

Il Mouse

terza parte

Eccoci giunti alla terza puntata riguardante la gestione via software di un mouse, non dal punto di vista dell'utente finale (cioè di colui che utilizza il computer per eseguire programmi già predisposti al funzionamento del mouse), ma dal punto di vista del programmatore, il quale viceversa deve «inventare» il programma che dovrà, tra le tante cose, gestire anche il mouse

Abbiamo già imparato dalla scorsa puntata come si va a gestire un mouse secondo funzionalità standard di base e cioè secondo lo stretto indispensabile: dal reset del mouse, all'accensione e/o spegnimento del cursore sul video, dalla definizione dei limiti del video entro cui il cursore può agire, alla lettura istante per istante dello stato dei pulsanti nonché della posizione corrente del cursore all'interno della pagina video.

In questa puntata analizzeremo altre funzioni molto utili, diciamo così di «secondo livello»: nella tabella di pagina 236 abbiamo come di consueto riportato un quadro sinottico delle funzioni di gestione del mouse.

La funzione 4: Set Mouse Position

Analizzando la figura 1, vediamo che la funzione in esame serve a predisporre la posizione all'interno della pagina video del cursore, nel caso in cui necessità di programmazione debbano far sì che il cursore si trovi proprio in una posizione ben prestabilita: in particolare, prima della chiamata alla funzione 4 (per i particolari rimandiamo alle puntate precedenti), bisogna porre nei registri CX e DX rispettivamente la posizione

del cursore espressa in «pixel», e ciò anche se siamo in modo testo e perciò saremmo portati a ragionare in termini di «righe-colonne».

Il registro dovrà contenere dunque la coordinata orizzontale, mentre DX quella verticale: il tutto ancora una volta in modo veramente molto semplice.

La funzione 5: Get Button Press Info

Con questa funzione (si veda la figura 2) noi abbiamo la possibilità di testare la «pressione» di uno dei due o tre pulsanti del mouse, registrando la posizione del mouse stesso, proprio nel momento in cui il pulsante è premuto: ciò è leggermente diverso dal comportamento della funzione 3 (Get Mouse Position & Button Status), della quale abbiamo parlato la scorsa puntata.

Infatti quest'ultima funzione legge la posizione del mouse e lo stato dei pulsanti nel momento in cui avviene la chiamata alla funzione stessa (o giù di lì, considerati i vari tempi interni sui quali non c'è possibilità di discernere più di tanto), e perciò magari, quando tali coordinate vengono effettivamente utilizzate, il mouse si trova in tutt'altra posizione (e ciò potrebbe succedere con

AX = 4H	Set Mouse Position
INPUT	CX = cursor x position DX = cursor y position
OUTPUT	-

Figura 1 - Questa funzione serve per predisporre le coordinate orizzontali e verticali del cursore, all'interno della pagina video: sono entrambe espresse in «pixel».

AX = 5H	Get Button Press Information
INPUT	BX = button
OUTPUT	AX = button status BX = count of button presses CX = cursor x position at press DX = cursor y position at press

Figura 2 - La funzione in questione fornisce la posizione del cursore, lo stato dei pulsanti nonché il numero di volte in cui è stato premuto un pulsante prescelto.

linguaggi non particolarmente veloci, quali il ben noto BASICA).

Invece la funzione di esame «congelata» lo stato del mouse nell'istante in cui è stato premuto un pulsante prestabilito, contando pure quante volte il pulsante è stato premuto dalla precedente chiamata alla stessa funzione: il tutto solo «dopo» che è stata attivata la funzione ponendo in AX il valore 5 ed in BX un valore a seconda del pulsante che bisogna tenere d'occhio. In particolare per testare il pulsante sinistro bisogna porre 0 in BX, mentre per il pulsante destro il valore è 1; per il pulsante centrale (senza un minimo di rigore logico però) il valore da porre in BX è invece 2.

In uscita otterremo, nei soliti registri CX e DX la posizione del cursore all'atto dell'ultima pressione del pulsante desiderato (abbiamo detto «del pulsante» e non «dei pulsanti», come invece sarebbe stato auspicabile), mentre in BX avremo il numero di volte che il pulsante in esame è stato premuto: infine in AX avremo lo stato dei pulsanti, e cioè un bit settato o meno a seconda dello stato dei due o tre pulsanti, così come abbiamo visto nella scorsa puntata per la funzione 3.

Almeno il tutto in teoria, secondo quanto riportato dal manuale di «mamma Microsoft», in quanto facendo viceversa girare il programmino di figura 3, abbiamo riscontrato un funzionamento strano: per vedere di cosa si tratta vediamo cosa fa il programma in questione.

In particolare, dopo aver inizializzato il mouse e reso visibile, si entra in un loop dal quale si esce solo se la variabile BX è maggiore di 0.

All'interno del loop abbiamo posto dapprima un ritardo (delay) di 2 secondi durante i quali l'altoparlantino interno suonerà una frequenza di 100 Hz: successivamente c'è la chiamata alla routine in esame, dopo aver posto nel registro BX (la variabile **reg. BX**) il valore 1, per mezzo del quale testiamo il pulsante destro.

Sappiamo già che in **reg. BX**, e successivamente nella variabile **BX**, verrà posto il numero di volte che il pulsante designato è stato premuto.

Proseguendo nell'analisi troviamo un altro ritardo (delay) di 2 secondi durante i quali l'altoparlantino emetterà una frequenza di 1 kHz, seguito infine da un ultimo delay di altri 2 secondi in cui l'altoparlantino resta muto.

Tutto questo è stato fatto per poter avere in ogni istante la sensazione immediata di quale istruzione del ciclo fosse correntemente in esecuzione, in

```

uses graph,crt,dos;

var gd,gm : integer;
    reg : registers;
    BX : integer;
    ch : char;

procedure mouse(ax : word);
begin
    reg.AX := ax;
    intr($33,reg);
end;

begin
    gd := detect;
    initgraph(gd,gm,'');
    mouse(0);
    mouse(1);
    repeat
        sound(100);
        delay(2000);
        nosound;
        reg.BX := 1;
        mouse(5);
        BX := reg.BX;
        sound(1000);
        delay(2000);
        nosound;
        delay(2000);
    until BX > 0;
    mouse(2);
    closegraph;
    writeln(BX);
    ch := readkey;
end.

```

Figura 3 - Programmino di prova della funzione 5 di gestione del mouse.

particolare per sapere in quali istanti dobbiamo premere il pulsante destro del mouse.

In definitiva dal momento in cui lanciamo il programma avremo una sequenza di 2 secondi di un tono grave, 2 secondi di un tono acuto e 2 secondi di

silenzio, che prosegue all'infinito a meno che non premiamo il pulsante destro.

Premendo dunque tale pulsante anche più volte durante la nota acuta (e cioè dopo che è stata attivata la funzione 5), sentiremo dunque i due secondi di pausa dopodiché ci aspetteremo l'uscita dal loop in quanto abbiamo premuto il pulsante destro.

Invece succede che, finita la pausa, sentiamo di nuovo la nota grave, poi la nota acuta e poi ancora il silenzio, dopodiché finalmente usciremo dal ciclo, ed allora avremo sullo schermo l'indicazione del numero di volte in cui abbiamo premuto il pulsante.

In definitiva si è verificata la seguente sequenza di eventi, della quale ci si poteva accorgere in qualunque altra maniera (ad esempio mostrando sui video qualcosa di mutevole, mentre abbiamo preferito un'indicazione "sonora"):

- 2 secondi (di nota bassa)
- prima attivazione della funzione 5
- 2 secondi (di nota acuta) durante i quali abbiamo premuto più volte il pulsante
- 2 secondi di silenzio
- 2 secondi di nota grave
- seconda attivazione della funzione 5
- 2 secondi di nota acuta in cui non è successo niente
- 2 secondi di silenzio
- l'uscita dal loop con stampa su vi-

AX = 6H	Get Button Release Information
INPUT	BX = button
OUTPUT	AX = button status BX = count of button releases CX = cursor x position at release DX = cursor y position at release

Figura 4 - Questa funzione è analoga alla funzione 5, ma è riferita al rilascio di un pulsante, anziché alla sua pressione.

AX = 0BH	Read Motion Counters
INPUT	-
OUTPUT	CX = horizontal count DX = vertical count

Figura 5 - Questa funzione permette di valutare di quanto si è spostato il cursore dall'ultima chiamata a questa stessa funzione.

AX = 0FH	Set Mickey/Pixel Ratio
INPUT	CX = horizontal ratio DX = vertical ratio
OUTPUT	-

Figura 6 - Questa funzione consente la variazione della sensibilità del mouse intesa come rapporto tra lo spostamento «fisico» sul piano d'appoggio ed il numero di pixel di cui si vuole muovere corrispondentemente il cursore sullo schermo.

deo del numero di volte che abbiamo premuto il pulsante.

Da questo possiamo dedurre che la funzione 5 non funziona così com'è «da sola», ma abbinata ad un'altra chiamata, tra le quali chiamate verrà effettuato il conteggio delle pressioni del pulsante e la registrazione della posizione all'atto dell'ultima pressione del pulsante, «prima» che si arrivi alla seconda chiamata.

Il sopraccitato manuale (che poteva viceversa essere un po' più chiaro in merito, mentre decisamente porta fuori strada) dice tra l'altro che all'atto della chiamata alla funzione 5 il contatore di «pressioni» viene azzerato («ettecredo», altrimenti che conteggio sarebbe!!!), ma non dice come e quando tale valore possa essere letto: abbiamo dunque «scoperto» che questo valore (rigorosamente nullo nell'intervallo tra la prima e la seconda chiamata) diventa maggiore di zero solo dopo la seconda chiamata, riflettendo così il numero di volte che è stato premuto il pulsante desiderato.

Tra l'altro tale valore (citiamo sempre il manuale) può arrivare fino ad un massimo di 32767, dopodiché non viene gestito l'overflow ed improrogabilmente ricomincia daccapo da 0.

Ciò è un grave problema perché non si riescono a leggere così le 40000 volte che il povero programmatore ha dovuto premere il dannato pulsante prima di accorgersi che non era uscito dal loop, ma viceversa era stato abbandonato dal mouse, incapace di reggere tanto stress...

A parte gli scherzi dunque la funzione 5 «apre una parentesi», richiusa da una successiva chiamata alla stessa funzione, all'interno delle quali parentesi vengono contate le pressioni di un tasto prestabilito (senza overflow!!!) ed infine viene congelata la posizione del cursore nello schermo, nell'istante dell'ultimissima pressione del pulsante.

Ciò come detto è solamente riportato «fra le righe» nel manuale della Microsoft e porta decisamente fuori strada l'esempio (assai scarno, in BASICA: 3 righe di programma) in cui è riportata una sola chiamata a tale funzione e non una coppia tra le quali andare a fare i propri conti...

La funzione 6: Get Button Release Info

Ci vuole poco per capire che questa funzione è esattamente la duale della (famigerata...) precedente: quello che cambia è il fatto che ora ha interesse il rilascio di un pulsante definito invece che la sua pressione, fermi restando il

AX = 0H	Mouse Reset
AX = 1H	Cursor Enable
AX = 2H	Cursor Disable
AX = 3H	Get Mouse Position and Button Status
AX = 4H	Set Mouse Position
AX = 5H	Get Button Press Information
AX = 6H	Get Button Release Information
AX = 7H	Set min & max horizontal position
AX = 8H	Set min & max vertical position
AX = 9H	Set Graphic Cursor Block
AX = 0AH	Set Text Cursor
AX = 0BH	Read Motion Counters
AX = 0CH	Set User-defined subroutine
AX = 0DH	Enable Light Pen Emulation
AX = 0EH	Disable Light Pen Emulation
AX = 0FH	Set Mickey/Pixel Ratio
AX = 10H	Window Conditional Off
AX = 12H	Set Large Graphic Cursor
AX = 13H	Set Speed Threshold

conteggio del numero di «click» (in gergo si chiamano così le «pressioni») tra una chiamata e la successiva nonché la memorizzazione della posizione del cursore in corrispondenza dell'ultimo «rilascio» del pulsante oramai celebre.

La funzione 0BH: Read Mouse Motion Counters

Anche questa è una funzione «parentetica», intendendo con tale termine una funzione che agisce in coppia con un'altra chiamata alla medesima routine e consente di calcolare di quanto si è spostato il cursore tra una chiamata e la successiva: i valori forniti in output (rispettivamente nei registri CX e DX) rappresentano due quantità legate al moto orizzontale e verticale del cursore, con valori che vanno da 0 a 32767 se positivi e fino a -32768 se negativi e sono espressi in un'unità di misura folkloristicamente (e americanamente...) chiamata «Mickey», nome che come ben noto negli USA è imprensabilmente legato a «Mouse». Per l'appunto...

Tale unità è legata ad uno spostamento del mouse sul piano d'appoggio ed al tempo richiesto per lo spostamento: 1 Mickey equivale a circa 1/200 di pollice (chissà quali calcoli ci sono sotto...) e per l'appunto la funzione in esame fornisce il conto di quanti Mickey si è spostato il mouse «fisico» nelle due direzioni, rispetto all'ultima chiamata.

Fatti i conti, 1/200 di pollice equivalgono a circa 0.127 millimetri, per cui il massimo conteggio di Mickey previsto (32767) significa che il mouse si è sposta-

to la bellezza di oltre 4 metri... (melius abundare quam deficere...).

Attenzione che anche in questo caso non viene gestito alcun tipo di overflow, tanto positivo che negativo!!!

Comunque esiste un significato meno «goliardico» del Mickey, legato alla funzione che ora andiamo ad analizzare.

La funzione 0FH: Set Mickey/Pixel Ratio

Si tratta dell'ultima funzione che andiamo ad analizzare in questa puntata: essa consente di cambiare la sensibilità allo spostamento del mouse, intesa come rapporto tra lo spazio percorso sul piano d'appoggio (tenendo anche conto del tempo impiegato per effettuare tale spostamento) e lo spostamento sullo schermo video del cursore.

In particolare i rapporti Mickey/Pixel per l'orizzontale e per la verticale vanno posti rispettivamente in CX e DX e rappresentano in realtà il numero di Mickey necessari a far muovere il cursore sullo schermo di 8 pixel.

Tali rapporti, se non modificati per mezzo di questa funzione, vengono già per default prefissati a dei valori ben precisi: per quanto riguarda lo spostamento orizzontale si hanno in genere 98 Mickey per 8 pixel, mentre per la direzione verticale ci vogliono 16 Mickey per percorrere 8 pixel verticali sul video.

Con questi concetti alquanto strani terminiamo la puntata e rimandiamo alla prossima dove parleremo del cursore in grafica ed in modo testo e di altre funzioni alquanto complesse, ma utili.

Quadro sinottico delle funzioni di gestione di un mouse per mezzo dell'INT 33H: ad eccezione della funzione 12H, sono tutte funzioni standard che sono implementate nel file MOUSE.COM che accompagna il mouse.

CORRISPONDENZA, SEMPRE

Tra noi e voi, oltre la vendita

Assistenza tecnica con HOTLINE telefonica su tutta la gamma dei nostri prodotti: e' un servizio

**PREZZI
IVA
INCLUSA**



**RICHIEDI IL
CATALOGO
GENERALE**



BABY AT 286 EGA Cod. X18 Lire 2.400.000
AT Baby Compact, tastiera 101 tasti, Turbo 12 Mhz 0 Wait state, mainboard 1/4 Mb VLSI, 512 K Ram 100 ns, Controller HD + FD Western Digital, 1 Floppy Chinnon 5.25 1.2 Mb, HD Seagate 20 Mb, scheda grafica EGA 640 x 480, monitor monocromatico bifrequenza autowitching.

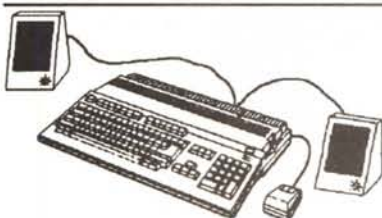
AT 286 PORTABLE Cod. X19 Lire 2.920.000
Caratteristiche analoghe al modello precedente (X18), con schermo LCD ed interfaccia per video esterno; 6 slot di espansione; HD 20 Mb, floppy 1.2 Mb, 512 K on board.

SUPEROFFERTA DEL MESE: STAMPANTE H136 Cod. X20 Lire 590.000
Buffer di 8 KBytes, emulazione IBM Proprinter, 138 colonne, 180 CPS, interfaccia Centronics (COMPATIBILE AMIGA), cavo parallelo IN OMAGGIO.

SCHEDA MUSICALE Cod. P93 Lire 590.000
Musio Synthesizer Card: 12 canali stereo polifonici, 32 strumenti preselezionati, fornita con 5 dischetti di software, programmabile con i piu' comuni linguaggi. Possibilita' di ascoltare in multitasking la musica composta; abbinabili ai favolosi DIGIBOX per Amiga (Cod. A04).

I.C. TESTER Cod. P26 Lire 520.000
Dispositivo indispensabile per verificare il corretto funzionamento dei chip delle seguenti famiglie: RAM, TTL, CMOS. Il software di gestione permette di eseguire test personalizzati su qualsiasi tipo di integrato.

INSTALLIAMO RETI SOFTWARE ED HARDWARE, SISTEMI 286-386 COMPLETI; ASSISTENZA TECNICA SPECIALIZZATA; PREVENTIVI DI AUTOMAZIONE D'UFFICIO. SISTEMI GRAFICI E DESKTOP PUBLISHING SU MS DOS ED AMIGA. CONSULENZE SU QUALSIASI SISTEMA DI TRASMISSIONE DATI.



NOVITA' PER AMIGA DIGIBOX Cod. A04 Lire 59.000
I DIGIBOX sono una coppia di minicasse amplificate collegabili direttamente alle uscite audio sinistra e destra di ogni modello Amiga; ognuna delle due casse e' munita di un proprio interruttore e regolatore di volume, e di un Led di accensione. Gli altoparlanti sono Larga Banda (40-12.000 Hz) del diametro di 6 cm.; le dimensioni sono estremamente contenute (7 x 10 x 7 cm - L x H x P) e non pongono alcun problema per il posizionamento. I DIGIBOX sono alimentabili con delle semplici pile a 9 Volts o tramite alimentatore DC esterno (non forniti). Queste piccole casse migliorano sensibilmente la qualita' della riproduzione sonora di Amiga ed hanno una resa eccellente malgrado le ridotte dimensioni. Nella confezione sono inclusi i cavi di collegamento ad Amiga ed un pratico treppiede di sostegno al quale possono essere fissati i DIGIBOX.



SCHEDA A 2088 JANUS XT Cod. A31 Lire 800.000
Scheda di compatibilita' Amiga-IBM Xt, microprocessore Intel 8088, 512 Kb RAM, controller per floppy e scheda grafica CGA "on board", floppy 5.25 da 380 K, Ms Dos 3.2 originale.

NOVITA': TAVOLA GRAFICA PER AMIGA 2000 Cod. A41 Lire 890.000
Eccellente strumento di precisione: risoluzione 1024 x 1024 su una matrice di 23 x 33 cm, due pulsanti per emulazione mouse, manuale in Italiano, drivers PAL ed NTSC; il software permette di impostare la riduzione in scala del disegno prima ancora del trattamento da parte del programma disegnatore (De Luxe Paint, X-Cad, Aegle Draw 2000...); l'interfacciamento con Amiga avviene tramite una scheda, permettendo cosi' l'uso contemporaneo del mouse. Disponibile anche per i modelli A500 ed A1000.

PENNA OTTICA Cod. A01 Lire 49.000
Lavora in emulazione mouse: potrete scegliere i menu direttamente sullo schermo, disegnare sullo schermo, lavorare con tutti i programmi che supportano il puntatore standard.

HARD DISK A500 Cod. A11 Lire 890.000
HD esterno con controller ed alimentatore, 20 Mb, superveloce, autoboot con Kickstart 1.3; possibilita' di installare nello chassis una espansione di memoria da 2 ad 8 Mb. In OMAGGIO un software di backup ed un utile File Manager. Disponibile anche nella versione interna per A2000, con le stesse caratteristiche, a Lire 720.000.

RICHIEDETE ALLA "DIGIMAIL VIDEOSERVICE - VIA CORONELLI 10, 20146 MILANO" IL VASTISSIMO CATALOGO DI VIDEOCASSETTE ORIGINALI, INVIANDO LIRE 2000 PER LE SPSE DI SPEDIZIONE: VIDEO MUSICALI, DOCUMENTARI, CARTOONS, FILM, COMODAMENTE A CASA VOSTRA IN BREVE TEMPO. UNA OCCASIONE DA NON PERDERE ASSOLUTAMENTE, UNA INTERA VIDEOTECA AL VOSTRO SERVIZIO CON UN SEMPLICE COLPO DI TELEFONO O DUE RIGHE SU UN TAGLIANDO.

VENDIAMO PER CORRISPONDENZA

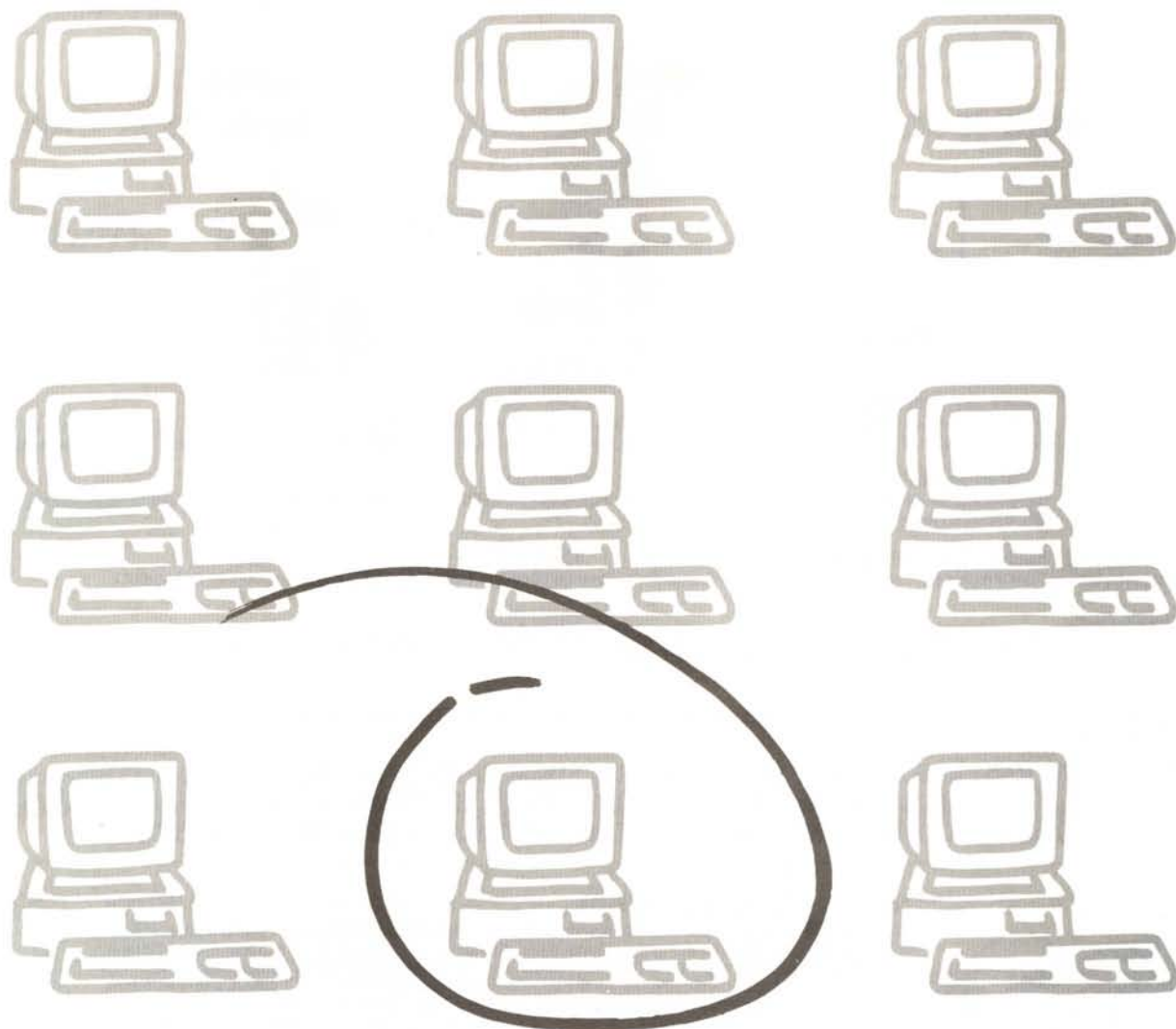
Ordinare e' semplice: basta spedire il tagliando a DIGIMAIL s.r.l. Via Coronelli 10 - 20146 MILANO oppure telefonare allo 02/426559 dalle ore 10 alle 20. Spedizione contrassegno con spese postali a carico del destinatario. Imballo gratuito. Spedizione a mezzo corriere su richiesta.

ORDINE MINIMO LIRE 30.000.

Non dimenticate di richiedere il nostro catalogo **HARDWARE e SOFTWARE ORIGINALE** comprendente centinaia di prodotti e programmi.

by DM-dtp

NOME		COGNOME		MC79
VIA		N.	CAP	
CITTA'		PROV.		
Firma				
<input type="checkbox"/> Allego Lire 1.000 per ricevere il catalogo Software originale. <input type="checkbox"/> Allego Lire 2.000 per ricevere il catalogo Videocassette originali. <input type="checkbox"/> Ho un videoregistratore <input type="checkbox"/> SI <input type="checkbox"/> NO				
DESCRIZIONE	COD.	Q.ta	PREZZO	
Il mio computer e' un (marca e tipo):			TOTALE	



DISCOM

Da sempre Discom è preparata per correre e vincere. La sua professionalità e il suo dinamismo fanno della Discom una società di distribuzione tra le più trainanti: le proposte più adeguate e i prezzi più competitivi per i prodotti vincenti, cioè i migliori, per Voi.

00128 Roma - Via Marcello Garosi, 23

Telef. (06) 52.07.839-52.07.917-52.02.293 - Telex 620238 - Telefax (06) 52.05.433

SOLO I MIGLIORI. PER VOI.