

AMIGhevole

Riepilogo dei comandi

■ *A conclusione del ciclo di articoli dedicati al sistema operativo di Amiga, l'AmigaDOS, questo mese vi proponiamo un articolo riassuntivo di tutti i comandi diretti (volendo, da staccare o fotocopiare e conservare) spiegandone brevemente il funzionamento e le opzioni più interessanti. Infine qualche trucchetto o curiosità riguardo all'AmigaDOS non guasterà di sicuro. Pronti?...* ■

di Andrea de Prisco

... Via!!!

Il primo set di comandi utente che mostreremo riguarda il trattamento dei file contenuti su disco. Fanno seguito a questi i comandi di sistema veri e propri che mostreremo nel secondo paragrafo.

Comando: DIR

Legge la directory dal disco mostrando prima i nomi delle subdirectory e poi i file, quest'ultimi in ordine alfabetico. È possibile indicare quale directory vogliamo vedere, comunque annidata e dovunque allocata (drive esterni, hard disk, ram disk). Esempi:

```
DIR
DIR RAM:
DIR DF1: FONT/OPAL
```

È possibile indicare una delle tre opzioni A, I, AI, (indicando di seguito ai comandi dir «OPT» e l'opzione scelta) ottenendo rispettivamente l'intero contenuto del dischetto (i nomi di tutti i file in tutte le directory), l'opzione interattiva, o entrambe. Con l'opzione interattiva Amiga mostra il nome di un file seguito da un punto interrogativo e si ferma. Battendo [Re-

turn] si procede passando a nuovo file, digitando «q» si esce dal comando. Se si tratta del nome di una directory digitando «e» possiamo «esplorarlo» (ciò avviene automaticamente con l'opzione AI), digitando «b» torniamo su di un livello, se si tratta di un file di caratteri possiamo vederne il contenuto con «t» (type) infine se vogliamo togliere un file basta digitare «del».

Comando: LIST

Mostra la directory del dischetto più altre informazioni relative ai file come le protezioni, data e ora dell'ultima scrittura, la dimensione, e un eventuale commento. Come per il comando DIR si può specificare a quale directory siamo interessati e con la chiave «S» possiamo indicare una sottostringa in modo da listare solo i nomi di file che la contengono. Esempi:

```
LIST
LIST DF1: C
LIST DF0: S txt
```

Comando: CD

Permette di esplorare l'organizzazione ad albero dei file settando una nuova directory come corrente. Si in-

dica di seguito al comando la directory (o il device logico) alla quale siamo interessati, oppure, indicando «/», risaliamo alla directory padre. Digitato senza parametri il comando restituisce il percorso sinora compiuto. Esempi:

```
CD DF1:
CD DF0:C
CD /
CD /LIBS
CD SYS:
CD
```

Comando: MAKEDIR

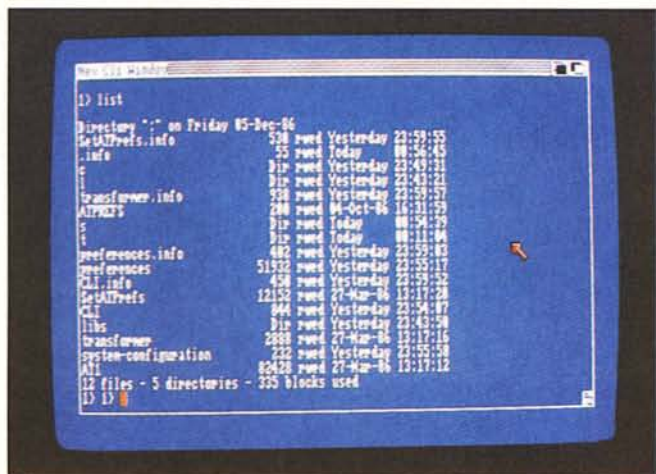
Permette di creare nuove directory. Si indica il nome eventualmente preceduto dal percorso nell'albero. Esempi:

```
MAKEDIR Comandi
MAKEDIR RAM: Pippo
MAKEDIR DF1: C/Comandi Ausiliari
```

Comando: TYPE

Si usa per vedere su video il contenuto di un file sotto forma di caratteri ASCII o, con OPT H, direttamente in codici esadecimali. L'opzione N mostra anche i numeri linea. Esempi:

```
TYPE filetesto
TYPE DF1: S/Startup-sequence
```

TYPE pippo OPT H

Comando: DELETE

Cancella un file o una directory (piena o vuota). Si indica il nome dell'oggetto da cancellare, nel caso di una directory è obbligatoria la chiave ALL e se non vogliamo vedere i nomi dei file mano mano cancellati aggiungiamo la chiave Q. Esempi:

```
DELETE nomefile
DELETE DF1: C ALL
DELETE FONTS/OPAL ALL Q
```

Comando: RENAME

Serve per rinominare un file (o una directory). Esistono varie forme sintattiche tutte semanticamente identiche. Esempi:

```
RENAME FROM OldName TO NewName
RENAME OldName AS NewName
RENAME TO NewName FROM OldName
```

Comando: COPY

Con questo comando possiamo copiare file, directory o interi dischetti (file per file). Come per DELETE esistono le chiavi ALL e Q (medesima semantica). La destinazione può essere un altro file (nome diverso) eventualmente locato in un'altra directory anche su un device differente (ram, hard disk, drive aggiuntivi). Esempi:

```
COPY NomeFile TO DF1:
COPY NomeFile TO NuovoFile
COPY NomeDir TO DF1: ALL Q
```

Comando: DISKCOPY

Permette la copia integrale di un dischetto non protetto, formattando contemporaneamente il disco destinazione. Ovviamente i dati eventualmente contenuti su questo andranno irrimediabilmente persi. È possibile dare un nuovo nome al disco copia. Esempi:

DISKCOPY DF0: TO DF1:
DISKCOPY DF0: TO DF0:
DISKCOPY DF0: DF1: NAME NomeDisco

Comando: FORMAT

È usato per formattare un disco vergine o uno già formattato di cui vogliamo cancellare tutti i file in esso contenuto. È obbligatorio dare un nome al disco. Esempi:

```
FORMAT DRIVE DF0: NAME
«NomeDisco»
FORMAT DRIVE DF1: NAME
«NomeDisco»
```

Comando: ED

Editor di schermo. È quasi un word processor, dispone di decine di comandi quindi non è possibile riassumerli in questa sede (MC n. 58).

Comando: EDIT

Editor di linea. Stesso discorso di ED (non riferirsi a MC n. 58... non è stato trattato).

Comando: FILENOTE

Permette di apporre un commento

Pattern Matching

Molti computer dispongono del noto meccanismo del pattern matching, la maggioranza degli utenti di queste macchine ne fanno uso molto intenso, anche se spesso non si conosce il suo nome. Per chi prima di «amigo» era 64-ista basta nominare la ben nota stringa LOAD «*», 8 in cui il pattern «*» match-ia con qualsiasi nome di file. Discorso analogo per l'MS-Dos dove comandi tipo DIR*, txt o similari vengono usati come il pane.

Amiga non fa eccezione anche se non permette di «allargarsi troppo» con le match-iate. Nella fattispecie questo meccanismo è possibile usarlo solo se il comando che stiamo impartendo, nel caso in cui il risultato del matching restituisca un insieme di file, continua ad essere valido. Ovvero, anche se dal matching risulta un solo file non possiamo utilizzarlo se il comando non lo prevede. Ad esempio possiamo cancellare più file indicando un pattern comune, ma non possiamo ad esempio impartire il comando RUN senza specificare per esteso a quale programma siamo interessati. Idem per ED, TYPE, RENAME ... e quasi tutti gli altri. Praticamente funziona solo con COPY, DELETE e LIST (quest'ultimo con la chiave P, pattern).

Al posto dell'universale asterisco, in AmigaDOS si utilizzano alcune combinazioni di caratteri speciali. Il punto interrogativo match-ia con un qualsiasi carat-

tere, il cancelletto # seguito da un pattern qualsiasi match-ia con nessuna o con un numero qualsiasi di occorrenze di questo. Combinando i due caratteri abbiamo l'equivalente dell'asterisco ovvero «un numero qualsiasi di caratteri qualsiasi» #?. Esiste inoltre il simbolo % che ha valore di stringa nulla, l'or rappresentato dalla barretta verticale e per raggruppare pattern si utilizzano le parentesi. Adesso qualche esempio per chiarire le idee:

DELETE #?.TXT

cancella tutti i file con estensione .TXT

DELETE file?

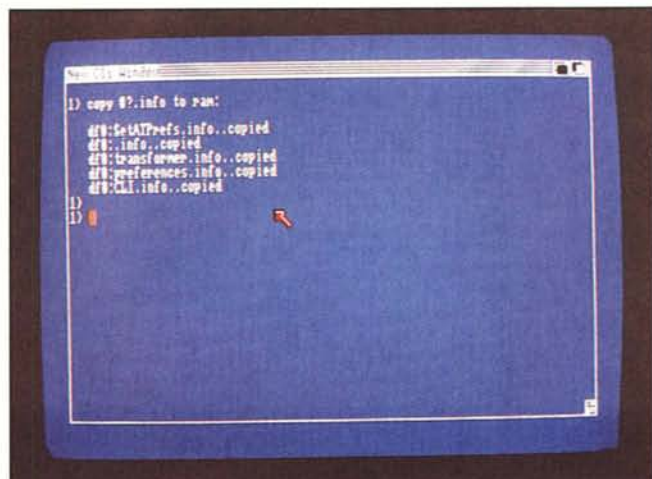
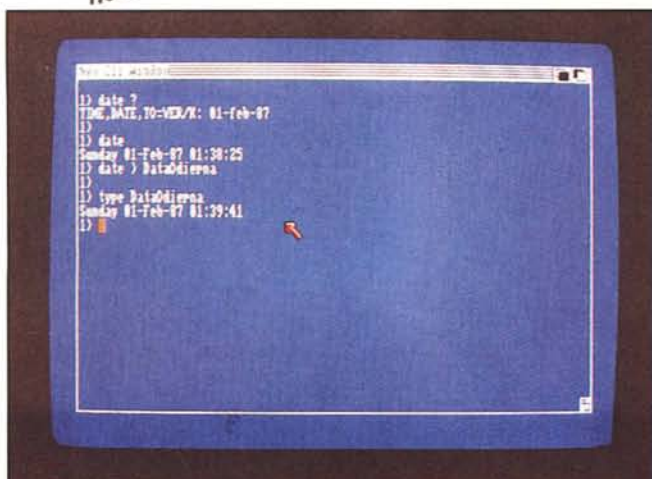
cancella tutti i file di nome «file» più un altro carattere (file1, file2, fileX ecc.)

DELETE pippo#(1:2)

cancella tutti i file il cui nome inizia per pippo e termina per una qualsiasi sequenza di 1 e di 2 (file1, file2, file12, file11212 ecc.)

LIST P prova(1:%)#a

lista tutti i file i cui nomi iniziano per «prova», possono avere o meno un 1 come sesto carattere (sempre del nome) e terminano per un numero imprecisato di a (prova, provaaaa, prova1, provaaaaa ecc.).



ad un file. Esso apparirà quando si richiede la directory tramite il comando LIST. Esempi:

```
FILENOTE Pippo COMMENT
"bla-bla-bla"
```

Comando: PROTECT

È usato per proteggere o sprotteggere file in lettura, scrittura, esecuzione e cancellamento. Si indica di seguito al comando il nome del file in questione e quali diritti vogliamo liberalizzare. Esempi:

```
PROTECT NomeFile r
PROTECT NomeFile rwe
PROTECT NomeFile
PROTECT NomeFile rwd
```

Comando: JOIN

Si usa per concatenare più file in un unico file destinazione. Esempi:

```
JOIN File1 File2 AS NewFile
JOIN File1 File2 File3 AS NewFile
```

Comando: SORT

Esegue un sort crescente (seguendo l'ordinamento ASCII) di un file di registrazioni separate da chr (13), creando un nuovo file. È possibile, grazie alla chiave COLSTART, indicare a partire da quale carattere di ogni registrazione effettuare il confronto. Veda anche comando STACK. Esempi:

```
SORT MioFile TO Pippo
SORT MioFile TO Pello COLSTART 5
```

Comando: SEARCH

Permette di ricercare in una directory o in tutto il dischetto i file che contengono la stringa indicata. È possibile inoltre includere o escludere i contenuti di subdirectory della directory indicata. Esempi:

```
SEARCH FROM Documenti «lettera»
```

```
SEARCH FROM Documenti «lettera» ALL
SEARCH FROM DF0: «gatto» ALL
```

Comandi di sistema

A questo insieme appartengono i comandi che controllano l'interfaccia CLI ed altre operazioni riguardo i dischetti.

Comando: NEWCLI

Serve per aprire una nuova finestra contenente una ulteriore interfaccia CLI. Tramite tale comando (non necessita di parametri, tranne il caso in cui non vogliamo redigere il nuovo CLI verso un terminale esterno oppure vogliamo aprire una finestra di dimensioni diverse o marchiata in un modo particolare) è possibile lanciare in una finestra un programma ed avere disponibile l'altra (o le altre) per continuare ad eseguire comandi CLI.

Comando: ENDCLI

Complementare al comando prece-

dente, esso serve per terminare una sezione CLI e chiudere la relativa finestra. Anche ENDCLI non necessita di parametri (salvo casi particolari, come sopra).

Comando: RUN

Seguito dal nome di un programma, permette di lanciare, parallelamente all'interfaccia CLI disponibile in quel momento, il programma in questione senza aprire ulteriori finestre. Esempi:

```
RUN Textcraft
RUN C/FORMAT DRIVE DF1:
NAME «pippo»
```

Comando: STACK

Ogni comando mandato in esecuzione necessita di una determinata quantità di memoria da utilizzare come area di lavoro. Normalmente tale area somma a 4000 byte ed è necessario aumentarla quando si eseguono operazioni che richiedono aree di lavoro molto estese. Ad esempio se dob-

DATE?

Lo strano titolo con cui è marchiato questo riquadro riguarda un truccetto per inserire in un command file (cfr. MC n. 60) la richiesta di immissione data, per aggiornare l'orologio calendario interno che come si sa non dispone di batteria tampone, eventualmente salvandola su disco in modo da lasciare traccia dell'ultimo «collegamento». Come spiegato nel numero scorso, un comando seguito dal punto interrogativo mostra su video l'argument template di quel comando e permette l'inserimento dei dati mancanti senza riscriverlo. Nella fattispecie, se da tastiera digitiamo:

```
DATE
```

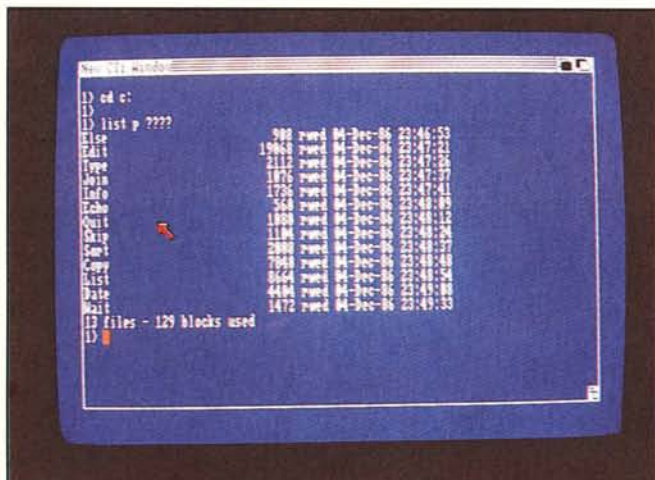
il sistema risponde:

```
DATE, TIME, TO=VER/K:
```

e aspetta col cursore pronto l'immissione della data, dell'ora o di entrambe. Se inseriamo tale «DATE?» in un command file, quando lo manderemo in esecuzione (specialmente se tale command file è la startup-sequence del dischetto) Amiga arrivato al comando arresterà automaticamente l'esecuzione per permetterci di inserire la data. Se infine vogliamo, come detto prima, immortalare il dischetto con la nuova data, di seguito al comando precedente nel command file adopereremo l'operatore di redirectione ">" (vedi l'apposito riquadro) nel seguente modo:

```
DATE > DataOdierna
```

dove DataOdierna è un file sul quale viene scritta ora e data appena impostata.



biamo fare un sort di un file più lungo di 200 registrazioni (approssimativo). Per aumentare lo stack è sufficiente indicare di seguito al comando la dimensione in byte. Esempi:

```
STACK 8000
STACK 20000
```

Comando: STATUS

Questo comando lista su video un determinato o tutti i processi CLI e per ognuno di questo indica una serie di informazioni riguardanti lo stack, la priorità, la dimensione del global vector e il nome delle sezioni nella segment list. Esempi:

```
STATUS FULL
STATUS 3 FULL
```

Comando: WHY

Al verificarsi di un errore, col comando WHY (senza parametri) è possibile ottenere chiarimenti sulla situazione.

Comando: FAULT

Se il sistema restituisce il numero di un errore, col comando FAULT è possibile farsi dire dal sistema tale codice a quale errore corrisponde. Esempi:

```
FAULT 103
FAULT 103 120 121
```

Comando: PROMPT

Con questo comando è possibile cambiare il prompt del CLI attivo in quel momento. La stringa «%N» viene interpretata come numero di CLI.

```
PROMPT «ready.»
PROMPT «Dica-Dica!»
PROMPT «%N>»
```

Comando: BREAK

Si usa per inviare un control C, D, E o F ad un determinato task. È anche

possibile mandarli tutt'e quattro con la chiave ALL. Esempi:

```
BREAK 3 C
BREAK 1 D E
BREAK 5 ALL
```

Comando: ASSIGN

Si usa per assegnare device logici a directory. Usato senza indicare la directory scollega il device logico. Esempi:

```
ASSIGN C: RAM: C
ASSIGN FONTS: DF1: FONTS
ASSIGN S:
```

Comando: DATE

Permette di modificare o di leggere l'ora e la data dell'orologio di sistema. È possibile usare diverse forme per aggiornare la data (se sono passati pochi giorni dall'ultimo aggiornamento).

Esempi:

```
DATE 22-jan-87
DATE 22-jan-87 23:04:55
DATE 23:04:55
DATE TOMORROW
DATE MONDAY
DATE TODAY
```

Comando: INFO

Mostra alcune informazioni relative ai drive collegati (RAMdisk compresa). Non necessita di parametri.

Comando: INSTALL

È usato per rendere boot-abile un dischetto formattato. Si ricorda che, una volta effettuata tale operazione, è necessario eseguirne altre per rendere il dischetto utilizzabile a tutti gli effetti (cfr. MC n. 60). Esempi:

```
INSTALL DF0:
INSTALL DF1:
```

Comando: RELABEL

Permette di cambiare da CLI il nome del dischetto inserito in uno dei drive. Esempi:

```
RELABEL DF0: NAME «Pippo»
RELABEL DF1: NAME «Pello»
```

Comando: EXECUTE

Con questo comando è possibile mandare in esecuzione un command file contenuto nel device logico S: oppure nella directory corrente. Esempi:

```
EXECUTE files/NomeFile
EXECUTE Startup-sequence
```



Operatori di redirectione

Per reindirizzare l'input o l'output di programmi da o verso periferiche diverse dalla tastiera e dal video, in AmigaDOS è possibile utilizzare i cosiddetti «redirection operator» '<' e '>'. Se ad esempio desideriamo una stampa su carta della directory del dischetto contenuto nel drive digiteremo da tastiera il comando:

```
DIR > PRT:
```

Analogamente per effettuare la stampa di un file di caratteri, utilizzando il comando TYPE:

```
TYPE > PRT: NomeFile
```

Possiamo indicare come destinatario anche un file su disco (esistente o meno, nel qual caso viene automaticamente creato) ad esempio per memorizzare la lista dei file:

```
LIST > NomeFile
```

Se, infine, disponiamo di un programma che chiede degli input qualsiasi da tastiera, possiamo preparare un file con tutti i dati e poi mandare in esecuzione il programma così:

```
NomeProgramma < FileData
```

Tanto per fare un esempio coi comandi AmigaDOS (che sono programmi) possiamo utilizzare l'opzione interattiva del comando DIR, che come input da tastiera prende i vari «b», «t», «e», «del» e «q», e preparare un file su disco con una manciata di queste possibilità, tutte ovviamente separate da un [return]. Se tale file si chiama Pippo, per vedere la nostra directory «programmata» scriveremo:

```
DIR < Pippo OPT I
```