

# Devices and Logical Devices

Il sistema operativo di Amiga, l'AmigaDOS, oltre ad avere una discreta gestione dei device fisici come stampante, porta seriale, porta parallela, drive interno ed esterno, hard disk, tastiera e video, permette di gestire dei device logici di input/output atti a facilitare la stesura di programmi nonché ad aumentare la flessibilità di tutto il sistema. In questo articolo ci occuperemo per l'appunto di questo tema, incentrato sul comando AmigaDOS Assign, col quale è possibile scollegare, collegare o creare nuovi device logici. ■

di Andrea de Prisco

## Device fisici

Molti si saranno chiesti come fare per azionare da CLI la stampante e ottenere così output su carta. Esistono un paio di procedimenti: essenzialmente dipende da ciò che vogliamo stampare e dal comando AmigaDOS usato. Un primo metodo consiste nell'usare i cosiddetti operatori di redirectione, «<» e «>», nel nostro caso quello di output «>» e specificare come periferica di uscita la stampante. Il tutto funziona se quanto stiamo digitando provoca effettivamente un output. Ad esempio, se digitiamo DIR, abbiamo l'effetto di veder scorrere su video la directory corrente. Se specifichiamo:

```
DIR > PRT:
```

(mi raccomando i «duepunti») l'output del comando DIR non avverrà su video ma sul device fisico PRT: che rappresenta la stampante.

Se, di contro, vogliamo stampare un file di testo, abbiamo due scelte, anche se forzate dal tipo di comando che useremo. Possiamo infatti sia dare il comando TYPE che il comando COPY. Il primo con l'operatore di redirectione, il secondo semplicemente specificando come periferica di uscita la stampante. Quindi, supponendo che il file da stampare si chiami Pippo, scriveremo alternativamente:

```
TYPE > PRT: Pippo
```

oppure

```
COPY Pippo TO PRT:
```

Si noti che l'operatore di redirectione deve stare «appiccicato» al comando e non dopo la specifica del file come si sarebbe tentati a fare (TYPE Pippo > PRT:).

Un'altra importante periferica, anche se spesso non considerata tale, è l'unità video, identificata dal simbolo\*

(asterisco). Ad esempio possiamo copiare il contenuto di un file su video con:

```
COPY NomeFile TO*
```

(perfettamente equivalente a un TYPE NomeFile) o, più interessante, fare l'operazione inversa: da video a file, senza scomodare ED. Scriveremo ad esempio:

```
COPY*TO NomeFile
```

per ottenere l'effetto di copiare nel file specificato tutto quello che digiteremo dopo aver impartito il comando di sopra, fino alla pressione dei tasti CONTROL\ (left slash, accanto al backspace) che chiude il file e fa tornare al prompt del CLI.

A proposito di COPY, questo comando, qualora non fosse evidente, permette anche di fare copie di file sullo stesso o su altri dischetti. Nel caso di due drive, ad esempio, per copiare un file di nome pippo da quello interno a quello esterno, digiteremo:

```
COPY DF0: pippo TO DF1:
```

omettendo eventualmente DF0: se questo è il drive corrente. Se disponiamo di un Hard Disk, per accedervi useremo il device DH0:.

Sempre a proposito della periferica\* un'altra utile applicazione potrebbe essere la visualizzazione in ordine alfabetico di un file non ordinato presente sul disco. Se non vogliamo creare uno ex novo dato che non ci interessa conservarlo possiamo usare il comando SORT nel seguente modo:

```
SORT NomeFile TO*
```

che come detto usa il video per scaricare il file ordinato (ricordiamo che i vari record del file in questione devono essere separati da un carriage return, ascii 13).

Porta seriale e porta parallela costituiscono anch'esse due device di Amiga: rispettivamente SER: e PAR:.

Analogamente a quanto fatto per la

stampante e i dischi, per trasferire un file all'esterno di Amiga tramite le interfacce parallela o seriale si utilizza o il comando COPY o gli operatori di redirectione. Ad esempio:

```
COPY NomeFile TO PAR:
```

oppure

```
TYPE > SER: NomeFile
```

## Device quasi fisico

In Amiga esiste un altro device molto importante: la ram, che può essere gestita proprio come una unità a dischi, con l'unica differenza che (ovviamente) allo spegnimento del computer si perde tutta l'informazione in essa contenuta. Come prevedibile, il device si chiama RAM:, e come sempre (questo è il bello di Amiga) l'accesso avviene per comando COPY o per redirectione di output. Importante è segnalare che avendo a che fare con un ram disk vero e proprio, le redirectioni devono avvenire non sulla periferica ram e basta, ma su un suo file, anche non esistente (nel qualcaso sarà automaticamente creato).

Ad esempio, se vogliamo registrare la directory attuale su ram non scriveremo semplicemente DIR > RAM: ma dovremo indicare un nome con cui identificare i caratteri trasferiti in ram (il nome del file). Possiamo chiamare il file pippo (tanto per cambiare) nel qualcaso scriveremo:

```
DIR > RAM: pippo
```

e per leggerlo successivamente possiamo digitare:

```
TYPE RAM: pippo
```

Come dicevamo, il ram disk si comporta esattamente come un drive e sono possibili su questo tutte le normali operazioni Dos. Possiamo ad esempio copiare file da ram a disco:

```
COPY RAM:NomeFile TO DF0:
```

da disco a ram:



```
AmigaDOS
1) assign c: ram:comandi
1)
1) assign
Volumes:
Workbench [Mounted]
cli [Mounted]
Directories:
c          Volume: RAM
s          Volume: cli Dir: s
l          Volume: cli Dir: l
L          Volume: cli Dir: fonts
FONTS     Volume: cli Dir: devs
DEVS      Volume: cli Dir: libs
LIBS      Volume: cli Dir: cli
SYS
Devices:
DF1 DF0 PRT PAR SER
RAM COM RAM
1)
```

Comando ASSIGN

```
AmigaDOS
1) mkdir ram:comandi
1)
1) dir ram:
comandi (dir)
1)
```

Costruiamo una directory COMANDI in RAM:

COPY DF0:NomeFile TO RAM:  
 creare directory in ram:  
 MAKEDIR RAM:NomeDirectory

così come esplorare la ram disk col comando CD.

A fine articolo vi mostreremo come rendere residente in ram i comandi più utili dell'AmigaDOS. Così facendo, nell'ipotesi economica di non disporre del drive esterno, non saremmo costretti a infilare continuamente il workbench nel drive ogni volta che dobbiamo usare comandi DOS su un disco diverso da quello di sistema.

### Device logici

Se prendiamo il disco workbench fornito con la macchina (o meglio, una sua copia) e cancelliamo da questo tutti i file e le directory in esso contenuti e visibili da «scrivania» (clock, preference, demo, utility, system, trashcan, empty) ci accorgeremo che il dischetto è tutt'altro che vuoto. Diverse centinaia di blocchi sono ancora occupati, tant'è che se in tale condizione digitiamo da CLI un bel DIR OPT A (che mostra tutti i file contenuti sul dischetto) vedremo una sfilza ancora interminabile di roba scorrere sul nostro video. Si tratta del «necessaire» del sistema operativo, usato al momento del boot e durante l'uso dell'interfaccia CLI.

Tutta la roba contenuta sul dischetto non è buttata lì alla rinfusa, ma intelligentemente raccolta in directory e subdirectory in maniera assai strutturata. Ad esempio avremo la directory C che contiene tutti i comandi; la directory S contenente tutti i command file eseguibili con EXECUTE (argomento del prossimo numero, ndr), la directory

dei font di caratteri, quella DEVS dei device, e delle librerie di sistema caricate in ram, (LIBS) o utilizzate a run time (L). Con uguale nome, ma (naturalmente) seguito dai due punti, per ognuna di queste directory l'AmigaDOS mette a disposizione un device logico inizializzato all'omonima directory. Ad esempio esisterà il device C: che punta alla directory C così come per S: per S e così via.

Se digitiamo un comando qualsiasi, il sistema va prima a cercarlo nella directory corrente, e se non lo trova, prova a cercarlo nella directory puntata dal device C:.. Analogamente per gli altri device: quando il sistema deve accedere ad una delle directory sopra indicate, non vi accede direttamente, ma in un certo senso «passa» per il device logico corrispondente. Ed è bene che chiunque scriva un programma faccia lo stesso: se ne guadagna in flessibilità.

Questo perché se riassegnamo un device a qualcos'altro, e poi mandiamo in esecuzione un programma che fa riferimento a questo, automaticamente questo «vedrà» ciò che noi vogliamo che «veda». Un caso classico sono i programmi che non prevedono l'accesso ad altri dischi se non quello di sistema, sul quale è presente il programma stesso. Esso fa riferimento a device SYS: ? Bene, col comando ASSIGN assegniamo a questo device un altro disco e il gioco è fatto. Da questo momento in poi il programma accederà al disco che vogliamo noi. Se, di contro, il programma è scritto in maniera meno pulita e al posto di accessi a SYS: fa riferimento al nome col quale è stato formattato il dischetto non potremo far nulla.

Altro esempio tipico: i font di caratteri. Se un programma usa solo i font

del suo dischetto, ma riferisce al device logico corrispondente, possiamo riassegnare questo ad una altra directory (ad esempio su un altro disco) ed avere il programma del disco A con i font presenti nel disco B.

Per assegnare device a directory è, come già detto, disponibile il comando ASSIGN con sintassi:

ASSIGN NomeDevice: NomeDirectory

Ad esempio, per assegnare al device logico FONTS: la corrispondente directory presente non su disco sistema ma sul disco nel drive esterno, digiteremo;

ASSIGN FONTS: DF1: FONTS

Per ripristinare il tutto, faremo macchina indietro con:

ASSIGN FONTS: DF0:FONTS

Analogamente per assegnare come disco sistema il dischetto contenuto nel drive esterno:

ASSIGN SYS: DF1:

Oltre a ciò, il comando ASSIGN digitato senza alcun seguito, permette di vedere tutti gli assegnamenti finora effettuati. Per «scollegare» un device, ovvero per farlo sparire e basta, è sufficiente digitare solo:

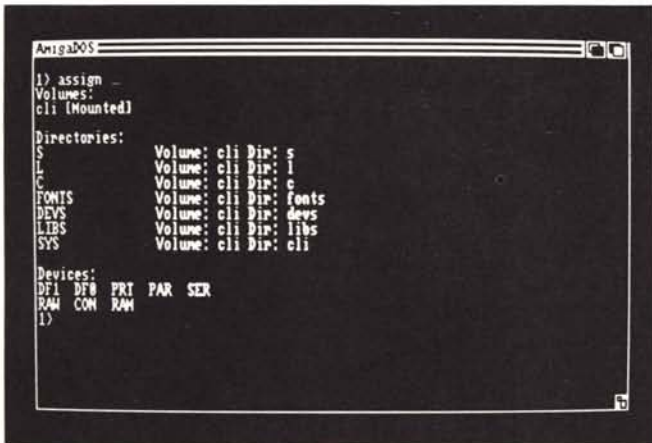
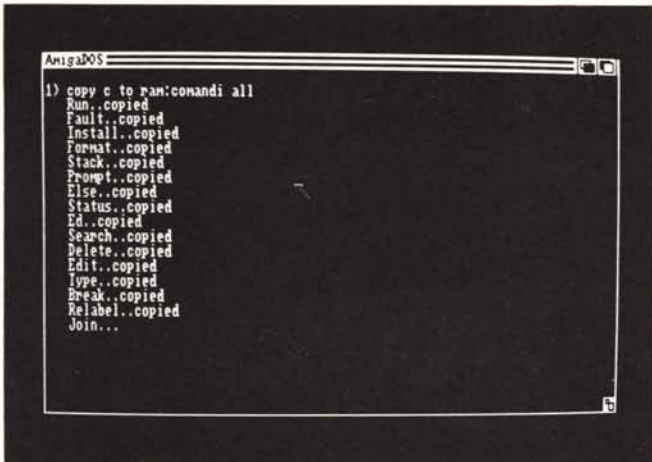
ASSIGN NomeDevice:

In tutti i casi, le modifiche non resteranno permanenti, ossia a nuovo boot o re-boot tutti i device saranno reinizializzati ai valori standard.

### Comandi in RAM

Come abbiamo più volte ripetuto nelle pagine di Amighevole, il sistema operativo di Amiga non è interamente





**Riassunto dei comandi AmigaDOS finora mostrati**

Comando	Sintassi	Descrizione
DIR	DIR DIR path	Legge la directory corrente Legge la directory indicata da path
CD	CD path CD /	Passa alla direct. indicata da path Torna alla directory padre Mostra il percorso finora compiuto
RUN	RUN NomeProgramma	Manda in esecuzione, parallelamente al CLI, il programma NomeProgramma
END-CLI	ENDCLI	Termina la sessione CLI nella quale è impartito tale comando
DELETE	DELETE NomeFile DELETE NomeDir ALL	Cancella dal disco il file NomeFile Cancella dal disco tutta la direct. di nome NomeDir
RENAME	RENAME Nome1 AS Nome2 RENAME Nome1 TO Nome2 RENAME FROM Nome1 TO Nome2	Cambia il nome al file Nome1 in Nome2. N.B.: 3 forme semanticamente uguali
PROMPT	PROMPT stringa	mostra "stringa" ad ogni prompt
SAY	SAY SAY stringa	Forma interattiva Recita la stringa indicata dopo SAY
WHY	WHY	Motiva una condizione d'errore
FAULT	FAULT n	Mostra l'errore corrispondente al numero n
TYPE	TYPE NomeFile	Mostra su video il contenuto di un file
DATE	DATE HH.MM.SS DATE GG-MM-AA	Setta data e ora dell'orologio interno
FILENOTE	FILENOTE File COMMENT str.	Aggiunge un commento a un file
FORMAT	FORMAT DRIVE dfr: NAME "X"	Formatta il dischetto presente nel drive n dandogli nome "X"
JOIN	JOIN ListaFile To NomeFile	Congiunge i file indicati creandone uno nuovo di nome NomeFile
SEARCH	SEARCH FROM File "stringa" SEARCH dfr: "stringa" ALL	Ricerca nel file indicato la stringa Ricerca nel disco n la stringa
SORT	SORT File TO NomeFile	Riordina File creando un file ordinato di nome NomeFile
COPY	COPY NomeFile To Nomefile	Esegue copia di file
MAKEDIR	MAKEDIR NomeDir	Crea una nuova directory
ASSIGN	ASSIGN Device: Directory	Assegna al device logico indicato la directory indicata
N.B.: NomeFile, NomeProgramma, NomeDir comprendono un eventuale path		

Copiamo tutti i comandi in RAM: (foto a sinistra in alto)  
Assegniamo al device C: la nuova directory (foto a sinistra in basso)  
I comandi ora in RAM

residente in memoria centrale ma, per quanto riguarda i comandi eseguibili da CLI, essi stanno tutti su disco, dove il sistema va a cercarsi ogni volta che l'utente ne usa qualcuno. Ovvero, quando noi digitiamo DIR e battiamo [Return] per ottenere su video la directory del dischetto, il sistema operativo va prima a prelevare il codice eseguibile del comando DIR e poi lo esegue mostrandoci la lista dei file. A parte l'ovvia perdita di tempo che si verifica per trovare ogni volta il comando su disco, il grosso svantaggio di tale scelta da parte della Commodore è il fatto che lavorare con un solo drive e due dischi risulta essere davvero oneroso. Ogni volta che vogliamo eseguire un comando sul disco non di sistema dobbiamo prima introdurre questo, digitare il comando facendo riferimento all'altro disco, attendere che il sistema legga il codice eseguibile del comando, scambiare i dischi e... ottenere quanto voluto. Come dire che dopo un po', a furia di togli e metti comincia a dolarci polso e pollice per quanti scambi di disco abbiamo effettuato.

Fortunatamente (e da Amiga non

potevamo aspettarci di meno) il modo per aggirare l'ostacolo esiste.... e come!

Grazie infatti alla flessibilità del sistema operativo con i soli comandi mostrati in questo articolo possiamo implementare un sistema a comandi in ram, in modo da non dover effettuare più scambi di disco, qualsiasi operazione si cerchi di fare. Basta implementare il device logico dei comandi in ram e il gioco è fatto: vediamo come.

Col dischetto di sistema inserito nel drive, la prima cosa che faremo è creare in ram una directory dandogli un nome qualsiasi, ad esempio Comandi. Scriveremo:

```
MAKEDIR RAM: Comandi
```

Provando ora a digitare DIR RAM: vedremo il contenuto del ram disk di Amiga, ora contenente la sola directory Comandi, vuota. La seconda operazione sarà quella di copiare l'intera directory C del dischetto nella directory Comandi in ram. Ricordiamo che C contiene tutti i comandi di CLI. Scriveremo:

**COPY C TO RAM: Comandi ALL**

l'opzione ALL indica che dobbiamo copiare tutto il contenuto di C. Per controllare che tutto è andato a buon fine, al termine della sopraindicata operazione, proviamo a digitare DIR RAM: Comandi. Dovremo veder scorrere su video la lista di tutti i comandi Amiga ora in ram.

Infine (si noti che in tutto sono necessarie tre sole operazioni) assegniamo al device logico dei comandi C: (che fino a questo momento puntava alla directory C del disco sistema) la Directory Comandi in ram. Scriveremo dunque:

```
ASSIGN C: RAM: Comandi
```

e abbiamo finito. Da questo momento in poi ogni comando che digiteremo non sarà caricato da disco ma da ram disk con l'ovvio vantaggio di una velocità molto elevata, ma soprattutto senza mai più richiedere il disco sistema quando dobbiamo lavorare su di un altro disco. Provare per credere: inserite un disco qualsiasi e provate a digitare un comando. Arrivederci.