

Listing: RECLOCK.PAS

```
{*****}
{*                R E C L O C K                *}
{*-----*}
{* Task          : Demonstrates the DOS record locking functions. *}
{*-----*}
{* Author        : Michael Tischer                *}
{* Developed on   : 09/19/91                        *}
{* Last update    : 04/07/95                        *}
{*****}
```

program reclock;

```
uses Crt, Dos,                { Add CRT and DOS units }
    NetFileP;                 { Add NetFileP unit }
```

```
const TFileName = 'Rec.dat';   { Filename for test file }
```

```
type Test      = array[ 1..160 ] of char;    { Data type for test }
    TestFile = file of Test;
```

```
var DFile : TestFile;          { Test file }
```

```
{*****}
{* CreateATRec : Creates a test data record.                *}
{* Input       : Characters for the record                  *}
{* Output      : Test data record                          *}
{*****}
```

```
procedure CreateATRec(      ReChars : char;
                           var DRec  : test );
```

```

var i : word;                                     { Loop counter }

begin
  for i := 1 to 160 do
    DRec[ i ] := ReChars;
  end;

{*****}
{ * OpenNetFile : Open available network file. If one does not exist, * }
{ *               create a new one and fill this new file with       * }
{ *               test data records.                                  * }
{ * Input        : File                                              * }
{ * Output       : File                                              * }
{*****}

function OpenNetFile( var DFile : testfile ) : boolean;

var i          : word;                                     { Loop counter }
    TestDRec   : Test;                                   { Needed for creating the test file }

begin
  {-- Open file for input & output in deny none mode -----}

  NetReset( TFileName, fm_rw or sm_no, sizeof( Test ), DFile );
  if ( NetError = NE_FileNotFound ) then                    { File not found? }
    begin
      {-- Create file and fill with test data records -----}

      NetRewrite( TFileName, fm_rw or sm_no, sizeof( Test ), DFile );
      if ( NetError = 0 ) then                                { No errors during creation? }
        begin

```

```

        if NetLock( DFile, 0, 26 ) then           { Store 26 records }
        begin
            NetSeek( DFile, 0 );                 { Pointer to start of file }
            for i := 1 to 26 do
                begin
                    CreateATRec( chr( ord( 'Z' ) + 1 - i ), TestDRec );
                    NetWrite( DFile, TestDRec );   { Write test data }
                end;
            OpenNetFile := NetUnlock( DFile, 0, 26 );
        end
    else
        OpenNetFile := false;                     { Error when locking }
    end
else
    OpenNetFile := false;                         { Error while creating the file }
end
else
    OpenNetFile := ( NetError = 0 );              { No errors while opening? }
end;

```

```

{ ***** }
{ * NetEdits      : Demonstrates network functions.      * }
{ * Input         : File                                  * }
{ * Output        : File                                  * }
{ ***** }

```

```

procedure NetEdits( var DFile : TestFile );

```

```

var CurRecord : longint;                        { Current record number }
    CurDRec    : Test;                         { Current data record }
    Action     : byte;                         { Desired action }
    Status     : boolean;                      { Record locked? }

```

```

ReChars    : char;

begin
  {-- Display menu -----}

  writeln( #13#10'Available functions' );
  writeln( '  1: Position file pointer' );
  writeln( '  2: Lock record' );
  writeln( '  3: Read record' );
  writeln( '  4: Edit data record' );
  writeln( '  5: Write record' );
  writeln( '  6: Unlock record' );
  writeln( '  7: Exit' );

  CurRecord := 0;                                { Current data record }
  Status := false;                                { Record not locked }
  CreateATRec( #32, CurDRec );                    { Create empty data record }

  repeat
    {-- Display information -----}

    gotoxy( 1, 16 );                                { Display file pointer position }
    writeln( 'Current Record: ', CurRecord : 4 );
    write( 'Status          : ' );
    if Status then
      writeln( 'Locked  ' )
    else
      writeln( 'Unlocked' );
    WriteLn( 'Network Status : ', NetError: 4, ' = ',
      copy( NetErrorMsg( NetError ) + ' ', 1, 30 ) );
    gotoxy( 1, 21 );                                { Display test record }
    writeln( 'Current Data Record:' );

```

```

writeln( CurDRec );

NetSeek( DFile, CurRecord );           { Position file pointer }

gotoxy( 1, 13 );
write( 'Select:                        ' );
gotoxy( 10, 13 );
readln( Action );
case Action of
  1 : begin
      gotoxy( 1, 13 );
      write( 'New data record number: ' );
      readln( CurRecord );
      Status := false;                    { Record not locked }
      CreateATRec( #32, CurDRec )
    end;
  2 : Status := Status or NetLock( DFile, CurRecord, 1 );
  3 : NetRead( DFile, CurDRec );          { Read data record }
  4 : begin
      gotoxy( 1, 13 );
      write( 'New character: ' );
      readln( ReChars );
      CreateATRec( ReChars, CurDRec );
    end;
  5 : NetWrite( DFile, CurDRec );          { Write data record }
  6 : Status := Status and not NetUnlock( DFile, CurRecord, 1 );
end;
until ( Action = 7 );
end;

{ ***** }
{ *          M A I N    P R O G R A M          * }

```

```

{*****}

begin
  clrscr;
  writeln( 'Demonstration of DOS File Locking Functions ',
           '(C) 1991 by Michael Tischer ' + paramstr( 1 ) );
  writeln( '=====',
           '=====' );

  if ( ShareInst ) then          { Share program installed? }
  begin
    if OpenNetFile( DFile ) then { File open or created? }
    begin
      NetEdits( DFile );      { Demonstration of network functions }
      NetClose( DFile );      { Close file }
    end
  else
    writeln( #13#10'Error while opening network file ' +
             'Error number: ', NetError );
    ClrScr;
  end
else
  writeln( #13#10'Please install SHARE before running this program.' );
end.

```