

# i trucchi dell' MS-DOS

a cura di Pierluigi Panunzi

## Il Sistema Operativo MS-DOS

### I comandi di gestione dei file (seconda parte)

La scorsa puntata abbiamo parlato di tre dei quattro comandi appartenenti a questo gruppo («del», «ren» e «type») mentre in questa parleremo dell'ultimo rimasto («copy»): nella tabella 1 abbiamo ancora una volta riportato l'elenco completo dei comandi interni dell' MS-DOS, suddivisi a seconda della funzione svolta.

#### Il comando «copy»

Si tratta di un comando molto usato e presente in praticamente tutti i sistemi operativi, in quanto è molto sentita l'esigenza presso i programmatori di avere più copie di un certo file: ad esempio, dovendo mettere mano pesantemente su di un file di testo è sempre buona norma averne una copia, come suol dirsi, di «backup» con la quale si possa in ogni momento tornare sui propri passi, dopo però aver effettuato molte correzioni al file originale.

Questo è ciò che accade automaticamente quando elaboriamo un testo con un word-processor (ad esempio il WordStar) il quale di norma crea una copia del file originario denominandola «filename.bak»: se si ha un qualche ripensamento basta semplicemente cancellare la nuova versione del file e rinominare il «.bak» per ripristinare dunque il nostro file.

Comunque il comando in esame permette, nel caso dell' MS-DOS, oltre che a copiare un certo file, anche di «appendere» uno o più file ad un altro, per creare dunque un unico file come concatenazione di più file.

Cominciamo dunque dalla funzione primaria del comando «copy».

#### Il comando «copy» per copiar file

In questo caso la sintassi del comando prevede un certo numero di parametri opzionali, mentre ovviamente deve essere sempre presente il nome del file sorgente.

Il comando completo è il seguente:

```
copy [<drive:>] <pathname> [<drive:>] [<pathname>] [/v] [/a] [b/]
```

dove «drive:» e «pathname» sono come sempre rispettivamente l'indicazione dell'unità logica (floppy-disk, hard-disk o ram-disk) in cui risiede il file ed il nome del file stesso, completo del proprio «path».

A seconda o meno che sia presente il «pathname» del file destinazione oppure l'indicazione di un'unità logica, allora si hanno comportamenti completamente differenti: in particolare esaminiamo una per una tutte le possibili combinazioni.

Con il comando

```
copy pippo.src
```

oppure (trovandoci nell'unità «A:») il comando

```
copy pippo.src a:
```

in particolare otteniamo subito una segnalazione d'errore da parte dell' MS-DOS in quanto non è possibile ricopiare su se stesso un certo file: a seconda della versione in possesso si può ottenere il messaggio

```
File cannot be copied onto itself
O File(s) copied
```

oppure la nostrana (ma più bruttina...)

```
File non può essere copiato su se stesso
O file copiato(i)
```

A parte dunque questo caso anomalo, vediamo gli altri con il comando:

```
copy pippo.src b:
```

supponendo di trovarci in «a:», otteniamo una copia del file «pippo.src» sull'unità «b:», chiamata con lo stesso nome dell'originale.

La stessa cosa si ottiene se, stando in «b:», scriviamo

```
copy a:pippo.src
```

Con comandi del tipo

```
copy pippo.src pluto.src
```

e

```
copy pippo.src b: pluto.src
```

stando in «a:» otteniamo rispettivamente in «a:» stesso ed in «b:» una copia del file di partenza, in entrambi i casi chiamata «pluto.src».

In questi casi in cui si rispetta più che la sintassi la «logica» del comando «copy», succede un fatto che in alcuni casi può non risultare gradito: se il file «destinazione», che appare o meno nella linea di comando, è già presente nella directory e nella unità logica eventualmente specificata, allora il «nuovo» file destinazione va a ricoprire, sostituire il file già esistente, senza che di ciò venga avvertito il programmatore, magari con un messaggio «sgrammaticato».

In qualsiasi caso infatti la copia avviene senza ulteriori indugi, fatto confermato dal messaggio che dice che il file è stato copiato.

Quindi, morale della favola è che anche una semplicissima operazione di copiatura di un file richiede in alcuni casi una certa attenzione, pena la distruzione di file ovviamente importantissimi per una semplice dimenti-

canza: niente paura! Le «Norton Utilities» sono lì pronte a venirci in aiuto...

Siamo dunque arrivati a parlare degli switch «/v», «/a» e «/b»: il primo permette al programmatore di far effettuare l'operazione di copiatura all'MS-DOS con la «verifica» settore per settore.

Anche se in generale dovrebbe essere alquanto remota la probabilità che una copia avvenga in maniera errata, esiste pur sempre la possibilità di forzare la verifica dell'operazione, a scapito però del tempo dell'esecuzione del comando stesso, tempo che ora si allungherà di molto: nel caso sfortunato di errore durante la copia il fatto verrà segnalato con un apposito messaggio d'errore.

Invece gli switch «/a» e «/b» significano che il file su cui si sta compiendo l'operazione è rispettivamente di tipo «ASCII» oppure di tipo «binario»: in particolare «/a» fa sì che il file venga considerato a tutti gli effetti un file «ASCII» e cioè di testo e perciò verrà copiato solo fino al primo segnalatore di «end-of-file» (escluso), in generale rappresentato dal carattere «Control-Z» equivalente al valore esadecimale 1AH. In questo caso tutto quanto sia eventualmente presente «dopo» l'end-of-file viene completamente ignorato.

Invece lo switch «/b» consente la copia integrale del file indipendentemente dall'eventuale presenza di byte esadecimali 1AH, che in questo caso sono da considerarsi tutto meno che indicatori di end-of-file, ma istruzioni o dati di un programma espresso in linguaggio macchina.

Vediamo dunque l'effetto dell'applicazione dei due switch appena visti su alcuni esempi:

Con il comando

```
copy testo.txt /a b: lettera.txt
```

otterremo nell'unità «b:» il file «lettera.txt» come copia del file «testo.txt», però fino al segnalatore di «end-of-file», ottenendo in generale un file leggermente più corto dell'originario.

Per curiosità abbiamo copiato il file «ws.com» (la parte principale del WordStar con cui stiamo scrivendo il presente articolo) con l'opzione «/a»: in questo caso si è ottenuto un file di appena 288 byte a differenza dell'originario che era di ben 25600 byte!! In questo caso la copia è stata troncata alla prima occorrenza di un byte pari ad 1AH.

Con il comando

```
copy testo.txt /b b: lettera.txt
```

si ottiene invece la copia integrale del

file «testo» nel file «lettera», dal primo all'ultimo settore: stessa cosa succede anche con il file «ws.com».

In particolare in questo caso la funzione svolta dallo switch «/b» non è altro che quella di default per il comando «copy».

Con il comando

```
copy testo.txt b: lettera.txt /a
```

si ottiene la copia del file originario con l'aggiunta ulteriore del marker di end-of-file.

In questo caso copiando il solito «ws.com» si otterrà un nuovo file stavolta di 25601 byte, come era previsto!

Con il comando

```
copy testo.txt b: lettera.txt /b
```

si ottiene ancora una volta la copia integrale del file sorgente, stavolta senza l'aggiunta di un segnalatore di end-of-file alla fine del file destinazione.

### Il comando «copy» per appendere dei file

Il comando «copy», come già accennato, serve anche per «appendere» uno o più file in coda ad un altro oppure a concatenare due o più file per crearne uno nuovo.

La sintassi in questo caso è la seguente

```
copy <pathname> + <pathname> ... <pathname>
```

e cioè basta semplicemente elencare i file da concatenare (nell'ordine desiderato!), separandoli con il carattere «+», seguiti dal nome del file destinazione.

Ad esempio con il comando

```
copy parte1.txt + parte 2.txt testo.txt
```

si ottiene il file «testo» come concatenazione di «parte1» e di «parte2».

Evidentemente se alcuni dei file da concatenare si trovano in altre unità logiche allora basterà indicarle corret-

tamente insieme al nome del file; stesso discorso vale nel caso in cui i file si trovino in sottodirectory differenti: in questo caso basta scrivere il «path» completo del file in questione. Non a caso infatti nella sintassi del comando non si parla di semplici «<filename>» ma di più completi «<pathname>».

Altra possibilità è quella di concatenare più file in un unico file, sfruttando i caratteri «wildcard» (il «?») e l'«\*»): ad esempio con il comando

```
copy *.txt lettera.txt
```

si ottiene il file «lettera» come concatenazione di tutti i file di tipo «.txt».

In questo caso particolare però bisogna fare attenzione che il file «lettera.txt» e cioè il file destinazione non esista in quanto verrebbe segnalato errore al momento di appendere «lettera.txt» al file già formato, che è proprio «lettera.txt»: a questo punto dunque il file originario da appendere sarà stato già alterato!!

In particolare quando il sistema operativo arriva al punto di concatenare al nuovo file «lettera.txt» il vecchio «lettera.txt» segnerà con il messaggio

Content of destination lost before copy  
oppure con il (quasi) corrispondente in italiano

Contenuto dell'area di destinazione perso prima della copia

il fatto che è stata tentata un'operazione illecita.

Perciò prima di effettuare una concatenazione bisogna accertarsi che il file destinazione non sia già presente (questo nel caso in cui abbia la stessa estensione dei file da concatenare); viceversa se non ha la stessa estensione, allora bisogna verificare l'eventuale presenza del file destinazione in quanto la nuova versione andrebbe a distruggere la vecchia, come purtroppo abbiamo già visto precedentemente.

C'è inoltre da aggiungere che la concatenazione ha senso solo per i file di tipo testo, mentre per i file binari può servire solo per divertimento, sicuri che il file prodotto solo in alcuni casi funzionerà a dovere: tra l'altro in questo caso «anomalo» bisogna per forza di cose porre lo switch «/b» al termine del comando, altrimenti otterremmo una concatenazione di lunghezza minore della «somma» dei file sorgenti...

In particolare abbiamo concatenato i soliti file «ws.com», «wsvoly1.ovr» e «wsmsgs.ovr» in un unico file, dapprima senza switch (che per default corri-

<b>Gestione delle directory</b>				
chdir	dir	mkdir	path	rmdir
<b>Gestione dei file</b>				
copy	del	ren	type	
<b>Comandi vari</b>				
break	cls	ctty	date	exit
prompt	set	time	ver	verify
vol				
<b>Gestione dei «batch file»</b>				
echo	for	goto	if	pause
rem	shift			

Tabella 1 - Elenco dei comandi interni dell'MS-DOS suddivisi in gruppi funzionali.

## Come ingannare l'MS-DOS se non si ha una stampante

Tra gli oramai tantissimi fortunati possessori di personal computer, sia IBM che compatibili, una buona parte (N.d.r. tra i quali c'è anche il sottoscritto...) non possiede una stampante pur avendo in dotazione all'interno del proprio PC la relativa scheda, anch'essa originale IBM o in versione «clone» compatibile (e migliorata...).

Il problema per il quale proponiamo una semplicissima soluzione, nasce dall'infelice scelta da parte dei progettisti della tastiera di far condividere il comando «PrtScr» (che effettua l'«hard-copy» dello schermo sulla stampante) con il più innocuo «\*»: dal momento che l'asterisco «normale» è posto nel tasto relativo alla cifra «8», ecco che dovendo impostare più volte il carattere «\*», risulta più pratico premere il tasto «PrtScr/\*» (senza SHIFT) piuttosto che il tasto «\*/8» che stavolta richiede la pressione contemporanea dello SHIFT (se si è bravi con la sola mano destra, sennò con tutte e due le mani).

Ora, dovendo digitare, per esempio, una serie di formule matematiche comprendenti appunto il carattere «\*» relativo alla moltiplicazione, può capitare inavvertitamente di premere lo SHIFT insieme al tasto «PrtScr/\*», con il risultato di impostare il comando che effettua l'hard copy dello schermo così com'è sulla stampante.

Dal momento che non abbiamo la stampante quello che succede è che riusciamo ad inchiodare (apparentemente) il nostro computer: in realtà, per fortuna, basta aspettare all'incirca un minuto e si riottorrà il controllo della situazione, tutto ciò perché l'unico modo per interrompere il comando appena impostato sarebbe resettare la macchina con la conseguente perdita di quanto si stava facendo, cosa che di certo non vogliamo...

Allora, alla settecentoventottesima volta che ci capitava di sbagliare comando (provate ad esempio a scrivere una lunga sequenza di istruzioni del tipo « $a = 0.33 * \sin(\alpha) * \cos(\beta)$ » dove le parentesi devono essere impostate con lo SHIFT e l'«\*» no!), abbiamo deciso di porre rimedio alla «tignosa» questione.

Come dunque rendere innocuo il tasto di «hard-copy» senza dover spendere i soldi della stampante oppure senza doversi inventare pratiche strane da inserire al boot-strap, ecc.?!?

Tra l'altro lo stesso problema si pone se si possiede la stampante e magari l'abbiamo messa «on-line»: in questo caso

avremmo settecentoventotto paginette contenenti altrettante hard-copy dello schermo...

Il problema dunque si risolve con un ponticello posto esternamente sul connettore Cannon a 25 poli, al quale possiamo collegare il cavo della stampante (si veda la figura 2): il ponticello in esame deve collegare il foro relativo al piedino 11 con uno qualsiasi dei piedini tra 18 e 25, per semplicità il pin 24.

Il trucco, per gli addetti ai lavori, è cortocircuitare a massa (appunto uno dei pin 18-25) il segnale «busy» (pin 11) proveniente dalla «stampante-che-non-abbiamo»: facendo riferimento all'articolo di Tommaso Pantuso relativo allo standard Centronics, sul numero 45 di MCmicrocomputer, vediamo che tenere basso il segnale di «busy» di una stampante significa che la stampante è sempre pronta a ricevere dei dati.

Nel caso dunque di stampante non presente, con quel ponticello noi ne simuliamo innanzitutto la presenza e poi soprattutto una velocità di risposta infinita, a tutto vantaggio dunque dei tempi di invio dei byte, da parte del PC, tempi che ora scendono drasticamente a qualche frazione di secondo: in pratica ora non ci accorgiamo nemmeno di aver premuto insieme lo SHIFT ed il famigerato «PrtScr».

Ora questa semplicissima modifica può essere fatta «al volo» ad esempio con un pezzo di reoforo, di una comunissima resistenza, piegato ad «U» con una pinzetta: mentre consigliamo per puro scrupolo di effettuare questa pur semplice operazione a computer spento (cosa che noi «non» abbiamo fatto...), diciamo che il ponticello può restare benissimo così «all'aria» senza alcun pericolo, toccandolo, di danneggiare noi stessi né tantomeno il computer (è un collegamento di massa). Eppoi il tutto non sposterà (se siamo stati bravi) se non di poco meno di un millimetro...

Nel caso infine in cui abbiamo una stampante e non vogliamo avere a che fare con un ponticello da togliere e mettere a seconda se vogliamo o meno collegare la stampante, allora consigliamo di comprare un connettore Cannon a 25 poli «maschio» sul quale effettuare il ponticello, stavolta con una saldatura: in questo caso, oltre all'inevitabile sporgenza del connettore aggiunto, bisogna provvedere ad isolare il connettore in esame per evitare di toccare i rimanenti contatti (cosa che il computer non gradisce), magari usando un apposito «guscio» in plastica.

In questo caso dunque si è creato quello che in gergo si chiama «tappo», da connettere o meno a seconda delle necessità.



Figura 2 - Schema del connettore Cannon a 25 pin presente sul retro del PC e che consente la connessione ad una stampante con protocollo Centronics. I pin contraddistinti con un «\*» possono essere cortocircuitati con un ponticello, secondo quanto detto nell'articolo.

sponde al porre lo switch «/a») e poi con lo switch «/b»), tramite i comandi  
copy ws.\* a.aaa  
e con  
copy ws.\* b.bbb /b

Abbiamo così ottenuto i due file «a.aaa» e «b.bbb», perfettamente inutilizzabili, rispettivamente di 613 e 106240 byte!!!

Altra possibilità notevolissima del comando «copy» in questo caso, e che sfrutta ancora l'uso dell'«\*», è quella di concatenare, ad esempio a due a due, file aventi lo stesso «filename» in file aventi il filename originario ma estensione prefissata: vediamo con un esempio di spiegare meglio il concetto.

Supponiamo di aver assemblato un certo numero di file sorgenti (di tipo «.src») e perciò di aver ottenuto altrettanti file di tipo «.obj» e «.lst» ed inoltre di avere altrettanti «.map» provenienti dalla «linkata» dei suddetti «.obj»: ora vogliamo appendere tutti i «.map» ai «.lst» per ottenere dei file da stampare, che chiameremo «.pri».

Questa operazione può essere dunque effettuata in un sol colpo per mezzo del seguente comando

```
copy *.lst + *.map *.pri
```

il quale appenderà in un file di tipo «.pri» solo le «coppie» di file di tipo «.lst» e «.map»: nel caso che ad esempio esista «pippo.lst» ma non «pippo.map» allora verrà generato il file «pippo.pri» di ampiezza pari a «pippo.lst» incrementata di uno (il solito end-of-file aggiunto).

Altra possibilità è quella, sempre riferendoci all'esempio precedente, di «collezionare» tutte e sole le coppie di file di tipo «.lst» e «.map» in un unico file: ciò si ottiene con il comando:

```
copy *.lst + *.map global.pri
```

Il tutto si ottiene ancora in un sol colpo, anche se lo stesso risultato si potrebbe ottenere in due passaggi, prima creando i vari file «.pri» e poi concatenandoli in un unico file.

Tornando un attimo all'esempio di comando che permette di concatenare più file aventi la stessa estensione in un unico file avente però ancora la stessa estensione, si può fare in modo che il file già eventualmente presente non venga distrutto, a patto però che nell'operazione di «append» sia proprio il primo: ad esempio scrivendo il comando

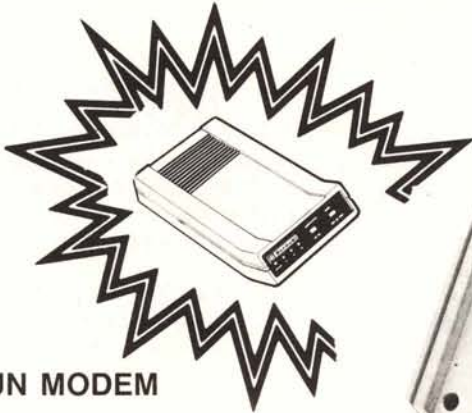
```
copy lettera.txt + *.txt
```

si ottiene correttamente un file «lettera.txt» a partire dal file «lettera.txt» stesso al quale vengono appesi tutti gli altri file di tipo «.txt», il tutto senza pericoli di sorta e perciò senza messaggi di errore da parte dell'MS-DOS.

# Bondwell™



CASELLA POSTALE 142  
56025 PONTEDERA (PI)  
VIA MISERICORDIA, 84  
TEL. 0587 - 212.312



UN MODEM

**Bondwell™**

IN REGALO  
ai primi 1.000 acquirenti



**MODEL 8**

## IL PIÙ PICCOLO E POTENTE PC PORTABILE SI CHIAMA **Bondwell™ 8**

- Facilmente trasportabile
- Peso: Kg. 4,500
- Dimensioni: cm. 28 x 31 x 78
- IBM-PC compatibile (DOS 2.11 su licenza Microsoft)
- Dischetto con MS/DOS 2.11, GW Basic 2.0 e manuali inclusi
- Basso consumo ottenuto con l'impiego di componenti CMOS
- Microprocessore: 80C88, 4.77 MHz
- Memoria RAM: 512K
- Schermo a cristalli liquidi ad alto contrasto, illuminabile, e con risoluzione 640 x 200 (grafica), 80 x 25 (testo)
- Floppy disk interno da 3" 1/2 doppia faccia/doppia densità da 720K formattati
- Orologio/Calendario mantenuto da batterie al nichel-cadmio ricaricabili
- Batterie ricaricabili 12V-3A

- Tastiera con 76 chiavi e basso profilo, compatibile con lo standard PC/XT, dotata di funzioni del PAD numerico, 10 tasti funzione ecc. ecc.
- Porta seriale standard R-232C
- Porta parallela per stampanti
- Porta per la connessione del 2° Drive (5" 1/4 oppure 3" 1/2)
- Uscite per video RGB/TTL e video-composito
- Led segnalatore intermittente di fine carica
- Alimentatore/Caricabatterie AC/DC
- Hard e Soft realizzati per ottenere il massimo della compatibilità IBM-PC. Possono essere eseguiti i più popolari pacchetti software come: Lotus 1-2-3, Symphony, D Base II e III, Wordstar, Flight Simulator, Frame work, Jem, Sidekick, PFS serie, ...

