```
{
DATE STARTED:-  8 MAR 84
BY:-            B.SCHRODER
LAST CHANGE:-   8 MAR 84
                09:00
BY:-            B.SCHRODER
}

PROGRAM SERXFER(INPUT,OUTPUT);

{This program permits the user to move files terminating in a CNTRL Z
from a disk file to the proteus printer port, or from the proteus printer
port to a disk file. XON ($11) and XOFF ($13) protocal is used to ensure
that the transfer is done in an orderly manner.

To use the program type either:
SERXFER IN  < DESTINATION FILE NAME >          to transfer to a disk file
SERXFER OUT < ORIGIN FILE NAME >               to transfer to the printer port
}

{*****************************************************************}

const   cpmlin=127;
        bufsiz=30000;   {size of recieve buffer}
        nrlyful=29000;  {point at which we send xoff}
        cmd=$08;        {proteus printer command port}
        data=$09;
        txon=$02;       {bit mask to look for tx on using wait}
        rxon=$01;       {bit mask for rx using wait}
        txont=$01;      {bit mask to look for tx on using tstbit}
        rxont=$00;      {bit mask to look for rx on using tstbit}
        onesec=1000;    {number times round upld for one sec of time}
        xoff=$13;
        xon=$11;
        cntrlz=$1a;

type    chfile=file of char;
        filename=string[cpmlin];

var     destfile:chfile;
        origfile:chfile;
        downnam:filename;
        upnam:filename;
        dirnam:string[10];
        buf:array[0..bufsiz] of char;
        cpmstr:absolute[$80] packed array[0..cpmlin] of char;
        cpmcmd:filename;
        result:integer;

{*************************************************************************}
procedure getname(var namstr:filename);

{gets the next string delimited by spaces from cpmcmd}

begin
namstr:='';
while (length(cpmcmd)<>0) and (cpmcmd[1]<>' ') do
        begin
        namstr:=concat(namstr,cpmcmd[1]);
```

```
        delete (cpmcmd,1,1);
        end;
if length(cpmcmd)<>0 then delete(cpmcmd,1,1);   {delete next delimeter}
end;

{*************************************************************************}

procedure help;

{this procedure provides help if the user is not familiar with the program}

begin
writeln('This program permits the user to move files terminating in a CNTRL Z');
writeln('from a disk file to the proteus printer port, or from the proteus printer');
writeln('port to a disk file. XON ($11) and XOFF ($13) protocal is used to ensure');
writeln('that the transfer is done in an orderly manner.');
writeln;
writeln('To use the program type either:');
writeln('SERXFER IN  < DESTINATION FILE NAME >          to transfer to a disk file');
writeln('SERXFER OUT < ORIGIN FILE NAME >               to transfer to the printer port');
writeln;
end;

{*************************************************************************}



procedure upld(var infile:chfile);

{this procedure gets a byte from the printer port and saves it in the input
file when the buffer is nrlyful. It also sends XOFF at that time and waits for
transmission to stop.}

var     ch:char;
        full,nearfull:boolean;
        inpnt,outpnt,timeout:integer;

begin
nearfull:=false;
full:=false;
inpnt:=0;
ch:=chr(0);
repeat
        begin
        if tstbit(inp[cmd],rxont) then
                begin
                ch:=inp[data]&chr($7f);
                buf[inpnt]:=ch;
                if not nearfull then nearfull:=(inpnt=nrlyful);
                inpnt:=inpnt+1;
                timeout:=0;
                end;
    timeout:=timeout+1;
    if nearfull then if tstbit(inp[cmd],txont) then
                begin
                out[data]:=chr(xoff);
                nearfull:=false;
                full:=true;
                end;
    if full then if (timeout>onesec) or (inpnt>bufsiz) then
```

```
                begin
                outpnt:=0;
                while outpnt<inpnt do
                        begin
                        infile^:=buf[outpnt];
                        put(infile);
                        outpnt:=outpnt+1;
                        end;
                inpnt:=0;
                timeout:=0;
                full:=false;
                while not tstbit(inp[cmd],txont) do;
                out[data]:=chr(xon);
                end;
        end;
until (ch=chr(cntrlz)) or (inp[5] = cntrlz);
outpnt:=0;                      {flush what remains in the buffer}
while outpnt<inpnt do
        begin
        if wnb(infile,buf[outpnt]) then
                begin
                writeln('error writing character to disk');
                exit;
                end;
        outpnt:=outpnt+1;
        end;
end;

{*************************************************************************}


procedure downld(var fromfil:chfile);

{this procedure gets a byte from the reading file (origfile) and sends it
to the proteus printer port}

var     ch:char;

begin
ch:=chr(0);
while ch<>chr(cntrlz) do
        begin
        ch:=fromfil^;
        get(fromfil);
        wait(cmd,txon,false);
        out[data]:=ch;
        if tstbit(inp[cmd],rxont) then if inp[data]&$7f=xoff then
                begin
                repeat wait(cmd,rxon,false);
                until inp[data]&$7f =xon;
                end;
        end;
end;

{*************************************************************************}


begin           {mainline}
writeln;
```

```
move(cpmstr,cpmcmd,cpmlin+1);
if length(cpmcmd)=0 then
        begin
        help;
        exit;
        end;
delete(cpmcmd,1,1);             (delete leading blank)
getname(dirnam);
if not (dirnam[1] in ['I','i','O','o']) then
        begin
        help;
        exit;
        end;
if dirnam[1] in ['I','i'] then
        begin
        getname(upnam);
        writeln('writing file: ',upnam);
        assign(destfile,upnam);
        rewrite(destfile);
        if ioresult=255 then
                begin
                writeln('cannot open ',upnam);writeln;
                exit;
                end;
        writeln('starting input transfer');
        upld(destfile);
        close (destfile,result);
        if ioresult=255 then
                begin
                writeln('cannot close ',upnam);writeln;
                exit;
                end;
        writeln('input transfer completed');writeln;
        end
else
        begin
        getname(downnam);
        writeln('reading file: ',downnam);
        assign(origfile,downnam);
        reset(origfile);
        if ioresult=255 then
                begin
                writeln('cannot open ',downnam);writeln;
                exit;
                end;
        writeln('starting output transfer');
        downld(origfile);
        writeln('output transfer completed');writeln;
        end;
end.
```