

Pascal listing: DIRP2.PAS

```
{*****}
{*               D I R P 2               *}
{*-----*}
{*   Task           : Displays all files in any directory on the   *}
{*                   screen, including subdirectories and volume     *}
{*                   label names. File handling is performed by a  *}
{*                   call to the FindFirst and FindNext functions  *}
{*                   found in the Turbo Pascal DOS unit.           *}
{*                   See also DIRP1.PAS.                           *}
{*-----*}
{*   Author          : Michael Tischer                             *}
{*   Developed on     : 07/08/88                                     *}
{*   Last update      : 04/07/95                                     *}
{*****}
```

program DIRP2;

Uses Crt, Dos; { Add CRT and DOS units }

{-- Type declarations -----}

type MonVec = array[1..12] of string[3]; { Array with month names }

{-- Constants -----}

```
const FENTS = 14; { Number of visible entries at a time }
      Months : MonVec = ( 'Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun',
                          'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec');
```

{*****}

```

{* PRINTDATA: Displays entry information.                                     *}
{* Input      : DIRBUF = Data structure with file information               *}
{* Output     : None                                                         *}
{*****}

```

```

procedure PrintData( DirBuf : SearchRec );

```

```

var Counter : byte;
    Date,           { For splitting the TIME field in SearchRec }
    Time      : word;

```

```

type longrec = record    { For splitting a LONG record into two words }
    LoWord,
    HiWord  : word
end;

```

```

begin
    writeln;           { Scroll up by one line }

```

```

    write( DirBuf.Name );      { Name already converted by Pascal }

```

```

    GotoXY(13, FENTS);
    write('|', DirBuf.Size:7);

```

```

    Date := longrec(DirBuf.Time).HiWord; { Date and time from SearchRec }
    Time := longrec(DirBuf.Time).LoWord;

```

```

    GotoXY(21, FENTS);
    write('|', (Months[Date shr 5 and 15]), ' '); { Display month }
    write(Date and 31:2, ' '); { Display day }
    write(Date shr 9 + 1980:5); { Display year }
    GotoXY(34, FENTS);

```

```

write('|', Time shr 11:2, ':');           { Display hour }
write(Time shr 5 and 63:3);               { Display minutes }

GotoXY(44, FENTS);
write('|');                               { Separator preceding each field }
Counter := 1;                             { Attribute display counter }
while ( Counter < 32 ) do
  begin
    if (DirBuf.Attr and Counter) <> 0 then write('X')
    else write(' ');
    Counter := Counter shl 1;
  end;
  write('|');                             { Right border of window }
end;

{ ***** }
{ * ScreenDesign   : Prepares screen for directory display.          * }
{ * Input          : None                                              * }
{ * Output         : None                                              * }
{ ***** }
procedure ScreenDesign;

var Counter : integer;                   { Loop counter }

begin
  ClrScr;                                { Clear screen }
  Window(14,(20-FENTS) shr 1+1,64,(20-FENTS) shr 1 +5+FENTS);
  GotoXY(1,1);                           { Cursor in upper-left corner of window }

  write('+-----+');
  write('|  Filename  | Size   | Date       | Time    |RHSVD|');
  write('|-----+-----+-----+-----+-----|');

```

```

for Counter := 1 to FENTS do
  write('|' | | | | |');
write('+-+-----+');

Window(15,(20-FENTS) shr 1+4,66,(20-FENTS) shr 1 +3+FENTS);
GotoXY(1, FENTS);           { Cursor in upper-left corner of window }
end;

{*****}
{ * Dir: Controls directory reading and output. * }
{ * Input   : SPATH      = Search path with file pattern * }
{ *         ATTRIBUTE = Search attribute * }
{ * Output  : None * }
{*****}
procedure Dir( SPath : string; Attr : byte );

var NumOfEntries,           { Total number of entries found }
    NumInScrn   : integer;  { Number of entries per screen }
    WKey        : char;     { Wait for a keypress }
    DirBuf      : SearchRec; { Indicates a directory entry }

begin
  clrscr;                  { Clear screen }
  ScreenDesign;           { Prepare screen for directory output }

  NumInScrn := -1;         { No more entries to display }
  NumOfEntries := 0;       { No more entries found }
  FindFirst( SPath, Attr, DirBuf ); { Search for first entry }
  if DOSError = 0 then
    repeat
      NumOfEntries := succ(NumOfEntries); { One more entry found }

```

```

NumInScrn := succ(NumInScrn);           { One more entry in window }
if NumInScrn = FENTS then                { Is the window full? }
begin                                     { Yes }
    Window(14, (20-FENTS) shr 1 + 5 + FENTS,
            66, (20-FENTS) shr 1 + 6+ FENTS );
    GotoXY(1, 1);                        { Move cursor to bottom line of window }
    TextBackground( LightGray );         { White background }
    TextColor( Black );                  { Black text }
    write('           Please press a key ');
    WKey := ReadKey;                     { Read a key }
    GotoXY(1, 1);                        { Cursor in upper-left corner of window }
    TextBackground( Black );             { Black background }
    TextColor( LightGray );              { White text }
    write(' ');
    Window(15, (20-FENTS) shr 1+4, 65, (20-FENTS) shr 1 +3+FENTS);
    GotoXY(1, FENTS);                   { Return cursor to old position }
    NumInScrn := 0;                     { Start counting at 0 }
end;
PrintData( DirBuf );                    { Display entry data }
FindNext( DirBuf );                     { Search for next file until }
until DOSError <> 0;                     { no more files remain }

Window(14, (20-FENTS) shr 1 +5+FENTS, 65, (20-FENTS) shr 1 +6+FENTS);
GotoXY(1, 1);                           { Cursor in upper-left corner of window }
TextBackground( LightGray );             { White background }
TextColor( Black );                      { Black text }
write(' ');

GotoXY(2, 1);
case NumOfEntries of
    0 : write('No files found');
    1 : write('One file found');

```

```

    else write(NumOfEntries,' files found')
end;

Window(1, 1, 80, 25);           { Make entire screen a window }
end;

{*****}
{**                MAIN PROGRAM                **}
{*****}
begin
  case ParamCount of
    0 : Dir( '*.*', AnyFile );           { Count parameters }
    1 : Dir( ParamStr(1), AnyFile );     { All files in current directory }
      { Display specified directory }
    else writeln('Invalid number of parameters');
  end;
end.

```