

```

*****
*                               F I L E L O C K                               *
*-----*
*   Task       : Opens files in network using file locking                 *
*               functions.                                                 *
*-----*
*   Author      : Michael Tischer                                           *
*   Developed on : 09/14/91                                                  *
*   Last update  : 01/29/92                                                  *
*****

program filelock;

uses Crt, Dos,                      { Add CRT and DOS units }
    NetfileP;                      { Add network unit }

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

type Test = array[ 1..4 ] of char; { Data type for test }

var AxsTypeA,                        { File access modes }
    AxsTypeB,
    LockMdA,                        { File lock modes }
    LockMdB : byte;

*****
* FiMode      : Create file mode from access type and locking.             *
* Input       : Access type, file lock mode                               *
* Output      : File mode                                                  *
*****

function FiMode( AxsType, LockMd : byte ) : byte;

var res : byte;                    { Help in calculating results }

begin
    case AxsType of
        1 : res := fm_r;           { Read-only }
        2 : res := fm_w;           { Write-only }
        3 : res := fm_rw;          { Read and write }
    end;
    case LockMd of
        1 : res := res or sm_comp; { No locking }
        2 : res := res or sm_rw;   { All prohibited }
        3 : res := res or sm_r;    { Read-only enabled }
        4 : res := res or sm_w;    { Write-only enabled }
        5 : res := res or sm_no;   { No locking }
    end;
    FiMode := Res;
end;

*****
* DFileTest   : Demonstrates access conflicts or file locks with           *
*               and without file locking.                                   *
* Input       : Access type and lock modes for both concurrent files*
* Output      : None                                                        *
*****

procedure DFileTest( AxsTypeA, LockMdA, AxsTypeB, LockMdB : byte );

const TestAOut : Test = 'AAAA';    { Test data records }
      TestBOut : Test = 'BBBB';

var TestAInp,                        { Data records for read test }
    TestBInp : Test;
    TFileA,   { Test files for normal access }
    TFileB : file of Test;

begin
    window( 1, 11, 80, 25 );
    clrscr;
    writeln( 'File A: Name = ', TFileName, ', Access type = ',
        AxsTypeA, ', Lock mode = ', LockMdA );
    writeln( 'File B: Name = ', TFileName, ', Access type = ',
        AxsTypeB, ', Lock mode = ', LockMdB );

    {-- Open files -----}

    write( #13#10'Opening file A:  ' );
    NetReset( TFileName, FiMode( AxsTypeA, LockMdA ),
        sizeof( Test ), TFileA );
    if ( NetError = NE_FileNotFound ) then
        NetRewrite( TFileName, FiMode( AxsTypeA, LockMdA ),
            sizeof( test ), TFileA );
    Writeln( 'Status ', NetError : 2, ' = ', NetErrorMsg( NetError ) );

```

```

write( 'Opening file B: ' );
NetReset( TFileName, FiMode( AxsTypeB, LockMdB ),
          sizeof( Test ), TFileB );
writeln( 'Status ', NetError : 2, ' = ', NetErrorMsg( NetError ) );

{-- Write files -----}

write( #13#10'Writing to file A:' );
if ( Is_NetWriteOk( TFileA ) ) then           { Write enabled? }
begin                                         { Yes --> Write it }
    Netwrite( TFileA, TestAOut );
    writeln( ' Record "', TestAOut, '" written ' );
end
else                                         { No --> Error }
    writeln( ' File not open for writing' );

write( 'Writing to file B:' );
if ( Is_NetWriteOk( TFileB ) ) then           { Write enabled? }
begin                                         { Yes --> Write it }
    Netwrite( TFileB, TestBOut );
    writeln( ' Record "', TestBOut, '" written ' );
end
else                                         { No --> Error }
    writeln( ' File not open for writing' );

{-- File pointers for both files moved to beginning -----}

if Is_NetOpen( TFileA ) then                 { File open? }
    NetSeek( TFileA, 0 );                     { Yes --> Continue }
if Is_NetOpen( TFileB ) then                 { File open? }
    NetSeek( TFileB, 0 );                     { Yes --> Continue }

{-- Read files -----}

write( #13#10'Reading file A:' );
if ( Is_NetReadOk( TFileA ) ) then           { Read enabled? }
begin                                         { Yes --> Read it }
    Netread( TFileA, TestAInp );
    writeln( ' Record "', TestAInp, '" read ' );
end
else                                         { No --> Error }
    writeln( ' File not open for reading' );

write( 'Reading file B:' );
if ( Is_NetReadOk( TFileB ) ) then           { Read enabled? }
begin                                         { Yes --> Read it }
    Netread( TFileB, TestBInp );
    writeln( ' Record "', TestBInp, '" read ' );
end
else                                         { No --> Error }
    writeln( ' File not open for reading' );

{-- Close file -----}

NetClose( TFileA );
NetClose( TFileB );
end;

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

begin
    clrscr;
    writeln( 'Demonstration of DOS File Locking Functions          ',
             '(C) 1992 by Michael Tischer' );
    writeln( '=====',
             '===== ' );

    if ( ShareInst ) then                     { Share program installed? }
    begin
        {-- Select file mode -----}

        writeln( #13#10'Available access types:          ',
                  'Available lock types:' );
        writeln( ' 1: Read-only                          ',
                  ' 1: Compatibility mode (no locking)    ' );
        writeln( ' 2: Write-only                          ',
                  ' 2: Prohibit other file accesses generally' );
        writeln( ' 3: Read and write                      ',
                  ' 3: Read access enabled only' );
        writeln( ' 4: Write access enabled only' );
        writeln( ' 5: All enabled (record locking)                ' );
    end
end

```

```
Write( #13#10'Access type: Test file A: ' );
read( AxsTypeA );
Write( 'Lock mode: Test file A: ' );
read( LockMdA );
Write( 'Access type: Test file B: ' );
read( AxsTypeB );
Write( 'Lock mode: Test file B: ' );
read( LockMdB );

DFileTest( AxsTypeA, LockMdA, AxsTypeB, LockMdB );
end
else
writeln( #13#10'Please install SHARE before running this program.' );
end.
```