



## i trucchi del CP/M

a cura di Pierluigi Panunzi

### Le funzioni del BDOS

*Pian pianino, puntata dopo puntata, stiamo analizzando le funzioni del BDOS, che, ricordiamolo, rappresenta il nucleo centrale del sistema operativo CP/M per accedere alle singole parti che lo compongono (le funzioni che stiamo proprio analizzando). Non è necessario conoscere l'indirizzo di partenza della singola funzione, ma bensì è sufficiente conoscerne il numero d'ordine (da porre nel registro C) ed i parametri da fornire in input. Fatto ciò sappiamo che basta semplicemente effettuare una CALL 0005: ci penserà il BDOS stesso, in base al valore posto in C, a saltare ad una predeterminata parte... di lui stesso. Inutile dire che a tale scopo all'interno del BDOS è presente una cosiddetta «jump table», che permette appunto di saltare dove necessario.*

*In tal modo il tutto risulta così «trasparente per l'utente», il quale è svincolato, come detto, da altre incombenze. Siamo oramai abbastanza avanti nella nostra analisi e, come è già successo in parecchi casi, daremo per scontata la conoscenza di quanto riportato nelle puntate precedenti: ai lettori basta eventualmente avere sotto mano i numeri precedenti.*

*Cercheremo però, come d'altro canto facciamo spesso, di non essere categorici, magari ripetendo per l'ennesima volta concetti scontati, soprattutto nel caso in cui riteniamo che il lettore si debba soffermare.*

Siamo così arrivati alla

#### Funzione 29: Get Read-Only Disks

Si tratta di una routine accessoria, forse di non fondamentale importanza ed in genere raramente usata: probabilmente ai progettisti del CP/M è «saltata fuori gratuitamente» come differenziazione di un'altra routine.

Comunque a parte ogni considera-

zione, la funzione in esame, come si potrà facilmente arguire dal suo nome, permette di avere un'indicazione di quali dei 16 dischi logici possibili (da A: a P:) sia nello stato di «Read-Only», sia per effetto della chiamata alla funzione 28, la quale pone appunto una unità logica nello stato R/O, oppure per effetto di elucubrazioni personali del BDOS stesso, quando si accorge che è stato cambiato il dischetto di una certa unità a dischi.

In particolare l'informazione sullo stato delle unità logiche a dischi ci viene mostrata come «bit map» nella coppia di registri HL.

Con tale termine intendiamo che la coppia HL, più che riportate un valore numerico, deve essere letta come un insieme di 16 bit, ad ognuno dei quali corrisponde una determinata unità logica: ecco che al meno significativo di tali 16 bit (e per la precisione l'LSB del registro L) corrisponderà l'unità logica A; al successivo bit corrisponderà l'unità B; e così via fino ad arrivare al bit più significativo (l'MSB del registro H) che corrisponde ad un'imprescindibile unità P:

Detto questo, non rimane altro che analizzare il significato di un «1» o di uno «0» in una certa posizione tra le 16: in particolare un valore «1» indica che il corrispondente disco logico è in stato di «Read-Only», come d'altro canto era semplice arguire, così come è fin troppo ovvio che un valore «0» indica la condizione opposta...

#### Funzione 30: Set File Attributes

Ecco una funzione, che a differenza della precedente, è di particolare interesse ed importanza, dal momento che consente di gestire le caratteristiche di un certo file manipolando opportunamente i caratteri che ne compongono il nome.

In particolare sappiamo che per identificare un file abbiamo la possibi-

lità di assegnarne un nome di al massimo 8 caratteri, più un'«estensione» eventuale di altri 3 caratteri: sapendo che i singoli caratteri utilizzati (come pure i «blank» che rimpiazzano spazi vuoti) vengono codificati in ASCII e perciò a 7 bit, si vede come abbiamo a disposizione l'ottavo bit (l'MSB) di ognuno di tali 11 caratteri per altrettanti flag.

Sappiamo già che i «bit alti» di due dei tre caratteri costituenti l'estensione hanno un significato ben preciso: per convenzione del CP/M si indicano con

$f1' f2' f3' \dots f8'$

i rispettivi MSB degli otto caratteri possibili per il «filename» mentre si indicano con

$l1' l2' l3'$

gli MSB relativi ai tre caratteri dell'estensione.

Dicevamo dunque che i primi dei due tre «t» hanno già un significato, se settati:

$l1'$  il file è di tipo *Read-Only*

$l2'$  il file è di tipo *System*

$l3'$  riservato per futuri usi del CP/M

Viceversa gli 8 «bit alti», gli «f» sono in parte utilizzabili ed in parte no, secondo lo schema seguente:

da  $f1'$  a  $f4'$  a disposizione dell'utente

da  $f5'$  a  $f8'$  riservati per usi futuri del CP/M.

Per far sì che il nostro file abbia i «bit alti» settati a seconda delle nostre necessità, bisogna costruire un FCB (File Control Block), in cui il nome completo del nostro file (senza «?» e perciò non ambiguo) ha gli 11 bit più significativi settati o meno a seconda delle necessità. Al solito, l'indirizzo di tale FCB deve essere posto nella coppia DE all'atto della chiamata alla

funzione in esame. Ovviamente anche se ciò poteva non essere implicito, il BDOS bloccherà ogni tentativo di scrivere un file che abbia l'attributo «Read-Only» settato, mentre viceversa (come stavolta è ovvio), la funzione e perciò il BDOS provvederà a cambiare i bit di attributo di tutte le directory entry relative al file in esame, alterando se necessario i bit fino ad arrivare alla configurazione proposta nel FCB.

Per quanto riguarda i quattro bit a disposizione dell'utente, è chiaro che possono servire bene per vari scopi, ad esempio per «marcare» un file dopo averlo letto e perciò per non rileggerlo daccapo, oppure per indicare che un file dovrà essere successivamente cancellato oppure al contrario reso «Read-Only», oppure ancora per catalogare i file a seconda del loro contenuto (ad esempio programmi, file di dati, file di stampa, file costituenti l'hard-copy della pagina video, ecc.).

Come si vede sarà poi compito del nostro programma l'andare a cercare, con le funzioni «Search First» e poi «Seach Next», i file desiderati: queste due funzioni in particolare, nel ricercare un determinato file, utilizzano tutti gli 8 bit sia del «filename» che del «type».

### Funzione 31: Get Disk Parameter Block Address

Ecco ora una funzione particolarmente semplice come effetti ottenibili, ma che invece implica concetti molto importanti.

Detta così, in parole povere, la funzione in esame fornisce in output nella coppia HL l'indirizzo del DPB (Disk Parameter Block) e niente più...

Tra l'altro, alla luce di quanto diremo tra breve, tale funzione non servirebbe nemmeno, se i programmi che la usano non andranno a girare su altri computer...

In breve il DPB (continueremo a chiamarlo così) è un'importantissima tabella che contiene informazioni riguardanti una ben determinata unità logica: sistemi aventi uno o due floppy disk, un hard disk e magari anche una RAM-Disk, possiederà tre differenti DPB, una per i floppy, una per l'hard ed una per la RAM. Promettendo di ritornare su questi ultimi due tipi di memoria di massa, magari dopo aver concluso l'analisi di tutte le funzioni del BDOS, andiamo a vedere in dettaglio come è formato un DPB, a partire da alcune notizie storiche.

Nelle versioni originarie del CP/M, il DPB era inglobato all'interno del BDOS e perciò non raggiungibile dall'occhio dell'utente «medio», ma viceversa non del tutto utilizzabile neanche dagli «smanettomani» in quanto

conteneva e contiene tuttora elementi di non immediata comprensione: anche in questo caso si aveva a che fare con «proprietary information» da parte della Digital Research stessa.

In tal modo però i sistemi dotati di CP/M potevano usare unità logiche a dischi solo di caratteristiche ed in numero specificati dal misterioso DPB.

Con l'avvento degli «hard-disk», che rivoluzionavano del tutto le caratteristiche delle unità logiche (a cominciare ad esempio dal numero di tracce a disposizione), la Digital ha deciso di spostare il DPB all'interno del BIOS, a completa disposizione dell'utente, lasciando però ancora alcune terminologie non del tutto chiare ed anzi alquanto criptiche.

Ecco che perciò a questo punto non ha molto senso una routine del BDOS che fornisce l'indirizzo di un qualcosa direttamente manipolabile dall'utente, il quale sa perfettamente «dove» si trova il DPB...

Comunque il DPB è un insieme di 15 byte, suddivisi tra singoli byte e word (coppie di byte), secondo il seguente schema in cui il numero posto a sinistra indica di quale byte si tratta: e la successiva lettera «B» o «W» indica appunto trattarsi di un byte singolo o di una coppia di essi (word, appunto):

1,2	W	Settori per traccia
3	B	Block shift
4	B	Block mask
5	B	Extent mask
6,7	W	massimo numero di Allocation Blocks
8,9	W	directory entry - 1
10,11	W	bit map degli alloc. block della directory
12,13	W	numero byte per test della directory
14,15	W	numero tracce fisiche prima della directory

Come si vede già dai nomi, alcune voci risultano poco rassicuranti, mentre altre traggono decisamente in inganno: le analizzeremo una per una cercando di aggiungere dove è possibile delle nozioni acquisite dall'esperienza del redattore nello studio delle caratteristiche del CP/M, studio che ha consentito l'installazione di tale sistema operativo in un computer «home-made», con risultati veramente notevoli.

Per inciso, se i lettori fossero interessati a tale tematica, allora si potrebbe in un prossimo numero parlare di questa «dura esperienza», avvenuta, lo ripetiamo, in maniera del tutto autonoma...

Iniziamo dunque dal primo campo del DPB.

#### 1,2) Sectors per track

Preferiamo lasciare il termine inglese,

almeno per titolo di questi paragrafi: in prima analisi la traduzione «numero di settori per traccia» trae senz'altro in inganno, dal momento che il CP/M è, per vetustà, ancorato (il poverino...) a settori fisici di 128 byte, a dispetto degli effettivi byte costituenti un settore fisico ad esempio di un 8" per non parlare di un hard-disk. Per vedere ancora meglio dove sta «il bussillis», analizziamo da vicino il caso del vecchio ma glorioso Osborne I: nel DPB relativo alle due unità a dischi (a 5") troviamo che tale valore è 20, come dire che ci sono 20 settori per traccia. Invece no! Ce ne stanno esattamente la metà, come ben sappiamo dal manuale o per «ispezione» tramite uno qualsiasi dei tanti «Disk Handler».

Come si spiega questa incongruenza?

L'unica spiegazione (grazie all'esperienza e non alla Digital...) viene dal fatto che in realtà con le routine del BIOS di lettura e di scrittura si fa incessantemente credere al CP/M di avere a che fare con settori da 128 byte, mentre in realtà si ha a che fare (attenzione!) con un numero prefissato di settori «logici», ognuno formato generalmente da più di 128 byte: in particolare la tecnica usata per far vedere al CP/M unicamente settori di 128 byte sfrutta il cosiddetto «Blocking-Deblocking Algorithm» (BDA), del quale forse parleremo in prossime puntate, dedicate da sole a tale vastissimo argomento. Ecco che perciò il valore 20 dell'Osborne I indica la presenza di altrettanti «settori logici», ognuno però formato da 256 byte. Inoltre, il CP/M assume che il primo di tali settori logici sia quello avente il numero 0, lasciando alla routine «SECTAN» del BIOS il compito ingrato di aggiustare tutto in termini di settori fisici e di primo settore. Per inciso poi a livello hardware, per quanto riguarda il circuito integrato che si occupa della gestione dei dischi, in genere si parte dal settore numero 1 e non da quello 0...

#### 3) Block shift

Ecco il primo dei tre campi aventi un nome strano, e cioè di non immediata interpretazione, strettamente legato all'ampiezza dell'«Allocation Block» utilizzato in quel particolare sistema. Rimandando per i dettagli alla puntata dei «Trucchi» della rivista numero 42, ricordiamo in breve che l'«Allocation Block» (AB) è in pratica un insieme di un certo numero di settori fisici, la cui ampiezza è scelta in base a criteri contrastanti di efficienza e di sicurezza: in particolare si possono avere AB di ampiezza variabile tra 1024 e 16384 byte, secondo le potenze del 2. Ricordando che un settore è for-

mato da 128 byte (80H in esadecimale oppure 10000000 in ottale), il «block shift» in esame non è altro che il «numero di shift a sinistra» di 80H per ottenere proprio l'ampiezza dell'AB. Detto così, suona stranamente sinistro, mentre in realtà, osservando la tabellina, non è così difficile:

Dimensione AB in byte	Block Shift
1024	3
2048	4
4096	5
8192	6
16384	7

L'uso di tale parametro, come pure dei prossimi due, è praticamente riservato al CP/M durante operazioni che coinvolgono le unità a dischi.

#### 4) Block Mask

Altro parametro ad uso quasi esclusivo del CP/M, rappresenta, come dice il suo nome, una maschera, formata da tanti «1» quanto è il valore del precedente «block shift».

Mentre ci domandiamo quale potrebbe essere la sua utilizzazione possiamo avere la conferma del suo valore dalla tabellina seguente:

Dimensione AB in byte	Block Mask	Valore decimale
1024	111	7
2048	1111	15
4096	11111	31
8192	111111	63
16384	1111111	127

#### 5) Extent Mask

Ecco dunque l'ultimo dei tre valori usati dal CP/M nei suoi calcoli durante la gestione dei file su disco.

Invece di cercare una formula matematica, peraltro veramente semplice da trovare, possiamo dare un'occhiata alla tabellina seguente dalla quale si evince che il valore in esame dipende sì dall'ampiezza dell'AB, ma anche dal numero totale di allocation block circolanti nel sistema. Agli abili lettori lasciamo le elucubrazioni del caso:

Dimensione in byte dell'AB	Extent Mask	
	se AB ≤ 255	se AB > 255
1024	0	impossibile
2048	1	0
4096	3	1
8192	7	3
16384	15	7

#### 6,7) Maximum Allocation Block Number

Il nome in questo caso non trae in inganno: dal momento che il primo allocation block ha il numero 0, ecco che il valore dato dal parametro in esame non è altro che il numero di al-

location block a disposizione, diminuito di un'unità.

Senza scendere troppo in dettagli, il numero di allocation block a disposizione in un certo disco è dato, come è facile immaginare, dal rapporto tra il numero totale di byte a disposizione per i file sul disco, diviso l'ampiezza di un allocation block in byte: dopo aver parlato di tutti i parametri del DPB ritorneremo brevemente su tutti i singoli valori analizzandoli all'interno dell'Osborne I.

Comunque nel caso presente si vede che il valore può facilmente superare i 255, specie per i floppy da 8" per non parlare degli hard-disk, per cui servono 2 byte.

#### 8,9) Number of Directory entries - 1

Anche in questo caso il termine inglese risulta decisamente chiaro: bisogna solo fare attenzione che si parla di «directory entry» e non di «file». In pratica i due valori coincidono solo nel caso di un dischetto pieno di file ognuno avente un'unica entry: invece in generale ad ogni file corrispondono più entry per cui si perde la coincidenza tra i valori.

Dal momento che una directory entry consta di 32 byte, ecco che ci semplifichiamo le cose scegliendone un numero finale tale da riempire completamente uno o più allocation block: ad esempio con un AB di 1024 byte avremo esattamente 32 directory entries. Un valore di 30 a questo punto non avrebbe senso in quanto si sprecherebbero byte sul disco, altrimenti inutilizzabili.

Però anche il numero fornito da tale parametro in esame non dovrebbe essere troppo elevato in quanto il CP/M scandisce sequenzialmente la directory, con un tempo direttamente proporzionale all'ampiezza della directory stessa: anche qui si ha da risolvere un compromesso tra l'ampiezza dell'AB ed il numero delle directory entry, nel senso che tanto più è grande un AB tante più saranno le entry, con i già visti problemi di accesso sequenziale.

#### 10,11) Allocation Blocks for the Directory

Diretta conseguenza dei discorsi precedenti o se vogliamo anche causa (è nato prima l'uovo o la gallina?), tale parametro dipende strettamente dal parametro precedente: noto perciò il numero di AB in cui verrà posizionata la directory, il parametro in esame ne rappresenterà la «bit map», secondo una prassi ormai consolidata in casi analoghi.

L'unica differenza è che i bit stavolta devono partire dall'MSB di una word invece che dall'LSB: supponendo che per la directory abbiamo bisogno di tre AB, allora la «bit map» sarà

la seguente:

(MSB) 1110000000000000 (LSB)

corrispondente ad un valore esadecimale di E0H.

Altro esempio: un dischetto «povero» con un solo allocation block per la directory fornirà un valore

(MSB) 1000000000000000 (LSB)

corrispondente all'esadecimale 80H.

Il perché di questa strana scelta, contraria a quella a cui eravamo abituati, lo sanno solo quelli della Digital...

#### 12,13) Size of Buffer for Directory Checking

Ecco un altro parametro dal nome alquanto sconcertante: in pratica si tratta di una zona di memoria nella quale il CP/M «comprime» le informazioni contenute nella directory, per potersi accorgere se eventualmente l'incauto utente ha cambiato il dischetto senza prendere le dovute precauzioni.

Chiamo il classico «velo pietoso» sulla gestione da parte del CP/M di tale zona di memoria dal momento che i soliti della Digital Research non forniscono altre informazioni se non il fatto che basta prevedere un byte appena per ben quattro directory entry: la prima idea che viene in mente è che tale byte rappresenti la «checksum» dei byte costituenti quattro directory entry (in totale 128 byte), ma questa è solo una nostra supposizione...

#### 14,15) Number of tracks before the Directory

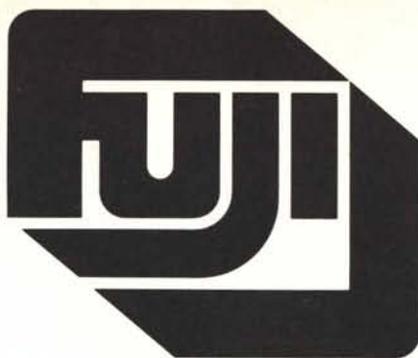
L'ultimo parametro ha un nome decisamente più abbordabile e perciò comprensibile: indica infatti, come dice il nome inglese, il numero di tracce riservate al Sistema Operativo (sì, è sempre lui, il CP/M!). Questo almeno nelle intenzioni degli sconosciuti «man-non-troppo» progettisti, in quanto tale valore è utilissimo per dividere un'unità di memoria di massa (un hard disk ad esempio) in più unità di memoria logiche. Si sente infatti spesso parlare di un hard disk da, supponiamo, 20 Mbyte, suddiviso in due unità logiche, rappresentate ad esempio come A: e B:. Come si ottiene ciò? Facciamo un esempio numerico, con valori puramente esemplificativi.

Supponiamo che il nostro disco possieda 101 tracce, delle quali una riservata, in quanto contenente l'immagine del CP/M da caricare in memoria all'atto del bootstrap.

Le restanti 100 tracce (capito perché siamo partiti da 101!?) si possono dividere in due, 50 per l'unità logica A: ed altrettante per la B:.



# Gli specialisti al vostro servizio



## C.B.S. S.r.l.

Via Comelico, 3 - Milano  
Tel. 02/5400421

## A SYSTEM S.r.l.

Via Turrone, 8 - Milano  
Tel. 02/7383531

## LEONI SHOP S.r.l.

Corso Porta Romana, 123  
Milano - Tel. 02/5466080

## MICROCORNER S.r.l.

Via U. Bassi, 3 - Milano  
Tel. 02/6881685

## MARCUCCI S.r.l.

Via Bronzetti, 37 - Milano  
Tel. 02/7386051

## MELCHIONI COMPUTERTIME

V.le Europa, 49 - Cologno M.  
(Mi) - Tel. 02/2540607

## POLISERVICE S.r.l.

Via XXV Aprile, 13  
Cinisello Balsamo (Mi)  
Tel. 02/6128749

## BERMAN S.r.l.

Bastioni P.ta Volta, 11  
Milano - Tel. 02/6570128

## BRUNO S.r.l.

Via Rubini, 5 - Como  
Tel. 031/260538

## INGROSCARTA S.r.l.

Via V. Emanuele II, 17  
Roncadelle (BS)  
Tel. 030/2780397

## TUTTO EDP S.r.l.

Strada Mongreno, 34  
Torino - Tel. 011/8999590

## AZETA ACCESSORI S.r.l.

Via Augusto Verità, 4  
Verona - Tel. 045/34668

## ESACOMP

Via Roveggia, 43 - Verona  
Tel. 045/581792

## STEMASOFT S.n.c.

P.zzola Gualdi, 1 - Vicenza  
Tel. 0444/33210

## MEDIA VENETA S.r.l.

Via Belzoni, 68 - Padova  
Tel. 049/39158

## 2PD S.n.c.

Via U. Foscolo, 22/1  
Padova - Tel. 049/34292

## IMOCO S.r.l.

Via Postioma, 63/B - Villorba  
(Tv) - Tel. 0422/911107

## STRAGÀ

Via Grazioli, 17/19 - Trento  
Tel. 0461/981101

## PERSONAL COMPUTER

Cannareggio 5894  
Venezia - Tel. 041/29040

## FIVE COMPUTERS S.r.l.

Via G. D'Annunzio, 29/1  
Trieste - Tel. 040/741025

## MEDIA S.r.l.

Via Mascarella, 59/B  
Bologna - Tel. 051/237022-3

## BITZEROUNO S.r.l.

Via Che Guevara, 55/B  
Reggio Emilia  
Tel. 0522/293241

## I.L. ELETTRONICA

Via Lunigiana, 481  
La Spezia - Tel. 0187/511739

## BIT BYTE

Via V. Veneto, 21/23  
Marina di Massa  
Tel. 0585/245496

## B.F. ELETTRONICA

Via Corridoni, 51 - Pisa  
Tel. 050/500620

## LOGOS INFORMATICA

Via S. Concordio, 537  
Lucca - Tel. 0583/55519

## C.D.E.

V.le Adua, 350 - Pistoia  
Tel. 0573/400712

## ELETTRONICA ALESSI

Via Cimarosa, 1 - Piombino  
Tel. 0565/39090

## TRIAD E INFORMATICA S.r.l.

Via di Brozzi, 72 - Firenze  
Tel. 055/310401

## SUMUS S.r.l.

Via San Gallo, 16/R  
Firenze - Tel. 055/295361

## PAOLETTI FERRERO

Via Il Prato 40/R - 42/R  
Firenze - Tel. 055/294974

## CBS UMBRIA S.r.l.

Via Campo di Marte, 4  
Perugia - Tel. 075/756843/4

## CBS SUD S.r.l.

Via Melchiorri, 2 - Roma  
Tel. 06/4241552-4241333

## AFTERPRINT

Via A. Ravà, 106 - Roma  
Tel. 06/5401837

## AIS S.r.l.

Via Jacopo da Lentini, 16  
Pomezia - Tel. 06/9122963

## G.T.I. S.a.s.

Via Romagnoli, 90 - Latina  
Tel. 0773/480176

## DATA OFFICE S.p.a.

Via Sicilia, 205 - Roma  
Tel. 06/4750576

## MESCHINI S.r.l.

V.le Giotto, 1/B - Roma  
Tel. 06/5740874

## CHOPIN DI M. IACOVACCI

Via Chopin, 27 - Roma  
Tel. 06/5030757

## INDIS S.r.l.

V.le M. Gelsomino, 10  
Roma - Tel. 06/575527

## PR. COMPUTER DI RUO L.

Via P. Giovanni XXIII, 9  
Vico del Gargano  
Tel. 0884/93750

## ERMES S.n.c.

Via S. Lucia, 50  
Napoli - Tel. 081/402419

## ENGINEERING S.r.l.

Via Carducci, 15 - Napoli  
Tel. 081/402660

## LUCANA SISTEMI S.r.l.

Via Don Minzoni, 4 - Matera  
Tel. 0835/214423

## E.D.P. SHOP S.r.l.

Via Temo, 4/A - Cagliari  
Tel. 070/285627

## Distributore esclusivista

Fuji per l'Italia:

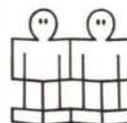


C.B.S. CONTROL BYTE SYSTEM

Via Comelico, n. 3 - 20135 Milano

Tel. 02/5400421 - Tlx. 350136 CIBIES I

## Assistenza tecnica e nastri originali



Anche le stampanti più affidabili e più robuste hanno la necessità, talvolta, di assistenza e manutenzione.

La Mannesmann Tally offre, su tutto il territorio nazionale, un Servizio di Assistenza tecnica rapida ed efficiente che viene proposto all'utente in varie forme, con o senza contratto, secondo le specifiche esigenze:

- Pronto intervento presso l'utente
- Riparazioni presso i laboratori Mannesmann Tally
- Riparazioni a "prezzo fisso" per i modelli di macchine a basso costo
- Contratti di estensione della garanzia

Per una maggiore affidabilità e una minore manutenzione è consigliabile utilizzare sempre nastri originali Mannesmann Tally.

Per maggiori informazioni chiamare:

Milano - Tel. (02) 4502850/855/860/865/870

Roma - Tel. (06) 8278458

Torino - Tel. (011) 8225171

Bologna - Tel. (051) 832508

# Tutte le garanzie del n. 1



**MANNESMANN**  
**TALLY**

20094 Corsico (MI) - Via Cadamosto, 3