Utilizzo di una stampante

di Sergio Polini

Nei mesi scorsi, abbiamo esaminato due unit mediante le quali un'applicazione Turbo Vision può accedere sia alle caratteristiche «permanenti» di una stampante (il nome, le sequenze di controllo, ecc.), sia a quelle che abbiamo definito «temporanee» (i tipi di carattere disponibili, il formato della carta, la porta di output). Abbiamo anche visto un programma PRINST che, usando quelle unit, permette all'utente di installare una stampante sul proprio sistema, creando un file di definizione con estensione PRN. Ci rimane ora da mostrare come un programma Turbo Vision possa utilizzare tali file per operazioni di stampa. Avremo bisogno di una nuova unit ma, prima di questa, vedremo un breve demo che illustrerà concretamente cosa ci proponiamo di ottenere; in questo modo, saranno poi più chiari i motivi delle scelte operate nella realizzazione della terza e ultima unit, TVPRINT

Ricordo che il nostro cammino era iniziato con un esame dei diversi meccanismi mediante i quali Windows ci permette di stampare prescindendo in buona misura dalle particolarità di ogni stampante; il nostro scopo era quello di pervenire ad una soluzione del problema dell'indipendenza dall'hardware anche sotto MS-DOS, pure se limitatamente alla stampa di soli testi.

Abbiamo per prima cosa realizzato un programma di installazione che consente di iniziare a usare rapidamente qualsiasi stampante, alla sola condizione di saperne leggere il manuale. Dobbiamo ora vedere come un'applicazione Turbo Vision possa avere rapido accesso alle informazioni circa le stampanti installate e l'eventuale stampante di default, cambiare in qualsiasi momento le caratteristiche «temporanee» di una stampante, scegliere tra i font disponibili, tenere sotto controllo la posizione della testina di stampa, interrompere in ogni momento una stampa in corso, mantenere pieno controllo in caso di errore critico.

Nel conseguimento di tali obiettivi cercheremo, ovviamente, di riprodurre meccanismi abituali sotto Windows; avremo quindi un menu *File* con le opzioni *Stampa* e *Imposta stampante*, e un'opzione *Tipo carattere* in un altro menu (*Modifica* nel nostro demo).

Selezione della stampante corrente

La figura 1 contiene il listato di DEMO.PAS, il nostro breve programma di esempio. Notiamo subito che il metodo *InitMenuBar* della classe *TDemo* predispone due menu *File* e *Modifica* secondo quanto abbiamo appena detto; vi sono poi dei metodi *Print, PrinterSetup* e *SelectFont* che corrispondono alle opzioni di tali menu. Notiamo anche che la classe *TDemo* ha una variabile d'istanza *Printer*, appartenente ad una classe *TPrinter*, definita nella unit TVPRINT. Il constructor di *Printer* viene chiamato nell'ambito del constructor di *TDemo*; non è necessario che sia così, ma è importante che, se si differisce l'inizializzazione della variabile *Printer*, i comandi *cmPrint, cmPrSetup* e *cmSelFont* siano disabilitati (con *DisableCommands*), per essere poi abilitati (con *EnableCommands*) solo dopo quella inizializzazione.

Il constructor di *Printer* tenta di associare tale variabile all'eventuale stampante di default; se tale associazione automatica non ha successo, occorrerà rendere nota all'utente la necessità di procedere ad una associazione manuale, mediante l'opzione *Imposta stampante*.

Abbiamo visto che il programma PRINST colloca i file di definizione delle stampanti in una apposita directory; questa sarà C: TVPRNS per default, ma potrà essere anche un'altra, a condizione che il suo nome venga assegnato ad una variabile TVPRNDIR dell'environment. Si può in modo analogo determinare la stampante di default, assegnando il nome del suo file, senza l'estensione PRN, ad una variabile TVPRN dell'environment, ad esempio con SET TVPRN=IBM4208 per una IBM Proprinter XL24, o con SET TVPRN=HPDJ500 per una HP DeskJet 500. Come vedremo, il constructor della classe *TPrinter* cerca il valore di tale variabile con *GetEnv('TVPRN')* e tenta di leggere il file corrispondente mediante la funzione *ReadPrinterInfo* (definita nella unit PRSETUP), passandole come parametro una variabile d'istanza *PD*, di tipo *TPrinterDef*. Se il tentativo non ha successo, il campo *Name* di *PD* conterrà una stringa nulla.

La classe *TDemo* comprende una funzione *PrinterSelected*, il cui compito è di ritornare TRUE o FALSE secondo che la variabile *Printer* risulti associata o meno ad una stampante; in caso negativo, viene anche prodotto un messaggio che invita l'utente a procedere con l'opzione *Imposta stampante* del menu *File* (la funzione è un metodo di *TDemo*

invece che di *TPrinter* proprio per consentire al programmatore di fare riferimento nel messaggio al sistema di menu dell'applicazione, senza bisogno di modificare la unit TVPRINT).

Modifica dell'impostazione

Il metodo *TDemo.PrinterSetup*, corrispondente all'opzione *Imposta stampante*, chiama il metodo *DoSetup* di *Printer*. Questo produrrà un messaggio d'errore se non troverà la directory contenente i file di definizione delle stampanti, altrimenti aprirà una dialog box come quella mostrata nella figura 2. La list box propone i nomi delle stampanti installate, seguiti dal nome dei relativi file di definizione e dalla indicazione della porta di output per ognuna. Per selezionare una stampante basta posizionarsi sul nome di quella desiderata, con la tastiera o con il mouse, e premere Invio o il pulsante *Ok*.

La dialog box comprende anche un pulsante *Imposta*, mediante il quale si accede ad una dialog box di tipo *TPrSetupDialog*, già vista nella unit PRSETUP e usata anche dal programma PRINST. Mediante tale dialog box è possibile modificare le caratteristiche «temporanee» della stampante correntemente selezionata, anche se questa non è

Prima o poi doveva succedere...

Evidentemente l'estate scorsa ha avuto sul sottoscritto effetti ... troppo benefici. Sole, mare, riuscire ad insegnare alla propria figlia a nuotare (anche sott'acqua!), conoscere la bellezza di altri paesi e il fascino di altri popoli. Ma anche un calo di concentrazione.

Eccomi quindi a dover precisare che la didascalia della figura 1 nel numero scorso è sbagliata: vi viene riprodotta la seconda parte della unit PRDEF, non PRSETUP. E fin qui poco male; ve ne sarete già accorti.

Per la prima volta in cinque anni, devo però anche correggere un bug. Si tratta di un errore che si era insinuato nel sorgente della funzione *CtrlToStr*, contenuta nella unit PRSETUP. Riproduco il suo ciclo **while**:

```
d := 100;
while d > 0 do begin
  n := m div d;
  if (n > 0) or (d = 1) then begin
    Inc(j);
    S30[j] := Chr(Ord('0')+n);
    if j = 30 then goto Out;
    m := m mod d;
  end;
  d := d div 10;
end;
```

La variabile *d* è un divisore che, valendo prima 100, poi 10, infine 1, consente di aggiungere alla stringa *S30*, nell'ordine, i caratteri per le centinaia, le decine e le unità, corrispondenti volta per volta al quoziente *n*. Volevo evitare zeri iniziali; non volevo, in parole povere, che il valore 3 venisse tradotto in '003'. Ho introdotto, a questo scopo, un **if** che esclude l'aggiunta di zeri a *S30* se *n* vale zero, tranne il caso in cui *d* sia ormai ridotto a 1, in cui cioè si sia giunti alle unità (il valore zero deve essere tradotto almeno in '0').

Non va. Ne risulta, infatti, che valori compresi tra 100 e 109 vengono tradotti in stringhe di soli due caratteri ('10', '11', e così via), in quanto, quando *d* vale 10 e *n* assume un valore nullo per le decine, lo zero nella seconda posizione non viene aggiunto alla stringa.

Per ovviare al problema, ho aggiunto una variabile locale *Divisor* di tipo *Integer* e modificato come segue il codice:

```
if m < 10 then Divisor := 10
else Divisor := 100;
d := Divisor;
while d > 0 do begin
  n := m div d;
  if (n > 0) or (d < Divisor) then begin
  ...
```

Non è tutto. Usando le unit, ho dovuto constatare che a volte si ha a che fare con stampanti «strane», magari prive di documentazione; altre volte può risultare poco opportuno utilizzare le sequenze di controllo della stampante per il grassetto o il sottolineato. La unit PRDEF, però, richiede obbligatoriamente l'immissione delle relative stringhe, passando TRUE come secondo parametro del constructor di *TInputCtrlString*. Ho preferito sostituire tutti i TRUE con FALSE in *TPrDefDialog.Init*, e vi esorto a fare altrettanto.

quella che si intende utilizzare (verrà utilizzata, infatti, la stampante che risulterà selezionata quando si chiuderà con Invio o col pulsante *Ok* la dialog box «madre»).

Quando abbiamo esaminato l'uso delle stampanti sotto Windows, abbiamo rilevato che, a partire dalla versione 3.0 di questo, è possibile apportare modifiche che abbiano effetto solo nell'ambito di un'applicazione, senza alterare l'impostazione di sistema. Vedremo che la unit TVPRINT è invece realizzata in modo da registrare sempre sul file di definizione le variazioni apportate alla impostazione di una stampante. Rimane possibile, comunque, realizzare applicazioni che adottino impostazioni variabili senza modificare i file di definizione; la variabile d'istanza *PD* della classe *TPrinter*, infatti, è pubblica, come anche la classe *TPrSetupDialog*, in quanto contenuta nella interfaccia della unit PRSETUP.

Scelta del carattere

La scelta del carattere è possibile sono dopo l'associazione della variabile *Printer* con una stampante definita mediante un file con estensione PRN. Il metodo *SelectFont*, quindi, chiama l'omonimo metodo della classe *TPrinter* solo se *PrinterSelected* ritorna TRUE. In tal caso, viene proposta una dialog box simile a quella riprodotta nella figura 3.

La scelta è semplice: basta posizionarsi sul carattere desiderato con i tasti o con il mouse e premere Invio o il pulsante *Ok*. È più importante tenere a mente alcune considerazioni.

Ogni stampante ha i suoi tipi di carattere. In linea di massima, sono spesso disponibili le tre densità di 10, 12 e 17 CPI (caratteri per pollice), magari con piccole differenze (il carattere compresso, ad esempio, può trovarsi con 16,66 o 17,1 CPI); rimane comunque arduo preparare un'applicazione in grado di usare qualsiasi stampante.

Possono venire in aiuto due metodi della classe *TPrinter*, che vedremo quando esamineremo la unit TVPRINT: *GetFontCount* ritorna il numero dei tipi carattere disponibili per la stampante correntemente selezionata, mentre *GetFont* ritorna le caratteristiche di un dato tipo, individuato secondo il suo numero d'ordine nella lista come visualizzata nella dialog box (l'ordine di questa, peraltro, è lo stesso secondo il quale i

```
Program Demo;
(*$X+*)
uses
  Objects, Views, Drivers, Dialogs, Menus, App, MsgBox, StdDlg,
  Dos, PrSetup, TVPrint;

const
  cmPrint   = 100;
  cmPrSetup = 101;
  cmSelFont = 102;

type

  TDemo = object(TApplication)
    Printer: TPrinter;
    constructor Init;
    destructor Done; virtual;
    procedure InitMenuBar; virtual;
    procedure InitStatusLine; virtual;
    procedure HandleEvent(var Event: TEvent); virtual;
    function PrinterSelected: Boolean;
    procedure Print;
    procedure PrinterSetup;
    procedure SelectFont;
  end;

constructor TDemo.Init;
begin
  TApplication.Init;
  Printer.Init;
end;

destructor TDemo.Done;
begin
  Printer.Done;
  TApplication.Done;
end;

procedure TDemo.InitMenuBar;
var
  R: TRect;
begin
  GetExtent(R);
  R.B.Y := R.A.Y + 1;
  MenuBar := New(PMenuBar, Init(R, NewMenu(
    NewSubMenu('~F~ile', hcNoContext, NewMenu(
      NewItem('~S~tampa...', '', 0, cmPrint, hcNoContext,
      NewItem('~I~mposta stampante...', '', 0, cmPrSetup, hcNoContext,
      NewLine(
      NewItem('~E~sci', 'Alt-X', kbAltX, cmQuit, hcNoContext,
      nil))))),
    NewSubMenu('~M~odifica', hcNoContext, NewMenu(
      NewItem('~T~ipo carattere...', '', 0, cmSelFont, hcNoContext,
      nil)),
    nil))))));
end;
procedure TDemo.InitStatusLine;
var
  R: TRect;
begin
  GetExtent(R);
  R.A.Y := R.B.Y - 1;
  StatusLine := New(PStatusLine, Init(R,
    NewStatusDef(0, $FFFF,
      NewStatusKey('~Alt-X~ Esci', kbAltX, cmQuit,
      NewStatusKey('', kbF10, cmMenu,
      nil)),
  nil)));
end;

procedure TDemo.HandleEvent(var Event: TEvent);
begin
  TApplication.HandleEvent(Event);
  if (Event.What = evCommand) then begin
    case Event.Command of
      cmPrint: Print;
      cmPrSetup: PrinterSetup;
      cmSelFont: SelectFont;
    else
      Exit;
    end;
    ClearEvent(Event);
  end;
end;
```

```
function TDemo.PrinterSelected: Boolean;
begin
  if Printer.PD.Name = '' then begin
    MessageBox(#3'Nessuna stampante selezionata.'#13+
      #3'Procedi con ''Imposta stampante''.',nil,
      mfError or mfOkButton);
    PrinterSelected := False;
  end
  else
    PrinterSelected := True;
end;
procedure TDemo.Print;
const
  S = 'Allineamento barre verticali';
var
  i: Integer;
begin
  if PrinterSelected then with Printer do begin
    Start;
    Print('Questo è il tipo carattere correntemente selezionato.');
    NewLine;
    NewLine;
    SetFont(0);
    Print('Segue un allineamento tra tipi diversi:');
    NewLine;
    NewLine;
    for i := 0 to GetFontCount - 1 do begin
      SetFont(i);
      Move(0.5);
      Print('normale');
      Move(3.0);
      Underline(True);
      Print('sottolineato');
      Underline(False);
      Move(5.5);
      BoldFace(True);
      Print('grassetto');
      BoldFace(False);
      NewLine;
    end;
    NewLine;
    for i := 0 to GetFontCount - 1 do begin
      NewLine;
      SetFont(i);
      Move(1.0);
      SetFont(0); PrintChar(#179); SetFont(i);
      Move(1.55);
      SetFont(0); PrintChar(#179); SetFont(i);
      Move(2.0);
      SetFont(0); PrintChar(#179); SetFont(i);
      Move(2.5);
      SetFont(0); PrintChar(#179); SetFont(i);
      Move(6.0-TextLength(S));
      Print(S);
    end;
    NewPage;
    Finish;
  end;
end;

procedure TDemo.PrinterSetup;
begin
  Printer.DoSetup;
end;
procedure TDemo.SelectFont;
begin
  if PrinterSelected then
    Printer.SelectFont;
end;

var
  DemoApp: TDemo;

begin
  RegisterType(RCollection);
  RegisterType(RFont);
  DemoApp.Init;
  DemoApp.Run;
  DemoApp.Done;
end.
```

Figura 1 - Il listato del programma DEMO.

diversi tipi sono stati definiti mediante *TPrSetupDialog*).

Esaminando le caratteristiche dei tipi disponibili, è possibile scegliere quello che abbia la densità più prossima a quella desiderata, oppure costruire una dialog box mediante la quale proporre all'utente la scelta da un ristretto sottoinsieme di tipi, quali quelli aventi tutti uguale densità (con una IBM Proprinter XL24 vi possono essere, ad esempio, fino a quattro tipi da 10 CPI, di cui due «bozza» e due «lettera»).

Stampa

Siamo così finalmente giunti al metodo *Print* della nostra applicazione. Notiamo che le istruzioni in esso contenute sono precedute da *Printer.Start* e terminano con *Printer.Finish*. Tali metodi non vanno confusi con il constructor e il destructor della classe *TPrinter*; questi ultimi vanno utilizzati come quelli di qualsiasi oggetto che sia istanza di una classe e, in particolare, il constructor ha il compito di tentare l'associazione automatica tra un'istanza di *TPrinter* e una stampante di default. Il metodo *Start*, invece, provvede all'apertura del dispositivo di output (una porta o un file) e ad altre inizializzazioni quasi tutte conseguenti alla impostazione come voluta dall'utente; *Finish* chiude il dispositivo di output (e altre cose...). L'impostazione delle caratteristiche «temporanee» di una stampante e la scelta tra uno dei suoi tipi di carattere richiedono, ovviamente, che sia stato chiamato il constructor della classe *TPrinter* (in quanto tutto avviene mediante metodi di questa); non richiedono, al contrario, che si sia aperto il dispositivo di output. È anzi preferibile chiamare *Start* subito prima di stampare, in quanto, in caso contrario, alcune variazioni di impostazioni non avrebbero effetto.

Il metodo *Print* produce una stampa come quella riprodotta nella figura 4, per la quale è stata usata una stampante HP DeskJet 500 definita con 3 tipi di carattere (10, 12 e 16.6 CPI), dopo aver selezionato il secondo.

Per prima cosa, viene stampata una riga che utilizza il carattere selezionato.

Si torna poi al primo tipo di carattere disponibile con *SetFont(0)*, assumendo che si tratti del tipo «normale», per introdurre un allineamento di stringhe stampate in tutti i tipi definiti (numerati

Figura 2 - La dialog box per la scelta tra una delle stampanti disponibili (installate con PRINST.EXE).

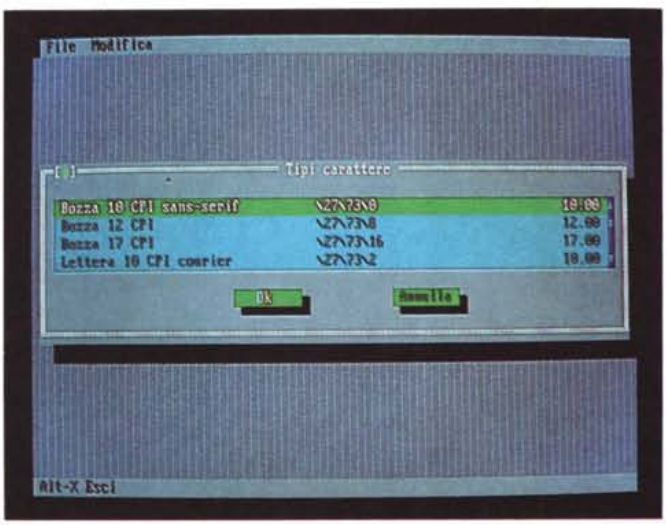

Figura 3 - La dialog box per la scelta del carattere (nella list box appaiono i primi quattro tipi di carattere di una IBM Proprinter XL24).

da 0 a *GetFontCount* −1), usando anche sottolineato e grassetto; l'allineamento è ottenuto mediante un metodo *Move* che accetta come parametro una distanza dal margine sinistro del foglio espressa in pollici. Vi sarà facile indovinare che il metodo *Move* muove la testina della stampante secondo quel «movimento per punti» che abbiamo visto in occasione dell'esame della unit PRDEF.

Segue un ulteriore esempio di allineamento, nel quale anche vengono usati tutti i tipi di carattere definiti; per sottolineare le potenzialità del movimento per punti, vengono allineate quattro barre verticali e vengono giustificate a destra diverse versioni della stringa «Allineamento barre verticali». A quest'ultimo scopo, viene utilizzato il metodo *TextLength* della classe *TPrinter*, che ritorna la lunghezza in pollici di una stringa stampata con il carattere correntemente selezionato.

Risultati analoghi si otterrebbero con qualsiasi stampante capace di un movimento per punti. Vedremo tuttavia che, anche quando questo non sia disponibile, la unit TVPRINT consente comunque di pervenire ad un output accettabile.

MC

Sergio Polini è raggiungibile tramite MC-link alla casella MC1166 e tramite Internet all'indirizzo MC1166@mc-link.it.

Questo è il tipo carattere correntemente selezionato.

Segue un allineamento tra tipi diversi:

normale
normale
normale

sottolineato
sottolineato
sottolineato

grassetto
grassetto
grassetto

Allineamento barre verticali
Allineamento barre verticali
Allineamento barre verticali

Figura 4 - Output del programma DEMO su una stampante HP DeskJet 500.

In tutto il mondo Greenwich è il riferimento unico per misurare il tempo. Ma da noi c'è qualcos'altro. Un riferimento per conoscere e scegliere gli strumenti che lo misurano: Orologi. Una rivista pensata con passione; nelle sue pagine il mondo del tempo, in tutte le sue forme: tecnica, storia, curiosità e futuro. Splendide immagini di orologi moderni e antichi accompagnano informazioni precise e articoli attenti e puntuali sulla tecnica, la cultura del tempo e sulle rarità. Insomma una guida sicura che non ha paralleli: proprio come il meridiano di Greenwich.

Il riferimento più autorevole dopo il meridiano di Greenwich.

technimedia
Pagina dopo pagina, le nostre passioni.

Orologi. I primi sui secondi.