

I «device driver»

seconda parte

Dopo aver descritto nella scorsa puntata le caratteristiche fondamentali di un driver, che ci sono servite per imparare a conoscere questo nuovo ed utile mezzo che abbiamo a disposizione, andiamo a vedere in questo numero come il Sistema Operativo colloquia con un driver.

Nell'MS-DOS esiste una differenza tra un programma «normale» ed un driver: mentre un programma in genere viene attivato e da quel momento ha il controllo sul sistema (e magari è lui a chiamare le routine del DOS per mezzo degli appositi INT), viceversa sappiamo che un driver è una specie di appendice all'MS-DOS, una sua parte integrante e come tale viene interpellato dal DOS stesso solo quando quest'ultimo ne ha bisogno. Insomma un driver non prende (almeno in generale...) il controllo della situazione, ma serve ad implementare alcune funzioni accessorie utilizzabili da altri programmi applicativi

Il protocollo di interscambio con l'MS-DOS

Se paragoniamo per semplicità un driver ad una «funzione dell'MS-DOS» (tipo una funzione dell'INT 21, tanto per fare un esempio), sappiamo che prima di attivarla dobbiamo innanzitutto caricare opportunamente i registri della CPU in base a ciò che desideriamo dalla funzione stessa: ad esempio nel registro AH sappiamo che dobbiamo porre il numero corrispondente alla funzione che vogliamo svolgere, mentre negli altri registri andranno, a seconda dei casi, indirizzi di memoria, valori chiave, ecc.

Analogamente nel caso di un driver, il DOS sa che potrà svolgere un certo numero di funzioni e perciò innanzitutto indicherà quale funzione è richiesta e poi comunicherà altri valori, parametri, puntatori, ecc. che serviranno per svolgere correttamente la funzione: il driver non farà altro che leggere tali valori, effettuare la funzione richiesta, ed eventualmente fornire valori in uscita (stato, condizioni di errore, ecc.).

Ecco che perciò, per maggiore generalità possibile, la comunicazione tra il DOS ed un driver non avviene tramite registri, ma per mezzo di una tabella chiamata «Request Header» (contenente appunto dati e informazioni varie), posta in memoria, e della quale il DOS fornisce l'indirizzo iniziale, sotto forma di «segment:offset», rispettivamente nei registri ES:BX.

Andiamo ora ad analizzare in dettaglio la struttura del «Request Header» (il cui nome riflette appunto il fatto che è il

DOS a richiedere un servizio da parte del driver), per poi analizzare successivamente le varie funzioni che il driver deve poter (in generale) svolgere.

Il «Request Header»

Vediamo innanzitutto dalla figura 1 che il «Request Header» è costituito da un certo numero di celle consecutive di memoria, numero che varierà a seconda della funzione richiesta, ma dove almeno un certo numero di byte è sicuramente fisso e presente in ogni situazione: la parte fissa è quella che propriamente si chiama «Header» (che significa infatti «testata»), ma per estensione possiamo intendere con tale nome «tutta» la tabella, compresa la parte variabile. In particolare si ha la struttura seguente:

— il primo campo («Length») indica il numero di byte che compongono l'header più la parte variabile, e dato che tale campo è formato da un byte si ha che la lunghezza totale non potrà mai superare i 256 byte, cosa che difatti succede, anzi in genere ne vengono usati molti di meno.

Comunque questo non sarà mai per noi un problema, dal momento che si tratta di una quantità fornita dal DOS e perciò presumibilmente corretta (ed in genere quasi mai usata dal driver... perché tutto sommato inutile).

— Il secondo campo («Unit Code») è usato solamente dai device driver di tipo «block» e rappresenta il numero dell'unità verso la quale viene svolta la richiesta: per i «character device» tale campo non ha alcun significato.

— Il terzo campo («Command Code») è praticamente quello più importante, in quanto contiene un numero che rappresenta proprio la funzione richiesta al driver: i valori possono variare tra 0 e 15 ed in particolare rimandiamo alla tabella 1 per la corrispondenza «numero-funzione richiesta».

— Il quarto campo («Status») è riservato al driver per poter scrivere una parola di stato, indicante fatti salienti riguardo il funzionamento (corretto o meno, in corso o terminato) della funzione attivata: è qui che il driver indicherà ad esempio l'insorgere di eventuali errori, nel qual caso il DOS prenderà i provvedi-

indirizzo	tipo	campo
00	BYTE	LENGTH : lunghezza dell'RH compresa la parte variabile
01	BYTE	UNIT CODE : codice dell'unità interpellata (solo per i "block device", mentre "e" senza significato per i "character device")
02	BYTE	COMMAND CODE : codice di comando (vedasi la tabella 1)
03	WORD	STATUS: word di stato
05	B BYTE	area riservata per l'MSDOS
0D	var.	zona contenente i dati relativi alla particolare funzione attivata

Figura 1
Struttura del «Request Header» e cioè di quella zona di memoria predisposta per la comunicazione tra l'MS-DOS ed un driver quando questo viene attivato.

menti del caso (sempre che siano già stati previsti...). In particolare rimandiamo alla tabella 2 per il dettaglio del significato dei vari bit.

— L'ultimo campo della parte fissa e perciò del «Request Header» vero e proprio non è altro che una zona formata da 8 byte riservata al DOS e sui quali nulla è dato sapere, come in tutti quei casi analoghi in cui troviamo la dicitura «DOS reserved».

— L'ultimissimo campo è quello che per l'appunto varia a seconda della funzione richiesta e che in alcuni casi può addirittura mancare, senza che il driver ne abbia a dolersi...

Nella tabella 1 abbiamo dunque l'elenco delle possibili funzioni che il driver dovrebbe poter svolgere: il condizionale rappresenta la possibilità da parte del programmatore di implementare le routine che desidera, ovviamente nel rispetto delle regole che via via signaleremo, nonché di una congruenza a livello logico. Mentre infatti deve esistere almeno la routine Init, potrebbe essere illogico e incongruente implementare un driver per dischetti in cui manca totalmente la parte relativa all'Input, a meno che intenzionalmente non si voglia gestire un driver di tipo «write-only»!

Dicevamo dunque della routine Init: essa deve essere sempre presente e serve per installare il driver o meglio inizializzarlo.

Al limite in alcuni casi può magari essere l'unica funzione presente nel driver, che viene così «lanciato» all'inizio dei tempi e mai più toccato: bisogna però prevedere eventualmente la gestione dei «command code» diversi da 0 nel caso in cui il DOS richieda una funzione non implementata, ma teoricamente esistente.

Abbiamo detto la scorsa puntata che l'header del driver (attenzione, non il «Request Header»!) contiene due campi relativi a due entry point del driver stesso, la «strategy routine» e la «interrupt routine»: sappiamo che la prima è la parte del driver che viene eseguita dal DOS quando ha bisogno delle funzioni del driver stesso.

È proprio in tale occasione che il DOS fornisce nella coppia ES:BX l'indirizzo completo della tabella «Request Header»: a tal proposito la «strategy routi-

Tabella 1
In questa tabella è indicata la corrispondenza tra il «Command Code» presente nel «Request Header» e la funzione richiesta al driver, a seconda che sia di tipo «block» o «character»: «Nop» significa ovviamente che la funzione non ha senso per il driver di un particolare tipo.

codice	funzione	block device	character device
00	INIT	INIT	INIT
01	MEDIA CHECK	NOP	NOP
02	BUILD BPE	NOP	NOP
03	IOCTL input	IOCTL input	IOCTL input
04	INPUT	INPUT	INPUT
05	NOP	NONDESTRUCTIVE INPUT	NONDESTRUCTIVE INPUT
06	NOP	INPUT STATUS	INPUT STATUS
07	NOP	INPUT FLUSH	INPUT FLUSH
08	OUTPUT	OUTPUT	OUTPUT
09	OUTPUT with VERIFY	OUTPUT with VERIFY	OUTPUT with VERIFY
0A	NOP	OUTPUT STATUS	OUTPUT STATUS
0B	NOP	OUTPUT FLUSH	OUTPUT FLUSH
0C	IOCTL output	IOCTL output	IOCTL output
0D	DEVICE OPEN	DEVICE OPEN	DEVICE OPEN
0E	DEVICE CLOSE	DEVICE CLOSE	DEVICE CLOSE
0F	REMOVABLE MEDIA	NOP	NOP

ne» dovrà provvedere a memorizzare in un posto sicuro tale indirizzo per poterlo usare nel seguito. La seconda routine, l'«interrupt routine», invece è quella che viene attivata subito dopo la «strategy» e sarà proprio quella che esegue la funzione in base ai valori contenuti nel «Request Header», il cui indirizzo era stato salvato già in precedenza...

Tale routine, in base al valore trovato nel campo «command code», prenderà una strada piuttosto che un'altra, al limite entrando in una routine di errore, nel caso in cui la funzione non sia stata implementata.

Per vedere dunque in grandi linee come deve essere fatto un driver, ritorneremo all'esempio proposto la scorsa puntata e riguardante un driver chiamato «SER» relativo alla gestione di una porta seriale.

Avevamo detto che si poteva comporre di cinque routine base:

- una di inizializzazione
- una relativa all'input seriale
- una per l'output
- una di controllo dello stato di input
- ed una per controllare lo stato in output.

La prima serve ad esempio per inizializzare la scheda seriale, settando innanzitutto le caratteristiche della linea, quali ad esempio la baud rate, il numero di bit del dato e la presenza di parità e poi per settare l'interrupt che gestisce l'I/O per mezzo dell'UART presente sulla scheda di interfaccia e non da ultimo potrà servire ad allocare in memoria uno o più

buffer dove raccogliere i dati che mano a mano arrivano dalla linea ed uno o più buffer che viceversa contengono i dati da spedire lungo la linea: il tutto, come suol dirsi, in «full duplex» e cioè con rice-trasmissione contemporanea.

La seconda routine invece è prevista per la lettura di un dato alla volta di quelli ricevuti dalla linea e memorizzati nei buffer di input grazie ad un'apposita routine di gestione dell'interrupt generato dall'UART allorché ha ricevuto un messaggio dalla linea.

La terza routine provvederà a riempire, un dato alla volta, i buffer di output, dai quali un'apposita routine di interrupt estrarrà i singoli messaggi da inviare sulla linea allorché l'UART avrà la disponibilità a farlo.

La quarta e la quinta routine serviranno a gestire gli eventuali errori che si hanno tanto in ricezione che in trasmissione.

Tutto questo perché abbiamo detto che il nostro driver debba essere di tipo «character»: questo significa ad esempio che verso il dispositivo «SER:» potremo inviare il contenuto di un file con un comando MS-DOS del tipo

```
copy file.dat ser:
```

oppure potremo vedere su video quanto riceviamo dalla linea, per mezzo di un comando MS-DOS del tipo

```
copy con ser:
```

Però per ottenere questo si dovrà pensare abbastanza...

Viceversa avremmo potuto decidere di creare un driver di tipo «block» e perciò gestire i dati in input ed output non uno alla volta ma bufferizzati: si otterrebbe così una semplicità ancora maggiore (o no!?) pensando ai dati provenienti in toto da una linea seriale come un'unità chiamata ad esempio «S:», gestita a livello di «settori» come un'unità a disco vera e propria...

Inutile dire che per ottenere questo dovremo compiere sforzi ancor più grandi che non nel caso precedente: il tutto per avere una funzionalità nettamente più semplice ed automatica che non quella ottenibile («lanciando» un programma applicativo qualsiasi.

Il campo «command code»

Facendo riferimento alla tabella 1, vediamo che delle 16 funzioni previste per un driver, 11 vengono gestite dai driver di tipo «block», mentre saliamo a 13 per i «character device»: alcune, le più importanti, valgono per entrambi i tipi, mentre altre, le più specializzate, valgono solo per uno dei due tipi.

In generale non è detto che un singolo driver le debba prevedere tutte dal momento che sono state previste funzioni per qualsiasi evenienza e tipo di driver: in effetti bisogna poi ricordare che sarà sempre e solo l'MS-DOS ad attivare il driver e quindi sarà lui stesso a scegliere la funzione.

Questo per dire che il DOS non potrà inventare e pretendere funzioni inesistenti, al di fuori di quelle viste, a meno di non cambiare il DOS stesso...

Infatti c'è da dire che le funzioni 13, 14 e 15 sono state introdotte a partire dalla versione 3.00 e 3.10 dell'MS-DOS e perciò non possono funzionare con versioni più vecchie, a meno che chi ha progettato il driver non abbia previsto la segnalazione di errore allorché il «command code» fosse maggiore di quello massimo previsto.

Ecco perciò un primo consiglio: quando si scrive un driver, nella «interrupt routine» bisogna testare se il «command code» è maggiore del numero massimo di funzione consentito dalla versione usata dell'MS-DOS e dare segnalazione di «unknown command» in caso positivo: bisogna infatti pensare che il passaggio ad una versione successiva del DOS comporterà almeno un aumento del numero di funzioni, se non altro per salvaguardare la compatibilità verso il basso.

Ecco che perciò, senza addentrarci nell'analisi delle singole funzioni che inizieranno nella prossima puntata, diamo una breve scorsa alle varie funzioni a

-----		-----		-----	
[E] reserved: [B] [D]		ERROR		STATUS WORD	
-----		-----		-----	
campo	significato				

E	:	se vale 1 c'è un errore (specificato nel campo ERROR CODE)	:		
B	:	bit di BUSY, settato in alcune situazioni	:		
D	:	bit di DONE, settato al termine della funzione	:		
ERROR	:	(se bit E = 1) ha le seguenti possibilità:			
	:	00	:	write protect violation	
	:	01	:	unknown unit	
	:	02	:	device not ready	
	:	03	:	unknown command	
	:	04	:	CRC error	
	:	05	:	bad drive request structure length	
	:	06	:	seek error	
	:	07	:	unknown media	
	:	08	:	sector not found	
	:	09	:	printer out of paper	
	:	0A	:	write fault	
	:	0B	:	read fault	
	:	0C	:	general failure	
	:	0D	:	reserved	
	:	0E	:	reserved	
	:	0F	:	invalid disk change	

Tabella 2
Struttura della word di stato, che rappresenta un altro campo molto importante del «Request Header», in quanto consente al driver di comunicare al DOS eventuali errori e lo stato corrente della funzione in corso di esecuzione.

seconda che si tratti di «block» o «character device».

In particolare per i primi, oltre alle ovvie INIT, INPUT ed OUTPUT, troviamo le MEDIA CHECK, BUILD BPB nonché REMOVABLE MEDIA, tutte e tre strettamente legate alla natura del dispositivo originario a cui erano destinate e cioè il dischetto: la prima serve a stabilire di che tipo è l'unità che si vuole gestire, la seconda permette di costruire il cosiddetto BPB («BIOS Parameter Block») del quale parleremo in dettaglio nel seguito, mentre la terza permette la gestione di driver relativi ad unità in cui il supporto magnetico è rimovibile (floppy, micro-floppy, nastri di back-up e CDROM), contrapposti ai «non-removable media» e cioè i dischi rigidi.

Per quanto riguarda invece i «character device» si hanno le varie routine che gestiscono lo stato sia in input che in output oltreché la loro «cancellazione» (FLUSH).

Per entrambi sono presenti anche funzioni relative all'apertura ed alla chiusura del dispositivo, che hanno un significato importante nel caso di dispositivi di tipo «character» in quanto consente di inviare comandi particolari di «inizio lavori» e di «fine lavori» ad un dispositivo: nel caso di una stampante ciò può essere utile ad inviare una stringa di caratteri di inizializzazione come primissima cosa (ad esempio settaggio del set di caratteri) nonché una stringa «di chiusura» al termine delle operazioni.

Altre funzioni in comune sono quelle riguardanti le già citate «stringhe di IOCTL» e cioè le stringhe di «I/O Control» che permettono di inviare al driver o ricevere da esso delle particolari stringhe di controllo dell'I/O, nettamente distinte da una normale stringa di dati inviata in output o ricevuta in input, tramite le altre funzioni di OUTPUT e di INPUT.

Gli errori

Facendo riferimento alla tabella 2, analizziamo in breve le possibilità di segnalazione di errore da parte di un driver:

— le prime tre riguardano nettamente un dispositivo di tipo «block» e sono relative ad un tentativo di scrittura laddove ciò non sia possibile (tacca di protezione sul dischetto...) ad un tentativo di indirizzare un'unità inesistente e al tentativo di gestire un'unità «non pronta» (un disk driver senza floppy inserito...).

— L'errore 03 è proprio quello da segnalare nel caso in cui venga richiesta l'attivazione di una routine non implementata.

— I cinque errori successivi riguardano ancora una volta un dispositivo di tipo «block» e sono relative ad un errore di CRC (utile anche per dispositivi che gestiscono porte seriali e perciò di tipo «character»), ad errori di «indirizzamento» di un certo settore sia per errata richiesta sia per errore di ricerca, ad errori di gestione di supporti di memoria non conosciuti e ad errori nel puntamento ad un settore inesistente.

— Il successivo errore riguarda ovviamente soltanto una stampante.

— Altri tre permettono la segnalazione di errori più generali, in scrittura, in lettura o più ancora generali (del tipo «non localizzato»).

— Un paio di errori riservati, tanto per non perdere l'abitudine...

— Ed infine un errore per il caso in cui c'è stato un cambiamento illecito di disco, magari nel bel mezzo della scrittura di un file.

Con quest'altra serie di informazioni terminiamo la puntata e diamo l'appuntamento alla prossima, dove parleremo più in dettaglio delle funzioni singole e di come devono essere implementate.

SOFTWARE

software originale, sigillato, con garanzia ufficiale e possibilità di aggiornamento

SPREADSHEET/INTEGRATI	
Microsoft Excel 2.0 MS-DOS (It.)	790.000
Microsoft Multiplan 3.0 (It.)	390.000
Microsoft Multiplan 2.0 XENIX	429.000
Microsoft Works (It.)	350.000
Lotus 1-2-3 Rel. 2.01 (It.)	690.000
Lotus Symphony 2.0 (It.)	950.000
Computer Associates SuperCalc 4.0 (It.)	680.000
Software Group Enable (It.)	1.090.000
Borland Quattro	350.000

IL SOFTWARE ORIGINALE NON HA IL VIRUS

Microsoft Excel Macintosh (It.)	690.000
Microsoft Works Macintosh (It.)	450.000
Lotus Jazz Macintosh (It.)	560.000

WORD PROCESSING

Microsoft Word 4.0 (It.)	750.000
Microsoft Word 3.0 LAN - 5 users (It.)	1.990.000
Microsoft Word 3.0 XENIX	890.000
Lotus Manuscript (It.)	690.000
MicroPro WordStar 4.0 (It.)	595.000
MicroPro WordStar 2000 Rel. 3.0 (It.)	890.000
Ashton-Tate Multimate Advantage II	790.000
Samna Word IV Plus (It.)	1.250.000
Satellite Word Perfect	1.190.000
Microsoft Word 3.0 Macintosh (It.)	690.000
Microsoft Word 3.0 LAN Macintosh 5 users (It.)	1.490.000

HARDCARD PLUS

Hard Disk su scheda
35 ms. tempo medio di accesso
Versione 20 MB L.1.190.000
Versione 40 MB L.1.590.000

DATABASE MANAGEMENT

Microsoft RBase System (It.)	1.090.000
Ashton-Tate Rapid File (It.)	590.000
Borland Paradox	1.190.000
Borland Reflex (It.)	290.000
Nantucket Clipper	1.190.000
Wordtech QuickSilver	1.090.000
Microsoft File Macintosh	295.000
Ashton-Tate dBASE MAC Macintosh	630.000

GRAFICI

Microsoft Chart 2.0 (It.)	395.000
Microsoft Chart 3.0	590.000
Lotus Freelance Plus (It.)	690.000
Autodesk AutoCAD Base (It.)	690.000
Autodesk AutoCAD ADE 2 (It.)	4.360.000
Autodesk AutoCAD ADE 3 (It.)	5.950.000

SPEDIZIONI GRATUITE IN TUTTA ITALIA

Ashton-Tate ChartMaster	730.000
Microsoft Chart Macintosh (It.)	280.000

DESKTOP PUBLISHING

Aldus PageMaker (It.)	1.150.000
RANK XEROX Ventura Publisher (It.)	1.390.000
Microsoft Pageview	90.000
Letraset Ready, Set, Go! 4.0 Macintosh (It.)	890.000

COMUNICAZIONI

Microsoft Access	390.000
DCA Crosstalk XVI	420.000
DCA Crosstalk Mark 4	430.000
Meridian Carbon Copy Plus	490.000

SPECIALE SOFTWARE

dBASE III Plus (It.)

(include un buono per aggiornamento a dBASE IV al prezzo di L. 200.000)

L. 1.090.000

Framework II (It.)

(include un buono per aggiornamento a Framework III al prezzo di L. 100.000)

L. 1.090.000

Computer Discount Italia®

è diventata

Quotha32

rivenditori indipendenti di hardware e software

NOVITA'!

Stampanti NEC 24 aghi in STOCK
P6 Plus, P7 Plus e P2200
con garanzia ITALIANA
TELEFONARE

LINGUAGGI

Microsoft QuickBASIC 4.0	175.000
Microsoft QuickC	195.000
Microsoft BASIC Interpreter	590.000
Microsoft BASIC Compiler	460.000
Microsoft C Compiler	595.000
Microsoft FORTRAN Compiler	595.000

CONDIZIONI AGEVOLATE PER ENTI PUBBLICI E SCUOLE richiedete i nostri preventivi!

Microsoft COBOL Compiler	990.000
Microsoft Pascal Compiler	390.000
Ryan MacFarland COBOL Full System	1.390.000
Lattice C Compiler	690.000
Borland Turbo Pascal 4.0 (It.)	250.000
Microsoft BASIC Interpreter XENIX 286	495.000
Microsoft BASIC Compiler XENIX 286	990.000
Microsoft COBOL Compiler XENIX 286	1.390.000
Microsoft FORTRAN Compiler XENIX 286	990.000
Microsoft Pascal Compiler XENIX 286	990.000
Microsoft XENIX 386 Toolkit	550.000
Microsoft BASIC Interpreter Macintosh	195.000
Microsoft BASIC Compiler Macintosh	290.000
Microsoft FORTRAN Compiler Macintosh	450.000

UTILITIES

Microsoft Windows 2 (It.)	216.000
Microsoft Windows 386 (It.)	370.000
Microsoft Windows 2 Toolkit	690.000
Norton Utilities 4.0	280.000
Norton Commander	170.000
Fastback	495.000
Software specifico/gestionale	Telefonare

RICHIEDETE PREVENTIVI PER PC/XT, AT COMPATIBILI e 386

HARDWARE

hardware originale con garanzia ITALIANA di 1 anno

PERSONAL COMPUTER

Olivetti M15, 2 FDU 720 KB, 512 KB RAM, alimentatore, cavi, borsa, manuali, MS-DOS	1.295.000
Olivetti M24, 1 HDU 20 MB, 1 FDU 360 KB, 640 KB RAM, completo di monitor e tastiera	2.650.000
Olivetti M240, 1 HDU 20 MB, 1 FDU 360 KB, 640 KB RAM, completo di monitor e tastiera	3.150.000

Olivetti M28, 1 HDU 20 MB, 1 FDU 1.2 MB, 512 KB RAM, completo di monitor e tastiera	3.850.000
Olivetti M290, 1 HDU 20 MB, 1 FDU 1.2, 2 MB RAM, completo di monitor, tastiera e MS-DOS	5.790.000
Olivetti Altri Modelli	Telefonare
PC Philips	Telefonare
PC Bit Computers	Telefonare
Portatili ZENITH	Telefonare
MONITOR	
Monitor NEC MultiSync II 14" colore 800x600	1.290.000
Monitor NEC MultiSync Plus 15" colore 960x720	1.890.000

Monitor NEC MultiSync II + scheda VEGA VGA L. 1.890.000

Monitor NEC MultiSync XL 20" colore 1.024x768	4.490.000
Monitor Hantarex Boxer 14" CGA/Herc	220.000
Monitor Colore Hantarex CT 9000 SHR EGA/CGA 14"	690.000
Sistema grafico WYSE 700 15" CRT TTL, monoc. fosfori bianchi scheda grafica ris. max. 1280x800 compatibile CGA e MGA base basculante	1.490.000

STAMPANTI/PLOTTER

Stampante PANASONIC KX-P1081, 80 col., 120 c.p.s.	495.000
Stampante PANASONIC KX-P1592, 136 col., 180/270 c.p.s.	950.000
Stampante NEC P7, 136 col., 24 aghi, 216 c.p.s. con trascinatore monodirezionale	1.495.000
Stampante NEC P9XL a colori, 136 col., 24 aghi, 400 c.p.s.	2.990.000
Altri Modelli PANASONIC in STOCK	Telefonare
Plotter Roland DXY 880/A	1.750.000
Stampanti OKI	Telefonare

Hard Disk Seagate ST225, 20MB completo di controller WD e cavi l'hard disk più venduto del mondo L. 495.000

DISK DRIVE - STREAMER

Floppy Disk Drive da 3.5" 720 KB completo di kit per alloggiamento e cavi	270.000
Floppy Disk Drive da 3.5" 1.44 MB completo di kit per alloggiamento e cavi	320.000
Hard Disk NEC D5126H 20 MB "veloce" (37 ms.)	850.000
Hard Disk NEC D5146H 40 MB "veloce" (37 ms.)	1.090.000
Streamer e sistemi di backup	Telefonare

SCHEDE - CHIP - MOUSE

Coprocessore matematico Intel 8087-1 10 MHz	470.000
Coprocessore matematico Intel 80287-10 10 MHz	650.000
Coprocessore matematico Intel 80387 25 MHz	Telefonare
Scheda Microsoft MACH 20	Telefonare
Scheda Grafica VEGA VGA	750.000
Microsoft Mouse 7 (NUOVO!)	280.000
Scheda Telefax LEXICON LEXIFAX	1.150.000
Schede INBOARD 386 Intel per PC/XT e AT	Telefonare
Schede Varie	Telefonare

VARIE

Modem Discovery 1200H su scheda, V21/V22, 300-1200 b.p.s., set esteso HAYES	330.000
Modem Discovery 2400C esterno, V22/V22B, 1200-2400 b.p.s., set esteso HAYES	750.000
Scanner ottico LEXICON LEXISCAN	650.000

DISCHETTI

Formato 3 1/2"

Micro Mito DS/DD	2.600
Sony DS/DD	3.500
Verbatim DS/DD	4.600
3M DS/DD	6.900

Formato 5 1/4"

Duratech DS/DD	1.050
Verbatim DS/DD	2.700
3M DD/DS	3.500
Mito-Mega 96TPI/DS/HD 1.2 MB	3.100
Verbatim DS/HD 1.2 MB	5.200
3M DS/HD	5.900

ordine minimo 100 pezzi

Tutti i prezzi sono al netto di I.V.A.

TERMI E CONDIZIONI DI VENDITA - Tutti i prezzi sono al netto di I.V.A.
- Spese di spedizione a carico della Quotha 32 s.r.l. - Pagamento in contrassegno con assegno circolare intestato a Quotha 32 s.r.l. o contante - Sconto del 2% per pagamento anticipato. - Ci riserviamo di accettare ordini di importo inferiore a 300.000 lire. - La merce si intende salvo il venduto. - Ulteriori sconti per quantità. - La presente offerta è valida sino al 15 novembre 1988 e sostituisce ogni nostra precedente offerta.

per ordini, informazioni o richieste di listini completi telefonare allo

055 - 22.99.851

oppure scrivere, precisando il recapito telefonico, a

Quotha32 s.r.l.

Via Accursio, 2 - 50125 FIRENZE - Telefax 055-2280674

Vuoi Scommettere?

Da anni la Computer Market propone i migliori prodotti informatici ai prezzi piú bassi d'Italia fornendo tutto il service e la competenza necessaria.

Non a caso possiamo vantare clienti quali l'Università di Roma, il Policlinico Gemelli, il Gruppo Fininvest, Ambasciate presso lo Stato Italiano e la Santa Sede, Enti Pubblici, la Democrazia Cristiana e ci scusiamo per tutti quelli che non abbiamo citato.

Noi siamo pronti a scommettere che quando ci contatterai per il tuo prossimo acquisto resterai sbalordito.

Computer Market

**P.zza San Donà di Piave 14 Roma Tel. 79.45.493
(P.zza Re di Roma. Metrò Ponte Lungo)**

Gli acquirenti che presenteranno questa pagina pubblicitaria riceveranno un grazioso omaggio

Condizioni e sconti particolari per Rivenditori