

Come sempre accade il miglior software esce quando ormai l'hardware è obsoleto (o comunque superato), ecco così fiorire ottimi programmi e routine interessanti proprio ora che l'MS DOS dovrebbe andare in pensione (almeno così vogliono farci credere).

Noi utenti imperterriti e smanettoni invece aspettiamo proprio questi annunci per dimostrare che l'MS DOS è più vivo che mai e, in attesa della versione numero 4, eccovi un bel po' di routine da aggiungere al Vostro Sempre Amato

Salva-video

di Rocco Bertè - Acireale (CT)

Questo breve programma, scritto in Assembler (MASM), si installa in memoria provvedendo a disattivare il video dopo 3 minuti d'inattività dello stesso o della tastiera.

L'idea nasce quando, lavorando per la prima volta su di un terminale di un sistema multistazione, mi sono accorto che il video dopo qualche minuto di inattività si autospegneva. Dopo essermi ripreso dallo choc che l'avvenimento mi aveva provocato (provengo dal mondo dei PC) il primo pensiero è stato quello di fornire al mio PC questa caratteristica. Purtroppo dovendo emulare via software questa possibilità piuttosto sofisticata, sono stato costretto ad utilizzare il linguaggio macchina. Anche per ottenere un codice molto compatto ed efficiente, quindi armatomi di MASM e di WS sono partito per il mondo impervio e senza ritorno dell'Assembler. Un doveroso riconoscimento va ad

Edward J. Batutis, perché grazie allo studio di un suo programma pubblicato su BYTE sono riuscito a risolvere un problema dell'intercettazione degli interrupt.

Prima di passare ad analizzare come tutto ciò avviene è mio dovere avvertire coloro che dopo aver copiato le poche righe di sorgente e aver superato indenni gli esami del MASM di non preoccuparsi più di tanto quando il linker risponderà con: «Warning: no stack segment», è tutto in ordine solamente che il povero linker non sa che abbiamo appositamente trascurato di definire il segmento di stack per poter ottenere

dall'EXE2BIN il file .COM. Infatti dopo aver salvato il sorgente con l'estensione .ASM, bisogna lanciare i seguenti comandi:

```
masm <nomefile>
link <nomefile>
exe2bin <nomefile>.exe <nomefile>.com
```

Dopo si possono tranquillamente cancellare i file .OBJ e .EXE.

Commento al programma

Dopo aver definito la macro DOSINT, che chiama il servizio del DOS corrispondente al numero passato come parametro (int 21h), ed impostato l'ori-

Salva-video

Per provare questo programma l'ho caricato su tutte le macchine della redazione. La prima reazione l'ho avuta direttamente dagli AT, su di questi la routine non funziona, anzi, cancella la RAM Cmos che contiene la configurazione; dopo aver riavviato il sistema con i diagnostici avanzati mi era passata la voglia di continuare gli esperimenti per scoprire cosa succede agli AT.

La seconda reazione l'ho avuta dagli umani che vivono davanti al video; alcuni di loro hanno protestato perché tre minuti sono troppo pochi! Mi hanno detto che proprio mentre fissavano la riga Basic che era andata in errore questa gli spariva rompendo la concentrazione. Se anche voi avete degli amici che in fase di concentrazione superano i tre minuti ecco come modificare direttamente il file SAVID.COM per aumentare, o diminuire, a piacere il time-out.

1 Calcolare quanti diciottesimi di secondo corrispondono al time-out desiderato.

2 Convertire il valore trovato in esadecimale.

3 Entrare nel SAVID.COM con il DEBUG

DEBUG SAVID.COM

modificare il contenuto delle locazioni 67B e 67C

battere la parte bassa del valore esadecimale e lo spazio

poi la parte alta e il return

salvare il tutto con

— W

Esempio:

15 minuti = 900 secondi = 16.200 diciottesimi.

16.200 = &H3F48 HH = 3F LL = 48

(in neretto le scritte del computer)

C> DEBUG SAVID.COM

—E67B <RETURN>

xxxx:067B CC.

battere 48 <spazio>

48 **0C.**

battere 3f <return>

—W

Writing 0170 bytes

—Q

C>

VDD

È disponibile, presso la redazione, il disco con i programmi pubblicati in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 249.

gine del programma a 100h per poter ottenere un file .COM, il programma salta alla parte di installazione in memoria, leggendo (grazie ai servizi DOS 35h e 25h che rispettivamente leggono e scrivono nella Interrupt Vector Table) l'indirizzo della routine che gestisce l'interrupt generato dalla pressione di un qualsiasi tasto e se questo indirizzo è uguale all'indirizzo della nuova routine (evidentemente state cercando di installare più di una volta il programma) il programma abortisce visualizzando un messaggio di non installazione.

Diversamente se l'esito è negativo, la routine legge e conserva gli indirizzi delle routine di gestione video e tastiera (int 9 e int 10h), legge il modo del video corrente e se valido controlla il tipo di scheda installata, se la scheda è una CGA inserisce nella variabile PORTA il valore appropriato altrimenti lascia il valore di default valido per le schede video monocromatiche.

Alla label START: si ha la scrittura nella Interrupt Vector Table degli indirizzi delle nuove routine, completando con un int 27h che termina il programma, ma lascia residente in memoria la parte di codice compresa tra l'inizio e la label

END_OF_RESIDENT_CODE

Tornati al prompt del DOS finisce l'installazione del programma mentre per le tre routine appena installate inizia il lavoro e precisamente:

le routine che intercettano gli interrupt 9 e 10h quando vengono chiamate non fanno altro che chiamare le vecchie routine, al ritorno azzerano il contatore del tempo trascorso e settano il flag di video acceso ON = 1;

la routine che viene eseguita dall'interrupt 1ch (circa 18 volte al secondo), dopo aver conservato i flag che utilizza ed indirizzato il segmento dei dati in quello del codice (vedi precedenti numeri di MC rubrica «Assembler 8088/86») testa la variabile ON che se contiene zero sta a significare che il video è spento e deve rimanere tale, altrimenti controlla se la variabile OFF = 0 (il video è acceso) e va ad incrementare il contatore del tempo controllando che non sia superiore a 0cch (tre minuti) in caso positivo provvede a disattivare il video mandando alla porta di I/O del video un byte a zero che disabilita appunto la visualizzazione di quanto presente nella memoria video. Se il video fosse spento (variabile OFF = 1) e deve essere acceso dopo aver letto il modo video, il valore viene usato come indice per mandare il byte opportuno alla porta video. Il tutto termina dopo aver ripristinato i registri con un bel IRET.

PG151

di A. Bruciamonti - S. Maria della Versa (PV)

Sono il felice possessore di una STAMPANTE...

Vi chiederete cosa ci sia di strano ad avere una stampante... È presto detto: dopo complicati calcoli basati sulla diminuzione cronica del contenuto del mio portafoglio ho potuto stabilire che il mio

consumo di carta in modulo continuo è circa pari a quello di un centro elaborazione dati di medie dimensioni.

Dopo molte meditazioni, sono giunto alla conclusione che avrei realizzato un risparmio non indifferente utilizzando anche la seconda faccia del modulo continuo.

Scartata l'idea (banale...) di prendere ciascun foglio e girarlo, ho optato per l'unica soluzione ragionevole: stampare

Salva-video

```
;Inizio definizione macro
dosint      MACRO  function
            mov    ah,function
            int    21h
            ; mette in ah la funzione da chiamare
            ; Chiama interrupt dos
            ENDM
;Fine definizione macro
;
;
; INIZIO PROGRAMMA
;
cseg        segment para public 'code'
            assume cs:cseg,ds:cseg
            org    100h
            ; Inizio per file .COM
;
savid      proc
            jmp    install
            ; Salta alla routine di installazione
;
; area dati residente
;
copyright  db      'SALVA-VIDEO (c) Copyright 1987'
            db      ' - RoccoSoft.',10,13,'$'
old_int9_vector label dword ; Indirizzo vecchio interrupt di
old_int9_offs  dw      ?      ; gestione tastiera
old_int9_seg   dw      ?
old_int10_vector label dword ; Indirizzo vecchio interrupt di
old_int10_offs dw      ?      ; gestione video
old_int10_seg  dw      ?
cont         dw      0        ; Contatore tempo trascorso
on           db      1        ; Interruttore video acceso: vero
off          db      0        ; Interruttore video spento: falso
porta       dw      3b8h     ; Indirizzo porta video (mono)
;
valore      db      2ch,28h,2dh,29h,2eh,2ah,3eh,29h
            ; Valori per i 7 modi video gestiti
;
; Fine area dati residente
;
; Inizio Routine
;
new_int9:
            ; Nuova routine interrupt tastiera
            pushf
            call cs:old_int9_vector
            ; Chiamata alla routine standard
;
            mov  cs:cont,0
            mov  cs:on,1
            ; Azzerà il contatore tempo trascorso
            ; Setta lo switch di accensione video
;
            iret
            ; Ritorna dall' interrupt
;
new_int10:
            ; Nuova routine interrupt video
            pushf
            call cs:old_int10_vector
            ; Chiamata alla routine standard
;
            mov  cs:cont,0
            mov  cs:on,1
            ; Azzerà il contatore tempo trascorso
            ; Setta lo switch di accensione video
;
            iret
            ; Ritorna dall' interrupt
;
new_int1c:
            ; Routine agganciata all' interrupt
            ; 1ch viene eseguita ogni clock-tic
            ;
            ; Salva i registri utilizzati
            ;
            mov  ax,cs
            mov  ds,ax
            ; Indirizza il data seg. nel code seg.
;
;
            cmp  on,0
            jz   fine
            ; Se il video e' spento ...
            ; termina
            cmp  off,1
            ; Se il video e' acceso ...
            jnz  incr
            ; va ad incrementare il tempo
            mov  off,0
            ; Il video e' spento e viene acceso
            mov  ax,0f00h
            int  10h
            ; Servizio 0fh dell' int 10h
            xor  bx,bx
            ; ritorna il modo video
            mov  bl,al
            ;
```

prima le pagine dispari e, dopo aver girato il blocco di fogli, stampare quelle pari...

Con un po' di immodestia posso dire che l'idea è semplice, efficace ed originale, dato che non ho mai visto niente di simile su nessuno dei computer che utilizzo normalmente.

I problemi sono cominciati quando si è trattato di passare al programma.

Scartato il Basic ed eliminato anche il

Fortran, poco adatti ad un programma simile e per di più lentissimi sul mio computer, mi sono rivolto al C Language che soddisfa in pieno tutte le richieste.

La versione di «PG151» (questo è il nome del programma...) scritta in «C» si è rivelata estremamente veloce e compatta.

«PG151» non è particolarmente curata dal punto di vista stilistico, ma è

stato migliorato nel corso dei mesi con l'aggiunta di opzioni particolari che permettessero la stampa di file generici.

Una parte dello «stile» è stata poi «sacrificata» per rendere il programma utilizzabile sia per la stampa di testi ASCII che di testi tipo WordStar.

Purtroppo, per non complicare troppo il programma, ho dovuto sottostare ad alcune limitazioni; infatti «PG151» non è in grado di elaborare correttamente tutti gli effetti speciali permessi da WordStar, come sottolineature, grassetto, ribattuti e così via.

Comunque non ci sono problemi, gli effetti speciali vengono trascurati senza che incidano minimamente sulla stampa.

In MS-DOS, «PG151» permette anche di stampare file con linee più lunghe di 80 caratteri, dato che provvede automaticamente a settare la stampante in compresso a 132 colonne, quando è necessario.

«PG151» si usa così

pg151 [+ -] lmm [cpd] file

dove «pg151» è il nome del comando, e parti tra [...] indicano opzioni varie che si possono anche evitare mentre «lmm» sono 4 numeri in sequenza che indicano il numero di linee per pagina e il numero di spazi da porre come margine sinistro nelle pagine dispari (il programma provvederà a calcolare il margine relativo alle pagine pari in modo che vengano stampate correttamente...).

Ad esempio:

pg151 5508 pippo

stamperà il file «pippo» utilizzando 55 linee per pagina e un margine di 8 spazi utile ad esempio per rilegare i fogli oppure per forarli e metterli in un raccoglitore.

Alla fine di ogni pagina, il programma ne stampa il numero ed invia un form feed in modo che la stampa sia corretta su tutti i tipi di stampante, però per formattazioni particolari, è possibile evitare questa possibilità ponendo un «+» all'inizio della stringa di controllo, ad esempio così:

pg151 +6610 pippo

stampando «pippo» con 66 linee per pagina, 10 spazi di margine e utilizzando la paginazione «assoluta», cioè lasciando il file esattamente com'è, senza stampare form feed alla fine delle pagine né numerarle.

Ho reso possibile questa opzione per-

```

mov     al,byte ptr valore[bx] ; Mette il AL il valore adeguato al
dx,porta
out     dx,al                  ; modo del video
; Accende il video
incr:
add     cont,1                 ; incrementa il contatore del tempo
cmp     cont,0ccch            ; Se inferiore a 3 minuti ...
jb     fine                    ; termina altrimenti spegne il video
mov     cont,0                 ; Azzerà il contatore
mov     on,0                   ; Video acceso: falso
mov     off,1                  ; Video spento: vero
mov     dx,porta
mov     al,0
out     dx,al                  ; Spegne il video
;
;
fine:
pop     dx                     ; Ripristina i registri
pop     bx
pop     ax
pop     ds
;
;
iret                                     ; Ritorna dall' interrupt
;
END_OF_RESIDENT_CODE LABEL BYTE ; Fine parte residente
banner db 'SALVA-VIDEO installato.',10,13,'$'
banner1 db 'SALVA-VIDEO non installato.',10,13,'$'
install:
mov     al,9                   ; Legge indirizzo interrupt 9
dosint 35h
cmp     bx,offset new_int9     ; Lo compara con offset routine
je     no_install              ; Se uguale termina
mov     old_int9_offs,bx       ; Conserva l' indirizzo letto
mov     old_int9_seg,es
mov     al,10h                 ; Legge indirizzo interrupt 10h
dosint 35h
mov     old_int10_offs,bx      ; Conserva l' indirizzo letto
mov     old_int10_seg,es
mov     dx,offset copyright    ; Visualizza copyright
dosint 9
mov     dx,offset banner       ; Visualizza installazione avvenuta
dosint 9
mov     ax,0f00h               ; Legge modo video
int     10h
cmp     al,7
jg     no_install              ; Se modo non ammesso termina
je     start
mov     porta,3d8h             ; Muove indirizzo per CGA
; Installazione routine
start:
mov     al,1Ch
mov     dx,offset new_int1c    ; Aggancia routine all' interrupt 1ch
dosint 25h
mov     al,9
mov     dx,offset new_int9     ; Nuovo indirizzo interrupt tastiera
dosint 25h
mov     al,10h
mov     dx,offset new_int10    ; Nuovo indirizzo interrupt video
dosint 25h
mov     dx,offset END_OF_RESIDENT_CODE+1 ; Zona residente
INT     27H                    ; Termina ma resta in luogo
no_install:
mov     dx,offset copyright    ; Visualizza copyright
dosint 9
mov     dx,offset banner1      ; Messaggio di non installazione
dosint 9
dosint 4ch                      ; Ritorno al DOS
savid  endp
cseg  ends
end     savid

```

ché molti file Unix hanno le pagine già numerate e contenenti il corretto numero di linee, perciò non è necessario elaborarle ulteriormente.

Se si desidera solo escludere la numerazione delle pagine, ma lasciare form feed come separatori di pagina, è possibile porre un «-» sulla stringa di controllo, così:

```
pg151 -5508 pippo
```

in modo da stampare il solito file quando questo ha già le pagine numerate, ma con un numero di linee per pagine diverso da quello standard che è 66.

Quando «pippo» contiene già form feed come separatori di pagina, ed è quindi necessario evitare di stamparli due volte, è possibile utilizzare l'opzione «+» unita ad un numero di linee per pagina certamente superiore a quello effettivo, in modo che «PG151» prenda appunto gli «ff» come separatori:

```
pg151 +9910 pippo
```

risolve questo caso piuttosto anomalo, permettendo di stampare «pippo» utilizzando i form feed interni come separatori di pagina, senza tenere conto della effettiva lunghezza di queste ultime.

Le altre opzioni possibili, che vanno poste rigorosamente alla fine della stringa di controllo sono:

c: permette di stampare in compresso anche se il numero di caratteri per riga è inferiore a 80

p: manda in stampa solo le pagine pari
d: manda in stampa solo quelle dispari.

Quindi

```
pg151 5510c pippo
```

stampa in compresso il file «pippo» mentre:

```
pg151 +6610cd
```

provvede a stampare (sempre in compresso...) le sole pagine dispari del file evitando di numerare le pagine e di porre form feed come terminatori di pagina.

Note

La versione del programma che vede è adatta per MS-DOS, se però desiderate utilizzarla sotto un sistema operativo diverso, è sufficiente cancellare la prima linea eliminando la:

```
#define MSDS 1
```

in questo caso non verranno più inviati caratteri di controllo per la stampa su 132 colonne e ovviamente il file non sarà più inviato sull'unità «prn», ma verranno creati 2 file chiamati «pari» e «dispari» che conterranno le pagine corrispondenti.

Naturalmente, l'opzione «c» viene riconosciuta anche se il programma gira sotto Unix però agisce in modo diverso, perciò è necessario specificare un margine sinistro corretto.

Per un buon utilizzo del programma è necessario lasciare un foglio bianco all'inizio del modulo in modo da posizionare correttamente la carta per la stampa delle pagine pari, ed è consigliabile evitare l'uso di spooler di stampa sotto MS-DOS, dato che in ogni caso si dovrà aspettare la fine del programma dovendo girare il blocco di fogli per procedere alla stampa della facciata posteriore.

Purtroppo, il panorama dei word processor è vasto e pochi sono compatibili tra loro, perciò adattare «PG151» in modo che li riconosca tutti sarebbe un lavoro «biblico» con scarse possibilità di riuscita.

Se il testo è ASCII, non ci sono problemi, io utilizzo da alcuni mesi il programma su un sistema Unix con ottimo successo, dato che posso elaborare ulteriormente il risultato con le varie utility di stampa presenti sotto questo sistema operativo («pr», «nroff», «cat»...) ottenendo stampe complete e di buon livello qualitativo.

Scritte scorrevoli anche in Basic

di Andrea Patelli - Livorno

Ho molto apprezzato la routine in Assembler 8086 che inserita nel programma in Turbo Pascal di Giovanni Benintende permetteva di far scorrere pixel per pixel una scritta a piacere.

Perché non fare partecipi di questa piacevole possibilità anche i cultori del Basic? Certamente le difficoltà sono un poco superiori, ma non insormontabili.

Il programma Basic inizia con la definizione del segmento in cui caricare da disco tramite BLOAD o dalle istruzioni DATA tramite POKE la routine in codice macchina. Il segmento viene scelto dopo la fine dell'area allocata dal GWBASIC o BASICA memorizzata nelle locazioni 510 e 511 (hex). Per il resto il programma Basic si commenta da solo.

Come si vede qualche variazione rispetto al programma in Pascal è stata necessaria. Infatti ho dovuto inserire il prelievo dallo STACK dei 4 parametri passati dalla CALL ed anche un Check della tastiera per verificare l'eventuale battuta del Ctrl-C.

Infatti in caso di parametri non ortodossi senza questa possibilità talvolta non rimane altro che spegnere il PC dopo aver contemplato sul video le cose più strane.



Scritte scorrevoli

```
10 * ----- scritta scorrevole -----
20 *
30 DEF SEG=0:SEGBASIC=PEEK(&H510)+PEEK(&H511)*256:SEGLM=SEGBASIC+4200
40 REM def seg=seglm:bload "scorri.bld",0
50 DEF SEG=SEGLM:FOR I=0 TO 99:READ A:POKE I,A:NEXT
60 PRINT:INPUT "riga,col. iniziale,larghezza ":RIGAX,COLX,LARGX
70 PRINT:INPUT "num. pixel,ritardo ":NPIXX,RITAX
80 PRINT:INPUT "messaggio ":A#
90 PRINT:INPUT "screen (1/2)":SCR:IF SCR=0 THEN SCR=2
100 INIZIOX=COLX-1+(RIGAX-1)*320-COLX*(SCR=1)
110 FINEX=INIZIOX+LARGX-LARGX*(SCR=1)
120 L=LEN(A#):SCREEN SCR
130 FOR I=1 TO L
140 LOCATE RIGAX,COLX+LARGX-1:PRINT MID$(A#,I,1)
150 CALL SCORRI(INIZIOX,FINEX,NPIXX,RITAX)
160 NEXT
170 DATA 85,139,236,139,118,12,139,60,139,118
180 DATA 8,139,28,139,118,6,139,4,139,118
190 DATA 10,139,52,30,80,184,0,184,142,216
200 DATA 88,64,83,139,222,186,0,0,83,181
210 DATA 4,83,177,4,209,226,208,87,255,131
220 DATA 210,0,129,195,0,32,254,201,117,240
230 DATA 91,131,195,80,254,205,117,229,91,75
240 DATA 59,223,117,220,80,180,11,205,33,88
250 DATA 139,216,83,139,216,75,117,253,91,75
260 DATA 117,246,91,75,117,192,31,93,202,8
```

RIVOLGITI AGLI ESPERTI PER SCOPRIRE LE AMIGA PROFESSIONALS PERFORMANCES



DESKTOP PUBLISHING

HARDWARE DTP

AMIGA 2000 2 DRIVE 3"1/2
MONITOR 2080 LUNGA PERSISTENZA
SCHEDA INTERNA 2MB
HARD DISK 20MB SCSI SU SCHEDA
STAZIONE DIGITALIZZATRICE
STAMPANTE LASER CANON LBP8 II
STAMPANTE LASER NEC LC890 POSTSCRIPT
STAMPANTE INK JET COLORE XEROX 4020

SOFTWARE DTP

PAGE SETTER ITALIANO
PROFESSIONAL PAGE
ELABORAZIONI POSTSCRIPT
*...E CON SHAKESPEARE
+ XEROX 4020
DESKTOP PUBLISHING
A COLORI !!!!*

WORKSTATION GRAFICA PROFESSIONALE

AMIGA 2000 2 DRIVE 3"1/2
MONITOR 2080 LUNGA PERSISTENZA
SCHEDA INTERNA 2MB
HARD DISK 20MB SCSI SU SCHEDA
STAZIONE DIGITALIZZATRICE
GENLOCK PROFESSIONALE
XEROX 4020 INK JET COLORE
POLAROID PALETTE AMIGA
TAV.GRAFICA KURTA IS 12 * 17 1000 ppi

SOFTWARE
DE LUXE PAINT
DIGIPAIN
AEGIS VIDEOSCAPE 3D
SCULPT 3D 68020
SCULPT 3D ANIMATE 68020
SCULPT 3D ANIMATE PROFESSIONAL
AEGIS DRAW PLUS
INTROCAD X-CAD CAO 3D

PIX COMPUTER S.R.L.
VIA F. D'OVIDIO, 6C
TEL. 06/8293507-825731
00137 ROMA
COMPUTER & CO.
P. IVA 08309630583

AMIGA 500	950000
AMIGA 500 + MONITOR 1084	1590000
AMIGA 2000 SENZA MONITOR.....	1950000
AMIGA 2000 2 DRIVE 3"1/2	2190000
ESPANSIONE 512K INTERNA A500	180000
ESPANSIONE 1MB INTERNA A1000	490000
ESPANSIONE 2MB EST.A500/A1000	850000
ESPANSIONE 2MB INTERNA A2000	650000
DISK DRIVE 3"1/2 EST.A500/A1000	330000
DISK DRIVE 3"1/2 INTERNO A2000	250000
HARD DISK 20MB EST.A500/A1000	1150000
HARD CARD 20MB SCSI A2000	1150000
HARD CARD 20MB MS-DOS A2000	850000
HARD CARD 40MB MS-DOS A2000	1150000
SCHEDA JANUS XT A2000	990000
KIT SOSTITUZIONE MOTOROLA 68010	99000
SCHEDA 68020 + 68881 16MHZ	1650000
AMIGA-EYE A500/A1000/A2000	150000
VD AMIGA FRAMEGRABBER	850000
STAZIONE DIGITALIZZATRICE (DIGI-VIEW + TELECAMERA PANASONIC WV1410 + STATIVO PROFESSIONALE)	1290000
AMIGA SOUND A500/A1000/A2000	150000
INTERFACCIA MIDI A500/A1000/A2000	99000
GENLOCK 8702 CVBS A500/A2000	750000
GENLOCK PROFESSIONALE	850000

TAVOLETTE GRAFICHE KURTA:	
PENMOUSE 6"*9" 200 PPI	250000
SERIE IS 8,5"*11" 1000 PPI	790000
SERIE IS 12"*12" 1000 PPI	990000
SERIE IS 12"*17" 1000 PPI	1690000
PENNA A DUE BOTTONI	290000
CURSORE A 4 BOTTONI	290000
CAVO E SOFTWARE PER AMIGA	110000
STAMPANTI:	
PANASONIC KX-P1081 80COL 120CPS	590000
NEC P2200 80COL 216CPS 24 AGHI.....	990000
NEC P6 80COL 216CPS 24 AGHI.....	1350000
NEC P6 KIT COLORE	1490000
NEC P7 136COL 216CPS 24 AGHI	1650000
NEC P7 136COL 216CPS 24 AGHI	1790000
XEROX 4020 INK JET COLORE	3450000
CANON LBP8 II LASER 8PPM	TELEFONARE
NEC LC890 LASER POSTSCRIPT	TELEFONARE
HARD COPIER SHINKO	TELEFONARE
POLAROID PALETTE PER AMIGA	3450000
SOFTWARE ORIGINALE:	
SHAKESPEARE,CITY DESK,PAGE SETTER,PAGE PROFESSIONAL,FORMS IN FLIGHT,INTROCAD, X-CAD,CAO 3D,DPAIN II,DIGIPAIN,AEGIS VIDEOSCAPE 3D,SCULPT 3D,SCULPT ANIMATE SERIE COMPLETA AEGIS E MOLTI ALTRI.	

LA PIX COMPUTER S.R.L. E' UNA SOCIETA' CHE SI OCCUPA DI CONSULENZA E VENDITA DI WORKSTATION GRAFICHE BASATE SU COMMODORE AMIGA 2000, ED ORGANIZZA CORSI DI COMPUTER GRAFICA IN QUALITA' DI COMMODORE COMPUTER CENTER. A RICHIESTA SI INVIANO LISTINI (ANCHE PER RIVENDITORI), SI INOLTANO OFFERTE E PREVENTIVI SU WORKSTATION GRAFICHE COMPLETE "CHIAVI IN MANO". E' POSSIBILE RICHIEDERE SCHEDE TECNICHE DEI PRODOTTI D'INTERESSE. CONSULTATECI AI TEL.06/8293507-825731. SPEDIZIONI IN CONTRASSEGNO IN TUTTA ITALIA-I PREZZI COMPREDONO L'IVA 18%.