ancora portato a un sistema grafico funzionante e appare come un sistema Dos, anche se non è compatibile con gli eseguibili Dos standard. A partire dalle versioni 0.2.* è già disponibile un sistema grafico elementare.

Teoricamente ReactOS dovrebbe essere in grado di funzionare a partire da elaboratori i386 in su; tuttavia, è probabile che si riesca a utilizzare solo da i686 in su, escludendo anche i microprocessori compatibili.

Installazione

Allo stato della versione 0.1.2, il programma di installazione non è ancora in grado di gestire le partizioni e nemmeno di inizializzarle; pertanto, la partizione dove va collocato ReactOS deve essere preparata con altri strumenti (un sistema GNU/Linux o FreeDOS). Inizialmente conviene limitarsi all'utilizzo di un file system Dos-FAT16;



Il programma di installazione tenta di riconoscere il settore di avvio del disco e della partizione in cui viene installato ReactOS; se questi settori risultano essere di qualche tipo particolare, non vengono modificati e poi il sistema non si avvia. Anche se la cosa è spiacevole, nel caso siano presenti altri sistemi nel disco fisso, può essere necessario fare in modo che nella partizione stabilita si avvii FreeDOS; in pratica, serve un dischetto di avvio di FreeDOS (capitolo [u184](#)), con il quale si deve ripristinare il settore di avvio del disco fisso:

```
A:>FDISK /MBR [Invio]
```

Quindi si deve trasferire il sistema minimo nella partizione, con il comando:

```
A:>SYS C: [Invio]
```

Naturalmente, la partizione in questione deve risultare attiva, ovvero deve essere quella che viene «avviata».

Per poter installare ReactOS, è necessario disporre di un CD-ROM, che si ottiene scaricando il file-immagine, seguendo le indicazioni contenute nel sito <http://reactos.com>. Il file-immagine dovrebbe risultare essere un file compresso, in formato '.zip'; una volta estratto, si può usare per incidere un CD-ROM.

Il CD che si ottiene dovrebbe essere autoavviabile, inserendosi automaticamente nelle procedura di installazione. L'installazione è relativamente breve, dal momento che il sistema è ancora allo stato iniziale del suo sviluppo. Al termine, se non ci sono stati problemi, viene sistemato anche il sistema di avvio; pertanto dovrebbe bastare riavviare l'elaboratore per mettere in funzione la copia di ReactOS

appena installata.

ReactOS viene installato a partire dalla directory ‘\ReactOS\’, ma il sistema di avvio rimane all’inizio ed è composto dai file ‘\freeldr.sys’ e ‘\freeldr.ini’; eventualmente, il secondo di questi due può essere modificato per cambiare il menù di Avvio.

Una volta fatto funzionare il sistema, se è necessario, occorre intervenire nuovamente nel settore di avvio del disco fisso, in modo da poter usare sistemi come GRUB o LILO, attraverso i quali si deve fare in modo di avviare il primo settore della partizione contenente ReactOS, oltre che gli altri sistemi operativi che probabilmente sono installati nello stesso disco.

Riferimenti

- *ReactOS*
<http://reactos.com>
- SourceForge.net: *ReactOS*
<http://sourceforge.net/projects/reactos>

¹ **ReactOS** GNU GPL

DOSEMU: l'emulatore di hardware DOS compatibile



Predisporre un ambiente adatto al Dos all'interno di DOSEMU	
4758	
Un disco «C:» immagine	4759
Un disco «D:» virtuale o di rete	4759
La struttura essenziale del disco «D:» virtuale	4759
La configurazione di DOSEMU	4760
File «/etc/dosemu.users»	4760
File «/etc/dosemu.conf»	4762
Installare e utilizzare il Dos	4763
I dischi virtuali con «LREDIR.COM»	4766
Il mouse	4767
Un esempio di «AUTOEXEC.BAT»	4767
DOSEMU e le console virtuali	4767
DOSEMU da X	4768
DOSEMU e Mtools	4768
Implicazioni sulla gestione dei permessi	4769

DOSEMU ¹ è fondamentalmente un emulatore dell'hardware x86 per vari sistemi Unix funzionanti su architettura i386. Il suo obiettivo è quello di permettere il funzionamento del sistema operativo Dos (MS-Dos o cloni). Si tratta di un progetto eternamente in fase

di sviluppo (*alpha*), anche se da diversi anni è sufficientemente funzionante. Tuttavia non ci sono punti fermi: da una versione all'altra si possono incontrare novità imprevedibili.

Dal momento che l'emulazione riguarda l'hardware, il Dos deve essere installato all'interno di questo sistema di emulazione; quindi, è necessaria una copia di questo sistema operativo, insieme alla licenza d'uso.

DOSEMU permette di utilizzare la stessa copia installata del Dos su più terminali contemporaneamente. Se si intende concedere l'utilizzo simultaneo di una singola copia di questo sistema operativo, è necessario un numero maggiore di licenze d'uso, oppure una licenza multipla.

A fianco del lavoro su DOSEMU è anche in corso quello sul progetto FreeDOS per un sistema operativo Dos libero (capitolo [u184](#)).

Predisporre un ambiente adatto al Dos all'interno di DOSEMU

«

Perché il sistema operativo Dos possa funzionare all'interno di DOSEMU, occorre preparare un file-immagine di un disco Dos dal quale si possa effettuare l'avvio del Dos stesso. Questo file che viene descritto di seguito, viene visto dal Dos come disco 'C:'.

Successivamente è conveniente predisporre uno spazio all'interno del file system del proprio sistema GNU/Linux da utilizzare per i programmi Dos che deve essere letto come un disco di rete.

Un disco «C:» immagine

Per effettuare l'avvio del Dos occorre che sia predisposta l'immagine di un disco di piccole dimensioni. Questo potrebbe essere un file contenuto nella directory `‘/var/lib/dosemu/’`, oppure `‘/var/state/dosemu/’`, il cui nome inizia normalmente per `‘hdimage’`.

Attualmente, il file dovrebbe chiamarsi `‘hdimage.first’` e al limite potrebbe essere un collegamento simbolico a un altro file che costituisce l'immagine vera e propria.

Se non esiste questo file è necessario copiarlo dal pacchetto sorgente. Il nome dovrebbe essere `‘hdimage.dist’`, o qualcosa di simile. Questa immagine deve essere preparata in seguito.

Un disco «D:» virtuale o di rete

In questa fase conviene preparare una directory che definisca l'inizio (la radice) del disco `‘D:’` virtuale utilizzato dai programmi Dos. Stabiliamo che questo sia `‘/var/emul/dos/’`. Da questo punto in poi, `‘D:\’` è equivalente a `‘/var/emul/dos/’`.

La struttura essenziale del disco «D:» virtuale

Il disco `‘D:’` virtuale dovrebbe contenere alcune directory che riproducono in pratica il classico ambiente Dos:

- `‘D:\TEMP\’`
equivalente a `‘/var/emul/dos/temp/’`;
- `‘D:\DOS\’`
equivalente a `‘/var/emul/dos/dos/’`.

Per evitare la proliferazione di directory temporanee, è possibile utilizzare al posto di `/var/emul/dos/temp/` un collegamento simbolico che punti a `/tmp/`.

```
# ln -s /tmp /var/emul/dos/temp [Invio]
```

La configurazione di DOSEMU

«

La configurazione di DOSEMU consiste nella modifica dei file `/etc/dosemu.conf` e di `/etc/dosemu.users`. Il file `/etc/dosemu.users` permette di definire gli utenti che possono utilizzare DOSEMU, mentre l'altro stabilisce tutte le altre caratteristiche.

Purtroppo, la configurazione di DOSEMU, specialmente per ciò che riguarda il file `/etc/dosemu.conf`, è complessa e cambia da versione a versione. Inoltre, DOSEMU può costituire anche un problema per la sicurezza del sistema dal momento che di solito l'eseguibile `dos`, deve essere SUID-root (cioè deve appartenere a `root` e avere il bit SUID attivato) per utilizzare funzionalità particolari dell'hardware (soprattutto l'adattatore grafico VGA).²

File «`/etc/dosemu.users`»

«

DOSEMU permette di distinguere alcune categorie di utenti, attribuendogli privilegi differenti, in base a una diversa configurazione nel file `/etc/dosemu.conf`. Tali categorie di utenti dipendono quindi dalla configurazione di questo file.

Il file `/etc/dosemu.users` può contenere righe di commento, introdotte dal simbolo `#`, righe bianche o vuote, che vengono ignorate, e direttive espresse dalla sintassi seguente:

utente [*variabile_di_configurazione...*]

In pratica, si possono abbinare a un utente una o più variabili di configurazione che fanno riferimento a elementi del file `/etc/dosemu.conf`. È da osservare, in particolare, che si può indicare anche un utente particolare, `all`, per fare riferimento a tutti gli utenti a cui non si fa menzione in modo esplicito.

A titolo di compromesso, viene mostrato un esempio di configurazione del file `/etc/dosemu.users` che dovrebbe essere sufficiente nella maggior parte delle situazioni. Si tratta in pratica della versione standard distribuita assieme a DOSEMU, con l'aggiunta di qualche utente ipotetico.

```
# This is a sample /etc/dosemu.users file
# For more details look at ./doc/README.conf

root c_all          # root is allowed to do all weird things
nobody guest       # variable 'guest' is checked in /etc/dosemu.conf
                   # to allow only DEXE execution
guest guest        # login guest treated as 'nobody'

# Utenti inseriti normalmente
tizio
caio
sempronio

# If you want to allow limited dosemu to all users, uncomment the line below
#all restricted     # all other users have normal user restrictions
```

Come si intuisce, l'utente `root` ha tutti i diritti necessari a compiere quello che vuole dall'interno di DOSEMU. Sono previsti gli utenti `nobody` e `guest`, a cui sono concesse solo poche cose, mentre agli utenti `tizio`, `caio` e `sempronio` sono concessi privilegi normali. Infine, appare commentata la direttiva

'all restricted', con la quale si potrebbe consentire l'utilizzo di DOSEMU a tutti gli altri utenti, con privilegi ridotti.

File «/etc/dosemu.conf»

«

La preparazione di '/etc/dosemu.conf' è invece più delicata. Il file di esempio già fornito all'interno del pacchetto di distribuzione di DOSEMU è commentato molto dettagliatamente, però è anche molto complesso. Di seguito vengono indicate solo alcune parti particolarmente importanti. Le altre direttive di questo file, possono essere lasciate come sono, ignorandole, almeno fino a quando non si raggiunge una buona esperienza con l'uso di DOSEMU.

```
# Viene impostata la mappa della tastiera per uniformarsi alla
# disposizione dei tasti in Italia.
$_rawkeyboard = (1)      # bypass normal keyboard input, maybe dangerous
$_layout = "it"         # one of: finnish(-latin1), de(-latin1), be, it, us
                        # uk, dk(-latin1), keyb-no, no-latin1, dvorak, po
                        # sg(-latin1), fr(-latin1), sf(-latin1), es(-latin1)
                        # sw, hu(-latin2), hu-cwi, keyb-user
$_keybint = (on)        # emulate PCish keyboard interrupt

# Vengono definite le potenzialità dello schermo
# (per poter utilizzare la grafica, come impostato in questo
# esempio, occorre avviare il programma dos con i privilegi
# dell'utente root).
$_video = "vga"         # one of: plainvga, vga, ega, mda, mga, cga
$_console = (1)         # use 'console' video
$_graphics = (1)        # use the cards BIOS to set graphics
$_videoportaccess = (1) # allow videoportaccess when 'graphics' enabled
$_vbios_seg = (0xc000)  # set the address of your VBIOS (e.g. 0xe000)
$_vbios_size = (0x10000) # set the size of your BIOS (e.g. 0x8000)
$_vmemsize = (1024)     # size of regen buffer
$_chipset = ""          # one of: plainvga, trident, et4000, diamond, avance
                        # cirrus, matrox, wdvga, paradise
$_dualmon = (0)         # if you have one vga _plus_ one hgc (2 monitors)

# Viene definito l'uso dei dischetti e dell'immagine del disco C:.
```

```

$_floppy_a = "threeinch" # or "fiveinch" or empty, if not existing
$_floppy_b = ""         # ditto for B:

$_hdimage = "hdimage.first" # list of hdimages under /var/lib/dosemu
                          # assigned in this order such as
                          # "hdimage_c hdimage_d hdimage_e"
                          # If the name begins with '/dev/', then partition
                          # access is done instead of virtual hdimage such as
                          # "/dev/hda1" or "/dev/hda1:ro" for readonly
                          # Currently mounted devices and swap are refused.
                          # Hdimages and devices may be mixed such as
                          # "hdimage_c /dev/hda1 /dev/hda3:ro"
                          # Note: 'wholedisk' is _not_ supported.
$_hdimage_r = $_hdimage # hdimages for 'restricted access (if different)

# Viene definita la stampante.
$_printer = "lp"         # list of (/etc/printcap) printer names to appear as
                          # LPT1, LPT2, LPT3 (not all are needed, empty for none)
$_printer_timeout = (20) # idle time in seconds before spooling out

$_ports = ""            # list of portnumbers such as "0x1ce 0x1cf 0x238"
                          # or "0x1ce range 0x280,0x29f 310"
                          # or "range 0x1a0, (0x1a0+15)"

```

Installare e utilizzare il Dos

Il problema successivo è quello di riuscire a installare il Dos nel file-immagine che serve per effettuare l'avvio del Dos stesso. L'immagine in questione, che probabilmente è il file `/var/lib/dosemu/hdimage.first`, contiene già una serie di programmi Dos che fanno parte di DOSEMU e come tali non vanno cancellati. Ma l'immagine che viene distribuita così non è avviabile e il problema è proprio quello di inserirvi il kernel del Dos e l'interprete dei comandi `COMMAND.COM`, salvo il caso in cui sia già presente una versione di FreeDOS.

1. Preparazione di un dischetto di avvio

Per prima cosa occorre preparare un dischetto Dos avviabile che contenga qualche programma di servizio indispensabile. Da un elaboratore che stia eseguendo il sistema operativo Dos si procede come segue:

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

```
C:> COPY C:\DOS\SYS.* A: [Invio]
```

```
C:> COPY C:\DOS\FDISK.* A: [Invio]
```

Oltre a questi file converrebbe preparare nel dischetto un programma per la creazione e modifica di file di testo. Questo serve per preparare i file 'CONFIG.SYS' e 'AUTOEXEC.BAT'.

2. Avvio del dischetto attraverso DOSEMU

È necessario quindi avviare il Dos contenuto nel dischetto appena creato attraverso DOSEMU. Per fare questo, dall'elaboratore GNU/Linux si avvia DOSEMU nel modo seguente:

```
# dos -A [Invio]
```

Se tutto è andato bene si avvia il Dos; quindi, dopo la richiesta della data e dell'ora appare l'invito classico (il *prompt*), per l'inserimento dei comandi attraverso la shell ('**COMMAND.COM**').

```
A:\>
```

3. Trasferimento del sistema

Per trasferire nel file-immagine il sistema contenuto nel dischetto, in modo da rendere questa immagine avviabile, occorre procedere prima con la creazione di un MBR (*Master boot record*):

```
A:\> FDISK /MBR [Invio]
```

quindi con il trasferimento del sistema:

```
A:\> sys c: [Invio]
```

Se è andato tutto bene, adesso il disco ‘C:’, cioè l’immagine, è pronto.³

4. Controllo del disco C:

Il disco ‘C:’ dovrebbe contenere alcuni file di DOSEMU. Per verificare il contenuto è sufficiente spostarsi in ‘C:’.

```
A:\> c: [Invio]
```

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

5. Modifica di config.sys

Trovandosi in ‘C:’, potrebbe essere conveniente modificare i file ‘CONFIG.SYS’ e ‘AUTOEXEC.BAT’. Si inizia con ‘CONFIG.SYS’. Si stabilisce di poter utilizzare tutte le lettere di unità (*drive*) a disposizione.

```
LASTDRIVE=Z
```

Si definisce attraverso il driver ‘EMUFS.SYS’ di DOSEMU che la prossima lettera di disco a disposizione punti alla directory ‘/var/emul/dos/’. Di conseguenza, quella directory viene interpretata come disco ‘D:’

```
DEVICE=C:\EMUFS.SYS /var/emul/dos
```

Viene avviato il driver ‘EMS.SYS’ di DOSEMU che si occupa della gestione della memoria estesa.

```
DEVICE=C:\EMS.SYS
```

Se in seguito se ne presenta l’opportunità, è sempre possibile apportare modifiche a questo file.

6. Modifica di 'AUTOEXEC.BAT'

Inizialmente il file non necessita di modifiche. È possibile vedere in seguito come configurare al meglio questo file.

7. Conclusione dell'installazione

Per terminare la sessione di lavoro dell'installazione occorre fare terminare l'esecuzione di DOSEMU, avviato in precedenza con il comando '**dos -A**'. Per chiudere si utilizza il programma '**EXITEMU.COM**':

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

8. Verifica

Se tutto è andato come previsto, il Dos è pronto. Si può provare ad avviare il Dos senza l'uso del dischetto semplicemente con il comando:

```
$ dos [Invio]
```

Se ha funzionato, si ottiene l'invito normale:

```
C:\>
```

Per uscire si utilizza il programma '**EXITEMU.COM**':

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

I dischi virtuali con «LREDIR.COM»

«

Il programma '**LREDIR.COM**' è in grado di consentire l'accesso a porzioni del file system di GNU/Linux attribuendo una lettera di unità. Per esempio:

```
C:\> LREDIR X: \linux\fs\home [Invio]
```

fa sì che il disco 'X:' corrisponda al contenuto della directory '/home/'. Invece,

```
C:\> LREDIR Y: \linux\fs\${home} [Invio]
```

fa sì che il disco 'Y:' corrisponda al contenuto della directory personale dell'utente che sta usando DOSEMU.

Il mouse

Teoricamente, DOSEMU è in grado di gestire da solo il mouse. In pratica potrebbe non essere così. In tal caso conviene provare ad avviare un programma apposito all'interno del 'CONFIG.SYS' o di 'AUTOEXEC.BAT'.

Un esempio di «AUTOEXEC.BAT»

Nell'esempio seguente viene utilizzato un programma per la gestione del mouse estraneo a DOSEMU. Il disco 'D:' è stato definito implicitamente all'interno di 'CONFIG.SYS' attraverso 'DEVICE=C:\EMUFS.SYS /var/emul/dos'.

```
@echo off

LREDIR H: linux\fs\${home}
LREDIR R: linux\fs\mnt/cdrom

PROMPT=$p$g
PATH=c:\
PATH=%PATH%;D:\;D:\DOS

SET TEMP=D:\TEMP

D:

D:\DOS\MOUSE

ECHO "Questo è DOSEMU. Benvenuto!"
```

DOSEMU e le console virtuali

«

Quando viene avviato il Dos attraverso DOSEMU, questo opera nella console virtuale sulla quale ci si trova. Di solito, per passare da una console virtuale all'altra è sufficiente premere la combinazione [*Alt F1*] o [*Alt F2*]... Quando ci si trova su una console virtuale all'interno della quale sta funzionando il Dos, per passare a un'altra si agisce con la combinazione [*Ctrl Alt F1*] o [*Ctrl Alt F2*]...

DOSEMU da X

«

Per avviare il Dos in una finestra del sistema grafico X, conviene avviare DOSEMU attraverso '**xdos**' che normalmente è un collegamento simbolico a '**dos**'.

DOSEMU e Mtools

«

Nelle sezioni precedenti si è visto l'uso del file-immagine '`/var/lib/dosemu/hdimage`', che costituisce normalmente il disco '`C:`' per DOSEMU. Questo file non è gestibile con strumenti Unix normali, soprattutto perché non è un'immagine standard. Si tratta dell'immagine di un piccolo disco fisso contenente una partizione, con l'aggiunta di un'intestazione aggiuntiva.

Questo disco '`C:`' può essere utilizzato principalmente attraverso strumenti Dos all'interno di DOSEMU, così come è stato già mostrato, oppure può essere raggiunto anche tramite Mtools, purché configurato opportunamente. Infatti, è sufficiente informare Mtools sulla posizione esatta in cui ha inizio la prima partizione all'interno del file-immagine, per potervi accedere anche con questo strumento. Potrebbe trattarsi della direttiva seguente, nel file di configurazione '`/etc/mtools.conf`'.


```
drive n: file="/var/lib/dosemu/hdimage.first" partition=1 offset=128
```

In tal modo, per Mtools, il disco 'N:' corrisponderebbe al disco 'C:' di DOSEMU.

È importante fare attenzione al valore dello scostamento (*offset*) che potrebbe cambiare da una versione all'altra di DOSEMU.

Implicazioni sulla gestione dei permessi

Il Dos non è un sistema operativo multiutente e di conseguenza non è in grado di attribuire dei permessi ai file. Quando si utilizza il Dos all'interno di DOSEMU, i permessi vengono gestiti in modo predefinito. <<

Quando si crea un file gli vengono attribuiti i permessi predefiniti in base a quanto stabilito con la maschera dei permessi; inoltre, l'utente e il gruppo proprietario corrispondono all'utente che ha avviato DOSEMU e al gruppo cui questo utente appartiene.

Quando si accede a un file, l'apparenza delle caratteristiche di questo cambiano a seconda che l'accesso avvenga da parte di un utente rispetto a un altro: l'utente che ha creato il file può modificarlo, un altro potrebbe trovarlo protetto in sola lettura.

In particolare, i file contenuti nel file-immagine che costituisce il disco 'C:' hanno le proprietà e i permessi del file-immagine stesso.

Ma il Dos non è in grado di gestire tutte le finezze che può invece amministrare un sistema Unix, di conseguenza, quando si tenta di fare qualcosa che i permessi non consentono, si ottengono per lo più

delle segnalazioni di errore che normalmente non si vedono quando si usa il Dos da solo senza emulazioni.

Quando si utilizza il Dos con DOSEMU su un sistema al quale accede un solo utente, non dovrebbero porsi problemi: basta che l'unico utente utilizzi sempre lo stesso nominativo (lo stesso UID). Quando lo si utilizza invece in un sistema al quale accedono più utenti, è ragionevole desiderare che i dati personali possano essere inaccessibili agli altri; quindi, questo modo trasparente di gestire i permessi può essere solo positivo. Quando si vogliono gestire alcune attività in gruppo si può aggirare eventualmente l'ostacolo utilizzando un utente comune creato appositamente per quel compito.

Un'ultima annotazione deve essere fatta per i file eseguibili che non necessitano dei permessi di esecuzione, come invece richiederebbe GNU/Linux. È generalmente sufficiente che ci siano i permessi di lettura. A volte sono necessari anche quelli in scrittura, ma prima di dare questi permessi è meglio verificare, onde evitare di lasciare campo libero a un possibile virus.

¹ **DOSEMU** GNU GPL

² Se ci si accontenta di uno schermo a caratteri, senza grafica e senza cornici, non dovrebbe essere necessario attivare il bit SUID.

³ Il comando '**FDISK /MBR**' riguarda precisamente MS-Dos, mentre nel caso di cloni le cose potrebbero essere differenti; per esempio potrebbe essere necessario avviare il programma nel modo solito e poi specificare la richiesta selezionando una voce da un menù.