

A causa del listato un tantino lungo questa volta dobbiamo accontentarci di un solo programma. In effetti il Pascal ha il difetto/pregio di essere un po' prolisso, non mi dispiacerebbe una via di mezzo tra il Pascal e il C, ad esempio invidio al C l'uso delle parentesi graffe al posto dei BEGING.. END.

Tornando al programma presentato, questo serve per listare sul video una directory (eventualmente ridotta tramite l'uso dei caratteri jolly) e di poter effettuare la scelta di un file direttamente sul video (tramite le frecce). Il tutto grazie alla possibilità del Turbo Pascal di effettuare facilmente le chiamate all'MS-DOS. Collegando questo programma con lo SPAWN presentato su MC n. 68 è facilissimo realizzare un menu di lancio automatico dei programmi presenti su di un disco evitando così ad un eventuale utente inesperto l'uso dei comandi del DOS

v.d.d.

Turbo Directory

di Luca Padovano e Stefano Marchi Imola

Se siete assidui utilizzatori del Turbo Pascal e non avete mai sognato di avere una routine che eseguisse la directory del disco, questo programma non vi interesserà troppo.

Ma per tutti gli altri, «Turbo Directory» rappresenterà sicuramente un ulteriore buon motivo per aver comprato questo numero di MC.

In poche centinaia di istruzioni, viene

È disponibile, presso la redazione, il disco con il programma pubblicato in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 208.

Listato 1

```
PROGRAM directory;
CONST dta=$2f;
      carc=$4e;
      next=$4f;
      secfree=$36;
      dsk=$19;
      eot=#00;
TYPE x=RECORD
      CASE INTEGER OF
        1:(ax,bx,cx,dx,bp,si,di,ds,es,flags:INTEGER);
        2:(al,ah,bl,bh,cl,ch,dl,dh:BYTE)
      END;
      line=STRING[80];
      disco=STRING[2];
VAR path:STRING[40];
      mask:STRING[13];
      file_name:STRING[12];
      prog:ARRAY[1..113] of STRING[12];
      drive:DISCO;
      error:BYTE;
      regs:X;
      lx,ly,punt,punt1,n,j,k,max,resto,alfa,beta,
      alfa1,beta1,c,num,spiazzamento,segmento,dir,posit:INTEGER;
      ret,exit:BOOLEAN;
      ch:CHAR;
(-----)
PROCEDURE inv (flag:INTEGER);
BEGIN
  IF flag=1 THEN BEGIN textcolor(black);textbackground(white) END
  ELSE BEGIN textcolor(white);textbackground(black) END
END;
PROCEDURE WRITEI(x,y:INTEGER;dato:LINE);
BEGIN INV(1);GOTOXY(x,y);WRITE(dato);INV(0) END;
PROCEDURE WRITEN(x,y:INTEGER;dato:LINE);
BEGIN GOTOXY(x,y);WRITE(dato) END;
(-----)
PROCEDURE store_name;
BEGIN
  num:=MEM[segmento:spiazzamento+21];
  dir:=0;
  IF num=16 THEN BEGIN dir:=1;file_name:='' END
  ELSE file_name:=' '
  FOR k:=1 TO 12 DO
    BEGIN
      num:=MEM[segmento:spiazzamento+29+k];
      IF num=0 THEN
        BEGIN
          file_name:=file_name+' '
          k:=12
        END
      ELSE file_name[k+dir]:=chr(num)
    END;
  IF COPY(file_name,1,3) <> '\. ' THEN
    BEGIN prog[punt]:=file_name;punt:=punt+1 END
END;
(-----)
FUNCTION current:DISCO;
VAR regs:X;
      drv:DISCO;
BEGIN
  WITH regs DO
    BEGIN
      ah:=dsk;
      MSDOS(regs);
      drv:=CHR(65+al)+'';
    END;
    current:=drv
END;
(-----)
PROCEDURE get_mask;
VAR trovato:BOOLEAN;
BEGIN
  WRITEN(24,5,' >MASK< ');
  WRITEN(24,6,' ');
  WRITEN(24,7,' *');
  WRITEN(24,8,' ');
  WRITEN(24,9,' ');
  GOTOXY(26,7);mask:='*.*';drive:=CURRENT;
  READLN(path);
  IF path='' THEN path:=''
  ELSE

```

```

BEGIN
  IF path[2]=';' THEN                                (il drive è specificato ?)
  BEGIN
    drive[1]:=UPCASE(path[1]);
    DELETE(path,1,2)                                (se si, aggiorna drive e path)
  END;
  IF path[1]<>'\' THEN path:='\'*path;              (mette '\' se manca)
  posit:=POS('\' ,path);
  IF posit>0 THEN                                   (acquisisce la maschera)
  BEGIN
    trovato:=FALSE;n:=posit;
    REPEAT
      n:=n-1;
      trovato:=path[n]='\' ;
    UNTIL (trovato) OR (n=1);
    mask:=COPY(path,n-1,posit+3);
    IF n=1 THEN n:=2;
    path:=COPY(path,1,n-1)
  END;
  IF path[LENGTH(path)]<'\' THEN path:=path+'\'
  END;
CLRSCR
END;
(-----)
PROCEDURE spacefree;
VAR regs:X;
    a,b,c,d,free,perc:REAL;
    lettore:INTEGER;
BEGIN
  lettore:=ord(drive[1])-64;                          (drive di default)
  WITH regs DO
  BEGIN
    ah:=secfree;                                       (controlla spazio disponibile)
    dl:=lettore;
    MSDOS(regs);
    a:=ax;                                             (settori liberi)
    b:=bx;                                             (cluster liberi)
    c:=cx;                                             (byte per settore)
    d:=dx;                                             (numero totale di cluster)
    free:=(a*b*c)/1024;                               (spazio libero in byte)
    perc:=b*100/d;                                     ( " " " in percentuale)
    WRITELN,GOTOXY(19,17);INV(1);
    WRITE(' ',free:3:0,' Kbyte liberi. Spazio disponibile: ',perc:3:0,'% ');
    INV(0)
  END
END;
(-----)
PROCEDURE get_dir(funz_num:BYTE);
VAR regs:X;
    nfc:STRING[22];
BEGIN
  nfc:=drive+path+mask+eot;                          (path (deve terminare con chr(0)))
  WITH regs DO
  BEGIN
    ah:=funz_num;
    cl:=16;                                           (ricerca subdirectory e file)
    ds:=SEG(nfc);                                     (mette in ds la POSizione di nfc)
    dx:=OFS(nfc)+1;
    MSDOS(regs);
    error:=ax                                         (0 se tutto bene; 16 fine; 3 non trovato)
  END
END;
(-----)
PROCEDURE WRITE_IT;
BEGIN
  WINDOW(2,5,79,16);CLRSCR;WINDOW(1,1,80,25);
  IF k=max THEN punti:=50 ELSE punti:=punt-50*max;
  lx:=punti DIV 10;ly:=punti MOD 10;                (calcola numero di colonne)
  FOR n:=1 TO lx DO
    FOR j:=1 TO 10 DO
      WRITEN(5+(n-1)*15,j+5,prog[(n-1)*10+j+50*(k-1)]);
    FOR j:=1 TO ly(-1) do
      WRITEN(5+lx*15,j+5,prog[j+lx*10+50*(k-1)]);
  SPACEFREE
END;
(-----)
PROCEDURE cornice;
BEGIN
  WRITEN(1,4,'r');
  FOR n:=2 TO 79 DO WRITE('-');
  WRITE(' ');
  WRITEI(35,4,'>DIRECTORY<');
  FOR n:=5 TO 16 DO
  BEGIN
    WRITEN(1,n,'|');WRITEN(80,n,'|');
  END;
  WRITEN(1,17,'L');
  FOR n:=2 TO 79 do WRITE('-');
  WRITE(' ');
END;
(-----)
PROCEDURE select;
BEGIN
  alfa:=1;beta:=1;
  REPEAT
    WRITEI(5+(alfa-1)*15,beta+5,prog[(alfa-1)*10+beta+(k-1)*50]);
    READ(kbd,ch);
    IF ch=#13 THEN ret:=TRUE                          (return)
    ELSE
      IF (ch=#27) AND (keypressed) THEN
      BEGIN
        ret:=FALSE;
        alfa:=alfa;beta:=beta;
        READ(kbd,ch);
      END
    END
  UNTIL ret;
END;

```

(segue a pag. 220)

fornito al gentil lettore una utilissima routine che non solo esegue il comando Dir ma permette anche la selezione del file o della directory col metodo freccia giù freccia-su-enter-selezione.

Cosa volete di più?

La Routine

Il programma è stato scritto in Turbo Pascal 3.0 su di un Olivetti M24 e provato su due XT compatibili (con e senza hard-disk) sotto MS-DOS 3.10 e 3.20.

TD stampa i file a gruppi di 50 in una finestra. Come è noto infatti, l'MS-DOS non può conservare più di 112 nomi di file nella directory radice; a loro volta, le subdirectory possono contenere altri 112 nomi.

Sono quindi necessarie tre passate da 50 file per avere la directory completa (veramente 112 file su un solo disco non li abbiamo mai visti...).

Naturalmente chi volesse stampare meno (o eventualmente più) file alla volta troverà Turbo Directory molto facile da customizzare (si dice così, vero?).

Posizionandosi su un file (con le frecce) e premendo return si esce dal programma e il nome del file viene stampato. Se invece si seleziona una subdirectory (sono individuate dal segno '/' prima del nome), ne viene mostrato il contenuto.

Naturalmente è possibile scegliere il percorso e la «maschera». In pratica volendo visionare tutti i file Pascal nella subdirectory "prog" di un dischetto nel drive B, basterà rispondere B:/prog/*.* alla richiesta del computer.

Il default è: <drive corrente>:/*.*

Procedure del programma

INV: scrive in inverso se 1, normale 0.
WRITEI: posiziona il cursore e scrive in inverso

WRITEN: posiziona il cursore e scrive normale

STORE_NAME: prende dalla DTA i dati e li conserva in prog [].

CURRENT: acquisisce drive corrente.

GET_MASK: richiede la path e la maschera.

SPACEFREE: spazio libero su disco.

GET_DIR: trova i dati della directory.

WRITE_IT: scrive i dati su schermo.

CORNICE: anche l'occhio vuole la sua parte.

SELECT: su, giù, destra e sinistra con le frecce.

FIND_IT: gestisce la memorizzazione della directory

DIR_or_FILE: controlla se è file o directory.

Note tecniche

Come è facile immaginare, Turbo Directory fa largo uso dei servizi del DOS attivabili tramite l'interrupt 21H. In particolare si sono usati i seguenti servizi:

- \$19 (25 dec):** acquisisce il drive corrente,
- \$4e (78 dec):** inizia la ricerca di un file,
- \$4f (79 dec):** trova le parti successive di un file,
- \$2f (47 dec):** locazione della DTA (Disk Transfer Area),
- \$36 (54 dec):** spazio libero sul disco.

Concettualmente il metodo usato è abbastanza semplice e immediato. Prima di tutto si acquisisce il drive di default grazie al servizio \$19 che restituisce in AL il lettore corrente (0=A, 1=B).

Fatto questo, dobbiamo trovare l'indirizzo in memoria della DTA (Disk Transfer Area). Ogni volta che il DOS deve restituire all'utente dei dati relativi ad un file, esso alloca un certo numero di byte proprio nella DTA.

In particolare nel nostro caso, la DTA contiene 43 byte così suddivisi:

- 0-20:** area usata dal DOS per problemi suoi,
- 21:** attributo del file trovato (16=directory),
- 22-23:** data di creazione,
- 24-25:** ora di creazione,
- 26-29:** lunghezza del file (in Byte),
- 30-42:** nome e suffisso del file in formato ASCII Z.

Chi volesse modificare il formato di stampa della directory (aggiungendo per esempio data e ora di creazione), non avrà altro da fare che modificare la procedura Store_Name in modo che oltre a leggere il nome legga anche queste informazioni aggiuntive.

Ma come facciamo a dire al DOS «prendi i dati relativi alla directory e mettili nella DTA»? Semplice, la funzione \$4e (FIND_FIRST) cerca il primo file che corrisponde ad un nome specificato (vedi procedura GET_DIR); la funzione \$4f invece continua la ricerca di un file iniziata dalla funzione \$4e o dalla stessa \$4f.

Quindi, specificando il percorso e la maschera, ci costruiamo la directory «pezzo per pezzo» andando a cercare i file per tutto il disco. Per trovare lo spazio libero su disco è ancora più semplice. La funzione \$36 ci fornisce tutti i dati necessari su settori e cluster (vedi procedura Spacefree). Tutto qui. Qualche riga per migliorare l'estetica e il gioco è fatto. Per maggiori informazioni, rimandiamo ai commenti del listato 1 e all'impareggiabile manuale di Peter Norton, (PC IBM: guida del programmatore) al quale tributiamo doveroso tributo di omaggio e riconoscenza (amen!). **MC**

(continua da pag. 219)

```

IF ch=#75 THEN alfa:=alfa-1;IF alfa=0 THEN alfa:=1;      (freccia sinistra)
IF ch=#77 THEN
BEGIN
  alfa:=alfa+1;
  IF (alfa=1x+2) OR ((alfa=1x+1) and (beta>ly)) THEN alfa:=alfa-1
END;
IF ch=#80 THEN
BEGIN
  beta:=beta+1;
  IF (beta=1) OR ((alfa=1x+1) and (beta=ly+1)) THEN IF k:=max THEN
  BEGIN k:=k+1;ch:=#13 END ELSE beta:=beta-1
END;
IF ch=#72 THEN
BEGIN
  beta:=beta-1;
  IF beta=0 THEN IF k=1 THEN beta:=1 ELSE BEGIN k:=k-1;ch:=#13 END
END;
WRITEln(b+(alfa-1)*15,beta+5,prog((alfa-1)*10+beta+(k-1)*50));
END
ELSE
BEGIN
  exit:=TRUE;
  ret:=TRUE;
  ch:=#13
END
UNTIL ch=#13;
END;

{-----}

PROCEDURE find_it;

BEGIN
GET_DIR(cerca);
IF error=0 THEN
BEGIN
  STORE_NAME;
  GET_DIR(next);
  WHILE error=0 DO
  BEGIN
    STORE_NAME;
    GET_DIR(next);
    END;
    punt:=punt-1;error:=0
  END;
  posit:=POS('\'.COPY(path,4,LENGTH(path)));
  IF (posit>0) AND (mask<'*.') THEN
  BEGIN
    prog[punt+1]:='\...
    punt:=punt+1
  END
END;

{-----}

PROCEDURE DIR_or_FILE;

BEGIN
c:=(alfa-1)*10+beta+(k-1)*50;
IF prog[c][1]='\' THEN
BEGIN
  punt:=1;
  IF prog[c][2]='.' THEN
  BEGIN
    path:='\';
    mask:='*.';
    END
  ELSE path:=path+COPY(prog[c],2,POS(' ',prog[c])-1)+'\'
  END
  ELSE exit:=TRUE
END;

{-----}

BEGIN
WITH regs DO
BEGIN
  ah:=dta;
  MSDOS(regs);
  segmento:=es;

  spiazzamento:=bx
  END;

  punt:=1;
  CLRSCR;
  GET_MASK;
  exit:=FALSE;
  CORNICE;

  REPEAT
  ret:=FALSE;FIND_IT;
  IF error=0 THEN
  BEGIN
    max:=punt DIV 50;resto:=punt MOD 50;
    k:=1;
    IF punt:=50 THEN
    BEGIN
      max:=0;WRITE_IT;SELECT
    END
    ELSE
    WHILE (k:=max) AND (NOT ret) DO
    BEGIN
      WRITE_IT;
      SELECT;
      IF (k:=max+1) AND (resto>0) AND (NOT ret) THEN
      BEGIN WRITE_IT;SELECT END
    END;
  END
  DIR_or_FILE
  END ELSE exit:=TRUE
UNTIL exit;
CLRSCR;

IF error=0 THEN WRITEln(drive+path+prog[c]) ELSE WRITEln('No files.')
END.

{-----}

```

Soluzioni Hardware & Software per i professionisti della comunicazione

Presidente di una nota associazione: "*Siamo purtroppo in ritardo con il nostro giornale, non possiamo aspettare i soliti 10 giorni*".

Tipografo: "*Mi faccia avere oggi il dischetto con i testi e le immagini e può mandare a ritirare la sua pubblicazione dopodomani*".

Fantascienza?

No, desktop publishing!

Ovvero editoria elettronica.

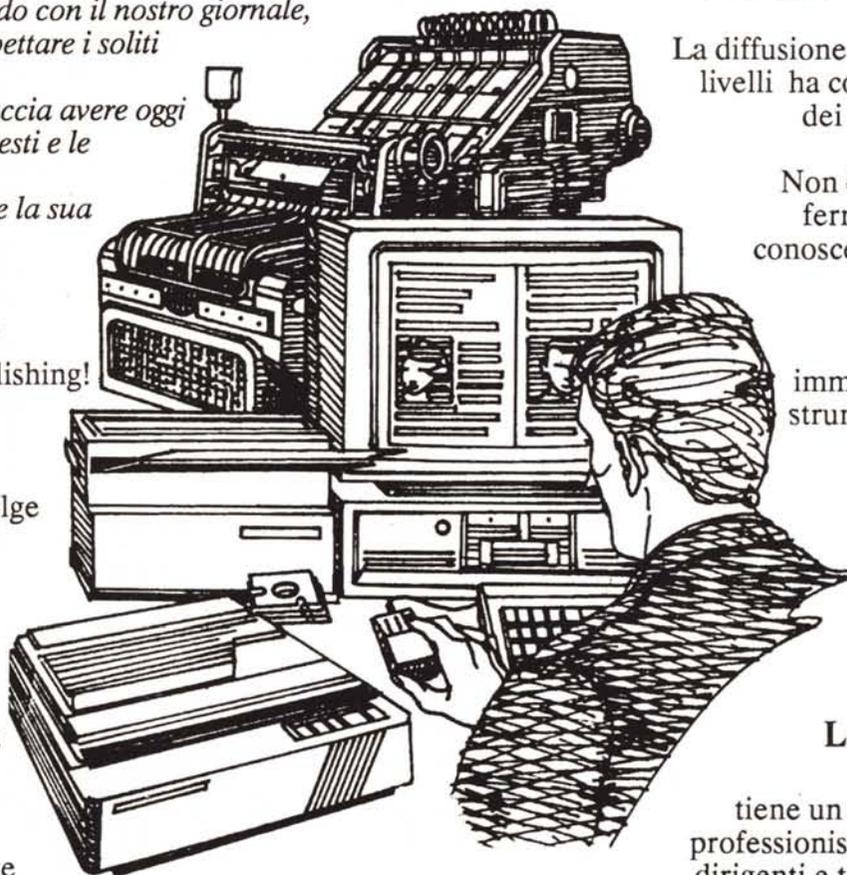
Questo nostro messaggio si rivolge ai tipografi, alle aziende, ai professionisti.

E' un messaggio che significa di fare attenzione a non lasciarsi sorpassare, significa che per continuare a stare sulla cresta dell'onda è necessario stare al passo con le tecnologie.

La carta che esce dalla vostra azienda vi qualifica, dice di voi, della vostra professionalità.

Un sistema completo, una tipografia sul vostro tavolo

Vi proponiamo uno strumento **completo e potentissimo** che richiede solo iniziativa, intelligenza, voglia di essere avanti.



Oltretutto un sistema che offre così tanto costa anche poco. La diffusione dell'informatica a tutti i livelli ha consentito l'abbattimento dei costi del computer e dei programmi. Non è quindi il costo che può fermarvi ma la mancanza di conoscenza delle possibilità del mezzo.

E' vero, non è facile né immediato padroneggiare lo strumento che come tutti gli strumenti richiede uno studio ben programmato. Ed è proprio a questo aspetto che noi della Computerline abbiamo posto una particolare attenzione.

La divisione didattica della Computerline

tiene un corso di base per neofiti professionisti, funzionari di azienda, dirigenti e tecnici, con il sistema del *master learning*. Si tratta di un corso di 20 ore ad unità programmate con test di padroneggiamento di ogni unità e test finale.

Non si accede all'unità successiva se non si è superata la precedente. Non si termina il corso se non dopo comprovata maestria.

A questo corso di base si affiancano i **corsi specialistici**, tra cui quello di desktop publishing che vi insegnerà a impaginare testi e figure, velocemente, con maestria, contribuendo a darvi professionalità e prestigio.

COMPUTERLINE

00188 Roma - Via Rubra 190/192 - (Via Flaminia - GRA)
Tel. (06) 300.56.46 - 300.58.57
Telefax (06) 69.12.285 - 79.70.966
Telex 62.1166 Compul-I



00173 Roma - Via U. Comandini 49 - (Seconda Università - GRA)
Tel. (06) 61.33.025 - 61.30.912
Telefax (06) 69.12.285 - 79.70.966
Telex 62.1166 Compul-I