Message 77 -- **********
11-Jun-84 14:58:01-EDT,799;000000000011
Return-Path: <knutson@ut-ngp.ARPA>
Received: from ut-ngp.ARPA by COLUMBIA-20.ARPA with TCP; 11 Jun 84 14:57:54 EDT
Date: Mon, 11 Jun 84 13:58:37 cdt
From: knutson@ut-ngp.ARPA
Posted-Date: Mon, 11 Jun 84 13:58:37 cdt
Message-Id: <8406111858.AA03906@ut-ngp.ARPA>
Received: by ut-ngp.ARPA (4.22/3.14)
        id AA03906; Mon, 11 Jun 84 13:58:37 cdt
To: cc.fdc@columbia-20.ARPA
Subject: Cyber Kermit document file

I have put together a .MSS file similar to the others referenced by KUSER.MSS. It is available via anonymous FTP from UT-NGP as file 170KERMIT.MSS. I think for the most part it is formatted correctly. You might want to take a look at it and check for yourself.

Jim Knutson
ARPA: knutson@ut-ngp
UUCP: {ihnp4,seismo,kpno,ctvax}!ut-sally!ut-ngp!knutson
Phone: (512) 471-3241

Message 45 -- **********
18-Oct-83 11:50:38-EDT,1324;000000000001
Return-Path: <SY.FDC@CU20B>
Received: from CU20B by CUCS20 with DECnet; 18 Oct 83 11:50:27 EDT
Date: Tue 18 Oct 83 11:50:08-EDT
From: Frank da Cruz <SY.FDC@CU20B>
Subject: [Bob Cattani <CATTANI@CUCS20>: Next Kermit is ready]
To: Info-Kermit@CUCS20

Announcing yet another release of UNIX Kermit. The major addition here is error packet processing; this brings UNIX Kermit up to the basic level of all other Kermits. All UNIX sites should convert to this latest version. Please report any problems directly to Bob (CATTANI@COLUMBIA-20) and me (CC.FDC@COLUMBIA-20). - Frank

---

Date: Mon 17 Oct 83 15:52:17-EDT
From: Bob Cattani <CATTANI@CUCS20>
Subject: Next Kermit is ready

Included fixes from Alan Crosswell (CUCCA) for IBM_UTS:
- Changed MYEOL character from \n to \r.
- Change char to int in bufill so getc would return -1 on EOF instead of 255 (-1 truncated to 8 bits)
- Added read() in rpack to eat the EOL character
- Added fflush() call in printmsg to force the output
  NOTE: The last three changes are not conditionally compiled since they should work equally well on any system.

Changed Berkeley 4.x conditional compilation flag from UNIX4X to UCB4X.
Added support for error packets and cleaned up the printing routines.

-Bob

Message 46 -- **********
18-Oct-83 12:58:14-EDT,1280;000000000001
Mail-From: CC.FDC created at 18-Oct-83 12:58:02
Date: Tue 18 Oct 83 12:58:02-EDT
From: Frank da Cruz <cc.fdc@COLUMBIA-20.ARPA>
Subject: FORTRAN-based KERMIT available
To: Info-Kermit@COLUMBIA-20.ARPA

Announcing a major new KERMIT release, written in FORTRAN for the CDC Cyber 170 by Jim Knutson of the Computation Center of the University of Texas at Austin (knutson@utexas-11, or knutson@UT-NGP). Although it won't be able to run as-is on many machines, it should be adaptable to a wide range of systems for which KERMIT is presently unavailable.

The source for the Cyber Version of Kermit is on COLUMBIA-20 in the file KER:170KERMIT.FOR, available via anonymous FTP. The installation information is in the Fortran source code. It will need mods for any Cyber site that tries to run it since there are probably no two Cyber sites anywhere that do ASCII I/O the same way. There is also an nroff source for additions to the Kermit User's Guide in KER:170KERMIT.NR.

The Cyber-170 version of KERMIT has been tested with KERMIT-20 and UNIX Kermit and runs with either. It is rather slow (500-900 effective baud on a 2400 baud line). It has also been tested with Kermit-86 on a Corona (IBM-PC clone), IBM-PC and a Z100 and CPMKermit on a Z100.

Message 9 -- **********
17-Oct-83 18:30:22-EDT,12913;000000000011
Return-Path: <knutson@utexas-11>
Received: from UT-NGP.ARPA by COLUMBIA-20.ARPA with TCP; Mon 17 Oct 83 18:29:58-EDT
Date: 17 Oct 83 17:29:46 CDT (Mon)
From: Jim Knutson <knutson@utexas-11.ARPA>
Subject: Re: Need FORTRAN based file transfer program
Posted-Date: 17 Oct 83 17:29:46 CDT (Mon)
Message-Id: <8310172229.AA18989@UT-NGP.ARPA>
Received: by UT-NGP.ARPA (3.326/3.7)
id AA18989; 17 Oct 83 17:29:46 CDT (Mon)
To: CC.FDC@COLUMBIA-20.ARPA

Sorry that it has taken me this long to get back on this. I've been real pressed for time lately.

The Cyber version has been tested with KERMIT-20 and UNIX Kermit and runs with either. It is rather slow (500-900 effective baud on a 2400 baud line), but I suppose that's what you get for running FORTRAN. It has also been tested with Kermit-86 on a Corona (IBM-PC clone), IBM-PC and a Z100 and CPMKermit on a Z100. I had to modify version 1.19 of Kermit-86 to get it to assemble on the Z100 because some new code that was IBM-PC specific was not IFDEFed. The new version will be sent shortly. I am still trying to track down a problem with running the Cyber version and the IBM-PC version together. Looking at the modem port registers after it quits working leaves me to believe that interrupts are being turned off because the overrun flag is turned on and not being cleared. Perhaps Daphne could give me some pointers on tracing this problem down.

The source for the Cyber Version of Kermit is on utexas-11 as the file "/u0/cc/knutson/kerm170.for" (224280 bytes), you can use FTP as an anonymous login account with any password. The installation information is in the fortran source code. It will need mods for any Cyber site that tries to run it since I don't believe there are two Cyber sites anywhere that do ASCII I/O the same way.

There is also an nroff source for additions to the Kermit User's Guide that can be converted to runoff format very easily in file "/u0/cc/knutson/kerm170.nr". The following is the nroff output from kerm170.nr and can be found in file "/u0/cc/knutson/kerm170.doc"

Supplement to the "Kermit User's Guide"

3.x. CYBER 170/750 KERMIT

Kermit is written for the Cyber 170/750 in FTN5, CDC's FORTRAN 77 compiler. It mimics the DEC-20 style command parser in that a question mark followed by a carriage return will display the possible inputs for a given field. This version of Kermit must be watched since it has no timeout capabilities.

Kermit only uses local files. To send a file, it must first be

copied to a local file. Kermit may then be run to send it from the Cyber to another Kermit. To receive a file, Kermit will write it to a local file. This file must be copied to a permanent file before logging out if it is to be saved.

A Cyber filename is restricted to 7 alphanumeric characters with the first character being an alpha. Kermit will use the first 7 valid characters of any file being sent to the Cyber. If a file under that name already exists, an Error packet stating so will be sent back and Cyber Kermit will abort the transfer.

Kermit can only deal with "8 in 12" ASCII files. At the University of Texas, an "8 in 12" ASCII file is defined to be an 8 bit ASCII character stored in a 12 bit byte. There are 60 bits in a word so that is 5 bytes per word. A null character is represented as an octal 4000 and the CR/LF pair is mapped into the single character NEL (octal 205). This is the same character set that Cyber C uses. Unfortunately, CDC hasn't officially defined it's "8 in 12" character set yet, but it is likely that the end of line convention will be the similar to the current one which is one or more 0000 bytes in the low order portion of a word.

You can run Kermit interactively by typing repeated commands in response to its "Kermit-170>" prompt, or you can run it as a remote server. Here are the Kermit-170 commands:

EXIT Exit from Kermit-170. You may also stop Kermit-170 in the midst of a transfer or during server operations by aborting the program (CTRL-C or CTRL-G ABORT); the program traps them and puts yor terminal back to normal before halting.

HELP [<topic>]
Give Help. There's a general help text, plus seperate help texts for each Kermit-170 command.

QUIT Synonym for EXIT.

RECEIVE Receive a file or group of files from the other host. If the name in the header packet is not a legal Cyber file name, the first 7 legal characters will be used.

If the file already exits as a local file, Kermit will abort the transfer. If an error occurs during transfer, the file being received will be removed from the local file list to allow the transfer to be retried. You should escape back to your local Kermit after entering RECEIVE mode and give the SEND command.

SEND filename
Send a file to the other host. The name of the file is passed to the other host in a file header packet, so that the file can be stored there with the same name. You should escape back to your local Kermit and give the RECEIVE command. If you don't do this fast enough the "send-init" packet may arrive prematurely. To prevent this, use SET DELAY or hit the RETURN key on your microcomputer if it does not timeout.

SERVER Act as a server for another Kermit. Whatever options

were previously SET will be used. The server may be shut down from the local Kermit by using the BYE or FINISH commands, or by connecting back to the Cyber, and typing Control-C or Control-G ABORT.

SET Establish system-dependent parameters. You can examine their values with the SHOW command. Numeric values may be decimal, octal (postfixed with a B), or hexadecimal (postfixed by an H). The following may be SET:

DEBUG <option>
Show packet traffic explicitly. Options are:

ALL Set all debug options.

LOG-FILE filename
Log states and packets to the specified file. The default log-file is file KERM-LOG.

OFF Don't display debugging information (this is the default). If debugging was in effect, turn it off and close any log file.

PACKETS Display each incoming and outgoing packet (lengthy).

STATES Show kermit state transitions and packet numbers (brief).

DELAY <decimal-number>
How many seconds to wait before sending the first packet. This gives you time to "escape" back and issue a RECEIVE command.

DUPLEX <keyword>
Changes the method of echoing characters when being prompted for commands. The choices are FULL and HALF. Full means the Cyber will echo the characters you type. Half means the local systems echos them. Full is the default, and is used by most hosts.

INIT-RETRY <decimal-number>
Set the maximum number of retries allowed for the initial connection before giving up.

PARITY <keyword>
Allows you to adjust the parity on characters send by Kermit-170 to another system's requirements. The possibilities are NONE, EVEN, ODD, SPACE, or MARK. If NONE, then the 8th bit can be used for data when sending and receiving binary files. If other than NONE, the 8th bit on outgoing characters will be set to achieve the desired parity, and the 8th bit on incoming characters will be discarded. The default is NONE.

RECEIVE <parameter> <value>
These commands allow you to specify to the other

Kermit what the packets it sends should look like, or to inform this Kermit what to expect.

EOLCHR <octal-number>
The octal value of the ASCII character to be used as a line terminator or packets, if one is required by the other system. Carriage return (15B) is the default.

PACKLEN <decimal-number>
Maximum packet length to send, decimal number, between 20 and 94, 94 by default.

PADCHR <octal-number>
Character to use for padding. Default is NUL.

PADLEN <decimal-number>
How much padding to send before a packet. Default is no padding.

QUOTECHR <octal-number>
What printable character to use for quoting of control characters.

SYNCCHR <octal-number>
The control character that marks the beginning of the packet. Normally SOH (Control-A, ASCII 1). There should be no reason to change this.

TIMEOUT <decimal-number>
How many seconds the other Kermit should wait for a packet before asking for retransmission.

RETRY <decimal-number>
Set the maximum number of retries for a given packet before giving up.

SEND <parameter> <value>
These commands allow you to specify how outgoing packets should look, in case the other Kermit has nonstandard requirements.

EOLCHR <octal-number>
The octal value of the ASCII character to be used as a line terminator for packets, if one is required by the other system. Carriage return (15B) by default.

PACKLEN <decimal-number>
Maximum packet length to send, decimal number, between 20 and 94, 94 by default.

PADCHR <octal-number>
Character to use for padding. Default is NUL.

PADLEN <decimal-number>

How much padding to send before a packet. Default is no padding.

QUOTECHR <octal-number>
What printable character to use for quoting of control characters. The default is "#" (43B). there should be no reason to change this.

SYNCCHR <octal-number>
The control character that marks the beginning of the packet. Normally SOH (Control-A, ASCII 1). There should be no reason to change this.

TIMEOUT <decimal-number>
How many seconds the other Kermit should wants before being asked for retransmission. Unfortunately, the Cyber has no way of timing out so this parameter is ignored.

SHOW Display current SET parameters, version of Kermit-170, and other information.

STATUS Give statistics about the most recent file transfer.

Planned further enhancements to Kermit-170 being done at the University of Texas include:

- Addition of new server functions: directory, type, delete, etc.

- Host commands -- pass a command string to the system for execution. This may be a problem when implementing at other NOS sites.

- Two and Three character checksums.

- 8th-bit quoting, to allow transfer of binary files with systems that can't use the parity bit for data.

- Temporary escaping back to control command level.

- Badly needed performance tuning. Currently we have been experiencing 500-900 baud effective data throughput on a 2400 baud line.

Message 28 -- **********
27-Sep-83 09:58:40-EDT,923;000000000001
Return-Path: <knutson@utexas-11>
Received: from UTEXAS-11.ARPA by COLUMBIA-20.ARPA with TCP; Tue 27 Sep 83 09:58:37-ED
T
Date: 27 Sep 83 08:57:25 CDT (Tue)
From: Jim Knutson <knutson@utexas-11.ARPA>
Subject: Re: IBM PC Kermit giving up on reading data
Posted-Date: 27 Sep 83 08:57:25 CDT (Tue)
Message-Id: <8309271357.AA00147@UTEXAS-11.ARPA>
Received: by UTEXAS-11.ARPA (3.326/3.7)
        id AA00147; 27 Sep 83 08:57:25 CDT (Tue)
To: CC.FDC@COLUMBIA-20.ARPA

The version that I am implementing is written in FTN5 (FORTRAN). Most Cyber sites have this compiler. Unfortunately, Cybers were never meant to handle ASCII so each site generally has its own way of doing ASCII I/O. The I/O routines that I have written implement an I/O system in which the system I/O routines are pushed as far down as possible. This should allow other sites to use their ASCII I/O routines with as little hassle as possible.

Message 59 -- **********
19-Aug-83 22:23:52-EDT,786;000000000001
Return-Path: <@CUCS20:knutson@utexas-11>
Received: from CUCS20 by CU20B with DECnet; 19 Aug 83 22:23:50 EDT
Received: from UTEXAS-11.ARPA by COLUMBIA-20.ARPA with TCP; Fri 19 Aug 83 22:22:36-ED
T
Date: 19 Aug 83 21:22:33 CDT (Fri)
From: Jim Knutson <knutson@utexas-11.ARPA>
Subject: please add me to your list
Posted-Date: 19 Aug 83 21:22:33 CDT (Fri)
Message-Id: <8308200222.AA01298@UTEXAS-11.ARPA>
Received: by UTEXAS-11.ARPA (3.326/3.7)
        id AA01298; 19 Aug 83 21:22:33 CDT (Fri)
To: info-kermit-request@columbia-20.ARPA
Cc: knutson@utexas-11

I am in the process of installing kermit on our Cyber 170/750 using
Cyber C. I will be rewriting it later in FTN5.

Please add me to the kermit mailing list.

Many thanks,
Jim Knutson
knutson@utexas-11

SY.FDC

0069

BIN 7W

CU20B, TOPS-20 Monitor 5.1(5101)-2

PS:<KERMIT>170KERMIT.FOR.1

Created: 18-Oct-83 11:56:39AM Copies: 1
File format:ASCII Print mode:ASCII
Spacing:SINGLE Style:NORMAL /PRESERVE

SY.FDC

User:SY.F Account:I-X- Priority:10 Limit:273
Forms:NORMAL Paper-type:NORMAL Orientation:PORTRAIT Duplex:YES
Request queued 18-Oct-83 2:20:45PM Printed 19-Oct-83 4:58:53AM

```
      PROGRAM KERMIT

CCC   KERMIT - A CYBER FILE TRANSFER PROGRAM USING THE KERMIT PROTOCOL
C
C     THIS PROGRAM MAY NOT BE SOLD FOR PROFIT.
C
C     PROGRAM VERSION 1.0, KERMIT PROTOCOL VERSION 2
C     14 OCT 83
C
C     JIM KNUTSON
C     COMPUTATION CENTER ROOM 1
C     UNIVERISITY OF TEXAS
C     AUSTIN, TX   78712
C
C     APRPANET ADDRESS:  KNUTSON@UTEXAS-11
C
C     SPECIAL THANKS TO KING ABLES FOR HIS CONTRIBUTION.
C
C     BUILD SEQUENCE:
C        GET THE SOURCE (USING MODIFY OR UPDATE OR WHATEVER)
C        CHANGE THE APPROPRIATE PARAMETER DEFINITION FOR YOUR
C           SITE.
C        COMPILE USING:
C           FTN5,I=<SOURCE>,OPT=2.
C        LOAD IT WITH THE FOLLOWING SEGLOAD DIRECTIVES:
C          TREE     KERMIT-(SET,SERVER-(RECEIVE,SEND))
C SET      INCLUDE  SHOW,STATUS,HLPCMD,MATCH,SETVAL
C RECEIVE  INCLUDE  RINIT,RFILE,RDATA
C SEND     INCLUDE  SINIT,SFILE,SDATA,SEOF,SBREAK
C KERMIT   GLOBAL   PROTO,PACKET,DEBUG,MESSAGE,FILEIO,FILEIOC
C          END      KERMIT
C
C     IMPLEMENTATION NOTES:
C
C        THIS VERSION OF KERMIT SHOULD BE PORTABLE EXCEPT FOR FOUR
C        CASES.  THE DELAY FOR SENDING THE FIRST PACKET IS DONE
C        THROUGH A CALL TO A SUBROUTINE PP WHICH SNOOPS THE SYSTEM
C        CLOCK IN LOW CORE AND THEN USES RECALL TO WAIT FOR THE
C        PROPER AMOUNT OF TIME.  THERE IS PROBABLY GOING TO BE
C        A PROBLEM WITH OTHER SITES NOT HAVING A SUBROUTINE PP
C        OR THE SYSTEM CLOCK (# OF JIFS SINCE DEADSTART) IS NOT
C        AT THE SAME ADDRESS.  SEE THE SUBROUTINES DELAY() AND SNP().
C
C        THE SERVER KNOWS HOW TO LOGOUT ON UT2D BUT THE SUBROUTINE
C        TO DO THE SAME ON NOS SITES MAY NOT BE THE SAME OR MAY
C        NOT EXIST AT ALL.  SEE SUBROUTINE LOGOUT().
C
C        THE ASCII I/O IS ALSO PROBABLY NOT PORTABLE SINCE THE
C        UNIVERSITY OF TEXAS IS PROBABLY THE ONLY CYBER SITE USING
C        IOP AND CDC DOES NOT REALLY SUPPORT ASCII I/O YET.
C        ASCII I/O ON UT2D (UNIV. OF TEXAS OP. SYS.) IS DONE BY
C        SETTING BIT 2**43 IN FET+1.  THE ASCII CHARACTER SET THAT
C        IS USED IS "8 IN 12".  THIS IS 8 BITS OF AN ASCII CHARACTER
C        STORED IN A 12 BIT BYTE.  NULLS ARE REPRESENTED AS 4000B,
C        AND THE NEWLINE CHARACTER (NEL) IS 205B.  THIS IS SLIGHTLY
C        DIFFERENT FROM THE CDC END-OF-LINE WHICH IS 0000B IN THE
C        LOW ORDER BYTE OF THE WORD.  CURRENTLY, 0000B BYTES ARE IGNORED
C        SINCE NULLS ARE GUARDED.
C
C        THE DISPLAY CODE CHARACTER MAPPINGS FOR UT2D ARE DIFFERENT
```

```
C        THAN THE 64 AND 63 CHARACTER SET (SIGH).  THIS WILL AFFECT
C        THE TABLES IN THE SUBROUTINES ASC(), AS2DPC(), AND DPC2AS().
C
C        SITES THAT MODIFY KERMIT TO RUN ON THEIR SYSTEM SHOULD
C        MODIFY THE APPROPRIATE PARAMETER DEFINITION TO ALLOW
C        CONDITIONAL COMPILATION FOR THEIR SITE.  TRY TO BE
C        AS GENERAL AS YOU CAN WHEN MAKING MODS.
C
C        ****** ABOVE ALL SEND YOUR MODS BACK TO UT *******
C
C     KERMIT I/O CONSIDERATIONS:
C
C        KERMIT USES TWO MODES OF I/O.  IT DOES CODED I/O WHEN READING
C        FROM THE TERMINAL TO GET COMMANDS.  THIS CAUSES THE FRONT
C        END TO MAP THE CR/LF PAIR INTO A SINGLE NEL CHARACTER.
C        COMMAND EDITING (BACKSPACE AND CANCEL)  AND PARITY IS TAKEN
C        CAREOF BY THE FRONT-END.  BINARY I/O IS USED WHEN READING
C        AND SENDING PACKETS.  THIS ALLOWS KERMIT TO CONTROL THE
C        PARITY BIT.  OTHER NOS SITES MAY HAVE TO SET TRANSPARENT MODE
C        TO DO THIS.  BINARY I/O ALSO CAUSES NO COMMAND EDITING TO BE
C        DONE (BACKSPACES ARE TREATED AS REGULAR ASCII CHARACTERS) AND
C        CR/LF IS NOT MAPPED TO NEL.
C
C        KERMIT OPENS TWO FILES (STDIN AND STDOUT) AND CONNECTS
C        THEM TO THE TERMINAL FOR DOING THE ASCII I/O.  THIS WAS
C        DONE TO PREVENT PROBLEMS WITH TRYING TO BUFFER READS AND
C        WRITES TO THE SAME FILE.  WHEN READING/WRITING DISK FILES,
C        KERMIT WILL TRY TO BUFFER 8 DISK SECTORS (512 WORDS) WORTH
C        OF DATA PER READ/WRITE.  OF COURSE, THIS CAN'T BE DONE WITH
C        TERMINAL I/O SO A LOT OF BUFFER SPACE IS WASTED THERE.
C
C        THE ONLY IMPLEMENTATION DEPENDENT I/O ROUTINES SHOULD BE
C        FOPEN(), FLCOSE(), FFLUSH(), GETC(), AND STTY().
C        SEE THESE ROUTINES FOR MORE INFO ON WHAT THEY DO.
C
C     SUBROUTINE ORDERING:
C        MAIN PROGRAM AND INITIALISATION
C           KERMIT
C           BLKDAT.
C        KERMIT COMMAND SUBROUTINES
C           EXITPGM
C           HLPCMD
C           RCVFILE
C           SNDFILE
C           SET
C           SHOW
C           STATUS
C           DBUGCMD
C           SETPACK
C           DPLXCMD
C           PARCMD
C        COMMAND PARSING SUBROUTINES
C           MATCH
C           OUTTBL
C           SETVAL
C           CONFIRM
C        SERVER SUBROUTINES
C           SERVER
C           LOGOUT
C        KERMIT RECEIVE STATE PROTOCOL SUBROUTINES
```

```
C           RECEIVE
C           RINIT
C           RFILE
C           RDATA
C        KERMIT SEND STATE PROTOCOL SUBROUTINES
C           SEND
C           SINIT
C           SFILE
C           SDATA
C           SEOF
C           SBREAK
C        PACKET I/O SUBROUTINES
C           SNDPACK
C           RDPACK
C           BUFFILL
C           BUFEMP
C        STANDARD I/O SUBROUTINES
C           FOPEN
C           FCLOSE
C           FFLUSH
C           GETC
C           UNGETC
C           GETWORD
C           PUTC
C           PUTSTR
C           PUTINT
C           PUTDAY
C           PUTMNTH
C           FPRINTF
C           SPRINTF
C           DOPRNT
C           STTY
C           GTTY
C        UTILITY SUBROUTINES
C           AS2DPC
C           ASC
C           DPC2AS
C           CTOI
C           ITOS
C           GETEMSG
C           GETFILE
C           GETNOW
C           FILCHK
C           RDPARAM
C           REMOVE
C           STRCPY
C           SLEN
C           SNDPAR
C           SLEEP
C           DELAY
C           SNP
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMSIT
C
C     KERMIT SITE DEFINITIONS:
C
      PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
C-----FILE: C167,KERMDEF
```

```
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
      EXTERNAL EXITPGM
      CHARACTER*10 CMD(9)
      DATA CMD / 'EXIT', 'HELP', 'QUIT', 'RECEIVE', 'SEND', 'SERVER',
     +           'SET', 'SHOW', 'STATUS' /

C
C     OPEN THE I/O FILES
C
      IF (FOPEN('STDIN',1) .NE. STDIN) THEN
         CALL DISPLA(' CANNOT OPEN STANDARD INPUT')
         STOP
      ELSE IF (FOPEN('STDOUT',2) .NE. STDOUT) THEN
         CALL DISPLA(' CANNOT OPEN STANDARD OUTPUT')
         STOP
      ENDIF
```

```
C
C     MAKE SURE THINGS GET FIXED DURING ABORTS
C
      CALL RECOVR(EXITPGM,O"77",0)
C
C     PARSE AND EXECUTE ANY COMMANDS
C
5     CALL FPRINTF(STDOUT,'↑KERMIT-170>',0,0,0,0)
      CALL FFLUSH(STDOUT)
      CALL FFLUSH(STDIN)
      INDX = MATCH(CMD,9,.TRUE.)
      IF (INDX .EQ. ERROR .OR. INDX .EQ. 0) GO TO 5
      IF (INDX .EQ. EOF) CALL EXITPGM
      GO TO (10, 20, 10, 40, 50, 60, 70, 80, 90), INDX
C
C
C     THATS ALL FOLKS
C
10    CALL EXITPGM
C
C     GIVE SOME HELP
C
20    CALL HLPCMD
      GO TO 5
C
C     RECEIVE A FILE
C
40    CALL RCVFILE
      GO TO 5
C
C     SEND A FILE
C
50    CALL SNDFILE
      GO TO 5
C
C     ENTER SERVER MODE
C
60    CALL SERVER
      GO TO 5
C
C     SET SOME ATTRIBUTES
C
70    CALL SET
      GO TO 5
C
C     SHOW CURRENT SETTINGS
C
80    CALL SHOW
      GO TO 5
C
C     GIVE THE STATUS OF LAST TRANSFER
C
90    CALL STATUS
      GO TO 5
      END
      BLOCK DATA
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
```

```
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
```

```
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,EMSGCOM
C
C     MESSAGE COMMON BLOCK
C
      COMMON /MESSAGE/ VERSION, AMBIG, NOMATCH, FOLLOW, NODIGIT,MISSING,
     +   CONFMSG, NOTCONF,
     +   HLPASCH, HLPDLFP, HLPDBFN, HLPPLEN, HLPPADL, HLPIPRC, HLPPRTR,
     +   HLPTIMO, HLPSNFN
      CHARACTER*37 VERSION
      CHARACTER*15 AMBIG
      CHARACTER*38 NOMATCH
      CHARACTER*24 FOLLOW
      CHARACTER*53 NODIGIT
      CHARACTER*31 MISSING
      CHARACTER*33 CONFMSG
      CHARACTER*19 NOTCONF
      CHARACTER*42 HLPDLFP
      CHARACTER*37 HLPDBFN
      CHARACTER*24 HLPPLEN
      CHARACTER*34 HLPPADL
      CHARACTER*74 HLPASCH
      CHARACTER*29 HLPIPRC
      CHARACTER*21 HLPPRTR
      CHARACTER*43 HLPTIMO
      CHARACTER*19 HLPSNFN
C-----END OF FILE: EMSGCOM
      DATA IFD, OFD          / STDIN, STDOUT /
      DATA FFD               / 0 /
      DATA MAXRTRY, MAXRINI / MAXTRY, MAXINIT /
      DATA PACKNUM           / 0 /
      DATA STARTIM, ENDTIM   / 2*0 /
      DATA SCHCNT , RCHCNT   / 2*0 /
      DATA SCHOVRH, RCHOVRH / 2*0 /
      DATA STATE             / C /
      DATA DELAYFP           / 5 /
      DATA SYNC   , SNDSYNC / 2*SOH /
      DATA PACKSIZ, SPKSIZ  / 2*MAXPACK /
      DATA TIMEOUT, STIMOUT / 2*MYTIME  /
      DATA NPAD   , SPAD    / 2*MYPAD   /
      DATA PADCH  , SPADCH  / 2*MYPADCH /
      DATA EOLCH  , SPEOL   / 2*MYEOL   /
      DATA QUOTECH, SPQUOTE / 2*MYQUOTE /
      DATA QUOTE8 , S8QUOTE / 2*QUOT8CH /
      DATA CHKTYP , SCHKTYP / 2*MYCKTYP /
      DATA RPREFIX, SREPEAT / 2*PREFXCH /
      DATA DEBUG  , DEBUGFD / DBGOFF, 0 /
      DATA DEBUGFN           / 75, 69, 82, 77, 76, 79, 71, 0 /
```

```
C                               K   E   R   M   L   O   G
      DATA VERSION / '↑CYBER-170 ↑K↑E↑R↑M↑I↑T VERSION 1.0\N' /
      DATA AMBIG   / '?↑AMBIGUOUS - "' /
      DATA NOMATCH / '?↑DOES NOT MATCH SWITCH OR KEYWORD - "' /
      DATA FOLLOW  / '↑ONE OF THE FOLLOWING:\N' /
      DATA NODIGIT /
     +   '?↑INVALID, ↑FIRST NONSPACE CHARACTER IS NOT A DIGIT\N' /
      DATA MISSING / '?↑INVALID, ↑MISSING PARAMETER\N' /
      DATA CONFMSG / '↑CONFIRM WITH A CARRIAGE RETURN\N' /
      DATA NOTCONF / '?↑NOT CONFIRMED - "' /
      DATA HLPASCH / '↑DECIMAL, OCTAL (↑O), OR HEXIDECIMAL (↑H) CODE FOR
     + ↑A↑S↑C↑I↑I CHARACTER \N' /
      DATA HLPDLFP / '↑NUMBER OF SECONDS TO DELAY FIRST PACKET\N' /
      DATA HLPDBFN / '↑DEBUG OUTPUT LOGFILE SPECIFICATION\N' /
      DATA HLPPLEN / '↑MAXIMUM PACKET LENGTH\N' /
      DATA HLPPADL / '↑NUMBER OF PAD CHARACTERS TO USE\N' /
      DATA HLPIPRC / '↑INITIAL PACKET RETRY COUNT\N' /
      DATA HLPPRTR / '↑PACKET RETRY COUNT\N' /
      DATA HLPTIMO / '↑NUMBER OF SECONDS TO WAIT BEFORE TIMEOUT\N' /
      DATA HLPSNFN / '↑FILENAME TO SEND\N' /
      END
      SUBROUTINE EXITPGM

CCC   EXITPGM - EXIT THE PROGRAM
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
```

```
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
      CALL STTY('RAW',OFF)
      IF (SAVEDPX .NE. HALF) CALL STTY('DUPLEX',FULL)
      CALL FCLOSE(STDIN)
      CALL FCLOSE(STDOUT)
      IF (DEBUGFD.NE.0) CALL FCLOSE(DEBUGFD)
      STOP
      END
      SUBROUTINE HLPCMD

CCC   HLPCMD - PROCESS THE HELP COMMAND.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
```

```
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
      PARAMETER (TSIZE=10)
      CHARACTER*10 HLPTYP(TSIZE)
      LOGICAL CONFIRM
      DATA HLPTYP / 'EXIT', 'HELP', 'KERMIT', 'QUIT', 'RECEIVE', 'SEND',
     +              'SERVER', 'SET', 'SHOW', 'STATUS' /

      INDX = MATCH(HLPTYP,TSIZE,.TRUE.)
      IF (INDX .EQ. EOF .OR. INDX .EQ. ERROR) RETURN
      IF (INDX .EQ. 0) GO TO 30
      IF (.NOT. CONFIRM(STDIN)) RETURN
      GO TO (10, 20, 30, 10, 40, 50, 60, 70, 80, 90), INDX
C
C     HELP EXIT
C
10    CALL FPRINTF(STDOUT,'↑EXIT FROM ↑KERMIT-170\N')
      RETURN
C
C     HELP HELP
C
20    CALL FPRINTF(STDOUT,'\N↑H↑E↑L↑P [TOPIC]\N\N↑TYPING ↑H↑E↑L↑P ALONE
     +PRINTS A BRIEF SUMMARY OF ↑KERMIT-170 AND ITS COMMANDS.\N↑YOU CAN
     +ALSO TYPE\N\N   ↑H↑E↑L↑P COMMAND\N\NFOR ANY ↑KERMIT-170 COMMAND, E
     +.G. "HELP SEND", TO GET MORE DETAILED INFORMATION\NABOUT A SPECIFI
     +C COMMAND.  ↑TYPE\N\N   ↑H↑E↑L↑P ?\N\NTO SEE A LIST OF ALL THE AVA
     +ILABLE HELP COMMANDS, OR CONSULT THE ↑KERMIT ↑USERS\N↑GUIDE.\N\N')
      RETURN
C
C     HELP KERMIT
C
30    CALL FPRINTF(STDOUT,'\N↑KERMIT IS A FILE TRANSFER PROTOCOL FOR USE
     + OVER AN ASYNCHRONOUS SERIAL\NTELECOMMUNICATION LINE.  ↑FILES ARE
     +BROKEN UP INTO "PACKETS" WITH CHECKSUMS AND\NOTHER CONTROL INFORMA
     +TION TO ENSURE (WITH HIGH PROBABILITY) ERROR-FREE AND\NCOMPLETE TR
     +ANSMISSION.\N\N↑KERMIT-170 IS THE IMPLEMENTATION FOR THE ↑CYBER 17
     +0/750 AND IS\NRUN "REMOTELY" FROM ANOTHER COMPUTER (E.G. A MICROCO
     +OPUTER).\N\N↑YOU CAN RUN ↑KERMIT INTERACTIVELY BY TYPING REPEATED
     +COMMANDS IN RESPONSE TO\NITS "↑KERMIT-170>" PROMPT, OR YOU CAN RUN
     + IT AS A REMOTE SERVER.\N\N↑KERMIT-170 COMMAND SUMMARY -- OPTIONAL
     + PARTS ARE IN [BRACKETS]:\N\N')
      CALL FPRINTF(STDOUT,'* ↑FOR EXCHANGING FILES:          ↑S↑E↑N↑D FIL
     +E\N')
      CALL FPRINTF(STDOUT,'                                  ↑R↑E↑C↑E↑I↑V↑
     +E\N\N')
      CALL FPRINTF(STDOUT,'* ↑FOR ACTING AS A SERVER:        ↑S↑E↑R↑V↑E↑R
```

```
     +\N\N')
      CALL FPRINTF(STDOUT,'* ↑SETTING NONSTANDARD TRANSMISSION AND FILE
     +PARAMETERS:\N        ↑S↑E↑T ↑D↑E↑B↑U↑G, ↑D↑E↑L↑A↑Y, ↑D↑U↑P↑L↑E↑X,
     +↑P↑A↑R↑I↑T↑Y, ↑I↑N↑I↑T-↑R↑E↑T↑R↑Y, ↑R↑E↑T↑R↑Y\N')
      CALL FPRINTF(STDOUT,'        ↑S↑E↑T ↑S↑E↑N↑D (OR ↑R↑E↑C↑E↑I↑V↑E) ↑
     +E↑O↑L↑C↑H↑R, ↑P↑A↑C↑K↑L↑E↑N, ↑P↑A↑D↑C↑H↑R, ↑P↑A↑D↑L↑E↑N, ↑Q↑U↑O↑T↑
     +E↑C↑H↑R,\N                ↑S↑Y↑N↑C↑C↑H↑R, ↑T↑I↑M↑E↑O↑U↑T\N\N')
      CALL FPRINTF(STDOUT,'* ↑GETTING INFORMATION:          ↑H↑E↑L↑P [TO
     +PIC], ↑S↑T↑A↑T↑U↑S, ↑S↑H↑O↑W\N\N')
      CALL FPRINTF(STDOUT,'* ↑LEAVING THE PROGRAM:          ↑E↑X↑I↑T, ↑Q
     +↑U↑I↑T\N\N')
      CALL FPRINTF(STDOUT,'↑FOR FURTHER INFORMATION, TYPE "HELP" FOR ANY
     + OF THE ABOVE, E.G. "HELP SET",\NOR SEE THE "↑KERMIT ↑USERS ↑GUIDE
     +" AND THE "↑KERMIT ↑PROTOCOL ↑MANUAL" FOR COMPLETE\NDETAILS.\N\N')
      RETURN
C
C     HELP RECEIVE
C
40    CALL FPRINTF(STDOUT,'\N↑R↑E↑C↑E↑I↑V↑E\N\N↑RECEIVE A FILE OR GROUP
     +OF FILES FROM THE OTHER HOST.  ↑IF THE NAME IN THE\N')
      CALL FPRINTF(STDOUT,'HEADER PACKET IS NOT A LEGAL ↑CYBER FILE NAME
     +, THE FIRST 7 LEGAL CHARACTERS\N')
      CALL FPRINTF(STDOUT,'WILL BE USED.\N\N↑IF THE FILE ALREADY EXITS A
     +S A LOCAL FILE, ↑KERMIT WILL ABORT THE TRANSFER.\N')
      CALL FPRINTF(STDOUT,'↑IF AN ERROR OCCURS DURING TRANSFER, THE FILE
     + BEING RECEIVED WILL BE\NREMOVED FROM THE LOCAL FILE LIST TO ALLOW
     + THE TRANSFER TO BE RETRIED.\N')
      CALL FPRINTF(STDOUT,'↑YOU SHOULD ESCAPE BACK TO YOUR LOCAL ↑KERMIT
     + AFTER ENTERING ↑R↑E↑C↑E↑I↑V↑E\NMODE AND GIVE THE ↑S↑E↑N↑D COMMAND
     +.\N\N')
      RETURN
C
C     HELP SEND
C
50    CALL FPRINTF(STDOUT,'\N↑S↑E↑N↑D FILENAME\N\N')
      CALL FPRINTF(STDOUT,'↑SEND A FILE TO THE OTHER HOST.  ↑THE NAME OF
     + THE FILE IS PASSED\NTO THE OTHER HOST IN A FILE HEADER PACKET, SO
     + THAT THE FILE CAN BE\NSTORED THERE WITH THE SAME NAME.\N\N')
      CALL FPRINTF(STDOUT,'↑YOU SHOULD ESCAPE BACK TO YOUR LOCAL ↑KERMIT
     + AND GIVE THE ↑R↑E↑C↑E↑I↑V↑E\NCOMMAND.  ↑IF YOU DON''T DO THIS FAS
     +T ENOUGH THE "SEND-INIT" PACKET MAY\NARRIVE PREMATURELY.  ↑TO PREV
     +ENT THIS, USE ↑S↑E↑T ↑D↑E↑L↑A↑Y OR HIT THE ↑R↑E↑T↑U↑R↑N KEY\NON YO
     +UR MICROCOMPUTER IF IT DOES NOT TIMEOUT.\N\N')
      RETURN
C
C     HELP SERVER
C
60    CALL FPRINTF(STDOUT,'\N↑S↑E↑R↑V↑E↑R\N\N')
      CALL FPRINTF(STDOUT,'↑ACT AS A SERVER FOR ANOTHER ↑KERMIT.  ↑TAKE
     +ALL FURTHER COMMANDS ONLY FROM\NTHE OTHER ↑KERMIT.  ↑AFTER ISSUING
     + THIS COMMAND, ESCAPE BACK TO YOUR LOCAL\NSYSTEM AND ISSUE ↑S↑E↑N↑
     +D, ↑R↑E↑C↑E↑I↑V↑E OR ↑G↑E↑T, ↑B↑Y↑E, OR OTHER SERVER-ORIENTED\NCOM
     +MANDS FROM THERE.  ↑IF YOUR LOCAL ↑KERMIT DOES NOT HAVE A ↑B↑Y↑E C
     +OMMAND,\NIT DOES NOT HAVE THE FULL ABILITY TO COMMUNICATE WITH A ↑
     +KERMIT SERVER (IN\NWHICH CASE YOU CAN ONLY USE THE ↑S↑E↑N↑D COMMAN
     +D).  ↑IF YOUR LOCAL ↑KERMIT\NDOES HAVE A ↑B↑Y↑E COMMAND, USE IT TO
     + SHUT DOWN AND LOG OUT THE ↑KERMIT\NSERVER WHEN YOU ARE DONE WITH
     +IT; OTHERWISE, CONNECT BACK TO THE ↑CYBER, TYPE\NSEVERAL ↑CONTROL-
     +↑C''S TO STOP THE SERVER, AND LOGOUT.\N\N')
      RETURN
```

```
C
C     HELP SET
C
70    CALL FPRINTF(STDOUT,'\N↑S↑E↑T\N')
      CALL FPRINTF(STDOUT,'  ↑ESTABLISH SYSTEM-DEPENDENT PARAMETERS.  ↑Y
     +OU CAN EXAMINE THEIR VALUES WITH THE\N  ↑S↑H↑O↑W COMMAND.  ↑NUMERI
     +C VALUES MAY BE DECIMAL, OCTAL (POSTFIXED WITH A ↑B),\N  OR HEXADE
     +CIMAL (POSTFIXED BY AN ↑H).  ↑THE FOLLOWING MAY BE ↑S↑E↑T:\N\N')
      CALL FPRINTF(STDOUT,' ↑D↑E↑B↑U↑G OPTIONS\N   ↑SHOW PACKET TRAFFIC
     +EXPLICITLY.  ↑OPTIONS ARE:\N')
      CALL FPRINTF(STDOUT,'   ↑A↑L↑L      ↑SET ALL DEBUG OPTIONS.\N')
      CALL FPRINTF(STDOUT,'   ↑L↑O↑G-↑F↑I↑L↑E ↑LOG STATES AND PACKETS TO
     + THE SPECIFIED FILE.  ↑THE DEFAULT\N            LOG-FILE IS FILE ↑
     +K↑E↑R↑M↑L↑O↑G.\N')
      CALL FPRINTF(STDOUT,'   ↑O↑F↑F      ↑DON''T DISPLAY DEBUGGING INFO
     +RMATION (THIS IS THE DEFAULT).  ↑IF\N            DEBUGGING WAS IN
     +EFFECT, TURN IT OFF AND CLOSE ANY LOG FILE.\N')
      CALL FPRINTF(STDOUT,'   ↑P↑A↑C↑K↑E↑T↑S  ↑DISPLAY EACH INCOMING AND
     + OUTGOING PACKET (LENGTHY).\N')
      CALL FPRINTF(STDOUT,'   ↑S↑T↑A↑T↑E↑S   ↑SHOW KERMIT STATE TRANSITI
     +ONS AND PACKET NUMBERS (BRIEF).\N\N')
      CALL FPRINTF(STDOUT,' ↑D↑E↑L↑A↑Y DECIMAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'   ↑HOW MANY SECONDS TO WAIT BEFORE SENDING T
     +HE FIRST PACKET.  ↑THIS GIVES YOU\N   TIME TO "ESCAPE" BACK AND IS
     +SUE A ↑R↑E↑C↑E↑I↑V↑E COMMAND.\N\N')
      CALL FPRINTF(STDOUT,' ↑D↑U↑P↑L↑E↑X KEYWORD\N')
      CALL FPRINTF(STDOUT,'   ↑CHANGES THE METHOD OF ECHOING CHARACTERS
     +WHEN BEING PROMPTED FOR COMMANDS.\N   ↑THE CHOICES ARE ↑F↑U↑L↑L AN
     +D ↑H↑A↑L↑F.  ↑FULL MEANS THE ↑CYBER WILL ECHO THE\N   CHARACTERS Y
     +OU TYPE.  ↑HALF MEANS THE LOCAL SYSTEMS ECHOS THEM.  ↑FULL IS\N
     +THE DEFAULT, AND IS USED BY MOST HOSTS.\N\N')
      CALL FPRINTF(STDOUT,' ↑I↑N↑I↑T-↑R↑E↑T↑R↑Y DECIMAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'   ↑SET THE MAXIMUM NUMBER OF RETRIES ALLOWED
     + FOR THE INITIAL CONNECTION\N   BEFORE GIVING UP.\N\N')
      CALL FPRINTF(STDOUT,' ↑P↑A↑R↑I↑T↑Y KEYWORD\N')
      CALL FPRINTF(STDOUT,'   ↑IF THE OTHER COMPUTER IS USING PARITY ON
     +THE COMMUNICATION LINE, YOU MUST\N   INFORM ↑KERMIT-170, SO IT CAN
     + SEND THE DESIRED PARITY ON OUTGOING CHARACTERS,\N   AND STRIP IT
     +FROM INCOMING ONES.\N')
      CALL FPRINTF(STDOUT,'\N   ↑THIS MUST BE SET IN ↑KERMIT AND THE FRO
     +NT-END.  ↑SEE A SYSTEM MANUAL FOR\N   SETTING PARITY IN THE FRONT-
     +END.\N')
      CALL FPRINTF(STDOUT,'\N   ↑CHOICES ARE ↑N↑O↑N↑E (THE DEFAULT), ↑E↑
     +V↑E↑N, ↑O↑D↑D, ↑M↑A↑R↑K, AND ↑S↑P↑A↑C↑E.\N   ↑N↑O↑N↑E MEANS NO PAR
     +ITY PROCESSING IS DONE, AND THE 8TH BIT OF EACH CHARACTER\N   CAN
     +BE USED FOR DATA WHEN TRANSMITTING BINARY FILES.\N\N')
      CALL FPRINTF(STDOUT,' ↑R↑E↑T↑R↑Y DECIMAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'   ↑SET THE MAXIMUM NUMBER OF RETRIES ALLOWED
     + FOR SENDING A PARTICULAR PACKET.\N\N')
      CALL FPRINTF(STDOUT,' ↑S↑E↑N↑D PARAMETER\N   ↑PARAMETERS FOR OUTGO
     +ING PACKETS AS FOLLOWS:\N\N')
      CALL FPRINTF(STDOUT,'   ↑E↑O↑L↑C↑H↑R OCTAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑THE OCTAL VALUE OF THE ↑A↑S↑C↑I↑I CHARA
     +CTER TO BE USED AS A LINE TERMINATOR\N     FOR PACKETS, IF ONE IS
     +REQUIRED BY THE OTHER SYSTEM.  ↑CARRIAGE\N     RETURN (15↑B) BY DE
     +FAULT.\N\N')
      CALL FPRINTF(STDOUT,'   ↑P↑A↑C↑K↑L↑E↑N DECIMAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑MAXIMUM PACKET LENGTH TO SEND, DECIMAL
     +NUMBER, BETWEEN 20 AND 94,\N     94 BY DEFAULT.\N\N')
      CALL FPRINTF(STDOUT,'   ↑P↑A↑D↑C↑H↑R OCTAL-NUMBER\N')
```

```
      CALL FPRINTF(STDOUT,'     ↑CHARACTER TO USE FOR PADDING.  ↑DEFAULT
     + IS ↑N↑U↑L.\N\N')
      CALL FPRINTF(STDOUT,'   ↑P↑A↑D↑L↑E↑N DECIMAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑HOW MUCH PADDING TO SEND BEFORE A PACKE
     +T.  ↑DEFAULT IS NO PADDING.\N\N')
      CALL FPRINTF(STDOUT,'   ↑Q↑U↑O↑T↑E↑C↑H↑R OCTAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑WHAT PRINTABLE CHARACTER TO USE FOR QUO
     +TING OF CONTROL CHARACTERS.\N     ↑THE DEFAULT IS "#" (43↑B).  THE
     +RE SHOULD BE NO REASON TO CHANGE THIS.\N\N')
      CALL FPRINTF(STDOUT,'   ↑S↑Y↑N↑C↑C↑H↑R OCTAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑THE CONTROL CHARACTER THAT MARKS THE BE
     +GINNING OF THE PACKET.  ↑NORMALLY\N     ↑S↑O↑H (↑CONTROL-↑A, ↑A↑S↑
     +C↑I↑I 1).  ↑THERE SHOULD BE NO REASON TO CHANGE THIS.\N\N')
      CALL FPRINTF(STDOUT,'   ↑T↑I↑M↑E↑O↑U↑T DECIMAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑HOW MANY SECONDS THE OTHER ↑KERMIT WANT
     +S BEFORE BEING ASKED\N     FOR RETRANSMISSION.  ↑UNFORTUNATELY, TH
     +E ↑CYBER HAS NO WAY OF TIMING\N     OUT SO THIS PARAMETER IS IGNOR
     +ED.\N\N')
      CALL FPRINTF(STDOUT,' ↑R↑E↑C↑E↑I↑V↑E PARAMETER\N   ↑PARAMETERS TO
     +REQUEST OR EXPECT FOR INCOMING PACKETS, AS FOLLOWS:\N\N')
      CALL FPRINTF(STDOUT,'   ↑E↑O↑L↑C↑H↑R OCTAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑THE OCTAL VALUE OF THE ↑A↑S↑C↑I↑I CHARA
     +CTER TO BE USED AS A LINE TERMINATOR\N     FOR PACKETS, IF ONE IS
     +REQUIRED BY THE OTHER SYSTEM.  ↑CARRIAGE\N     RETURN (15↑B) BY DE
     +FAULT.\N\N')
      CALL FPRINTF(STDOUT,'   ↑P↑A↑C↑K↑L↑E↑N DECIMAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑MAXIMUM PACKET LENGTH TO SEND, DECIMAL
     +NUMBER, BETWEEN 20 AND 94,\N     94 BY DEFAULT.\N\N')
      CALL FPRINTF(STDOUT,'   ↑P↑A↑D↑C↑H↑R OCTAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑CHARACTER TO USE FOR PADDING.  ↑DEFAULT
     + IS ↑N↑U↑L.\N\N')
      CALL FPRINTF(STDOUT,'   ↑P↑A↑D↑L↑E↑N DECIMAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑HOW MUCH PADDING TO SEND BEFORE A PACKE
     +T.  ↑DEFAULT IS NO PADDING.\N\N')
      CALL FPRINTF(STDOUT,'   ↑Q↑U↑O↑T↑E↑C↑H↑R OCTAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑WHAT PRINTABLE CHARACTER TO USE FOR QUO
     +TING OF CONTROL CHARACTERS.\N     ↑THE DEFAULT IS "#" (43↑B).  THE
     +RE SHOULD BE NO REASON TO CHANGE THIS.\N\N')
      CALL FPRINTF(STDOUT,'   ↑S↑Y↑N↑C↑C↑H↑R OCTAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑THE CONTROL CHARACTER THAT MARKS THE BE
     +GINNING OF THE PACKET.  ↑NORMALLY\N     ↑S↑O↑H (↑CONTROL-↑A, ↑A↑S↑
     +C↑I↑I 1).  ↑THERE SHOULD BE NO REASON TO CHANGE THIS.\N\N')
      CALL FPRINTF(STDOUT,'   ↑T↑I↑M↑E↑O↑U↑T DECIMAL-NUMBER\N')
      CALL FPRINTF(STDOUT,'     ↑HOW MANY SECONDS THE OTHER ↑KERMIT SHOU
     +LD WAIT FOR A PACKET BEFORE\N     ASKING FOR RETRANSMISSION.\N\N')
      RETURN
C
C     HELP SHOW
C
80    CALL FPRINTF(STDOUT,'↑DISPLAY CURRENT ↑S↑E↑T PARAMETERS, VERSION O
     +F ↑KERMIT-170, AND OTHER INFO.\N')
      RETURN
C
C     HELP STATUS
C
90    CALL FPRINTF(STDOUT,'↑GIVE STATISTICS ABOUT THE MOST RECENT FILE T
     +RANSFER.\N')
      RETURN
      END
      SUBROUTINE RCVFILE
```

```
CCC   RCVFILE - TOP LEVEL SUBROUTINE TO START RECEIVE STATE.
C
      IMPLICIT INTEGER (A-Z)
      LOGICAL CONFIRM
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C
C     CONFIRM THE COMMAND
C
```

```
      IF (.NOT. CONFIRM(STDIN)) RETURN
      CALL STTY('BINARY',ON)
      SAVEDPX = GTTY('DUPLEX')
      CALL STTY('DUPLEX',HALF)
      IF (RECEIVE(R) .EQ. OK) THEN
         CALL FPRINTF(STDOUT,'↑RECEIVE COMPLETE.\N',0,0,0,0)
      ELSE
         CALL FPRINTF(STDOUT,'↑RECEIVE FAILED.\N',0,0,0,0)
      ENDIF
      CALL STTY('BINARY',OFF)
      IF (SAVEDPX .NE. HALF) CALL STTY('DUPLEX',FULL)
      RETURN
      END
      SUBROUTINE SNDFILE

CCC   SNDFILE - SEND A FILE TO OTHER KERMIT.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
```

```
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,EMSGCOM
C
C     MESSAGE COMMON BLOCK
C
      COMMON /MESSAGE/ VERSION, AMBIG, NOMATCH, FOLLOW, NODIGIT,MISSING,
     +   CONFMSG, NOTCONF,
     +   HLPASCH, HLPDLFP, HLPDBFN, HLPPLEN, HLPPADL, HLPIPRC, HLPPRTR,
     +   HLPTIMO, HLPSNFN
      CHARACTER*37 VERSION
      CHARACTER*15 AMBIG
      CHARACTER*38 NOMATCH
      CHARACTER*24 FOLLOW
      CHARACTER*53 NODIGIT
      CHARACTER*31 MISSING
      CHARACTER*33 CONFMSG
      CHARACTER*19 NOTCONF
      CHARACTER*42 HLPDLFP
      CHARACTER*37 HLPDBFN
      CHARACTER*24 HLPPLEN
      CHARACTER*34 HLPPADL
      CHARACTER*74 HLPASCH
      CHARACTER*29 HLPIPRC
      CHARACTER*21 HLPPRTR
      CHARACTER*43 HLPTIMO
      CHARACTER*19 HLPSNFN
C-----END OF FILE: EMSGCOM
      LOGICAL LOCAL
      CHARACTER*10 FNAME
C
C     PICK UP THE FILE NAME AND SAVE IT FOR OPENING LATER
C
      CALL SETVAL(FILESTR,'S',IRET,7,0,0,HLPSNFN,.TRUE.)
      IF (IRET .EQ. ERROR) RETURN
C
C     CHECK TO MAKE SURE IT'S THERE TO SEND
C
      CALL AS2DPC(FILESTR,FNAME)
      INQUIRE(FILE=FNAME,EXIST=LOCAL)
      IF (.NOT. LOCAL) THEN
         CALL FPRINTF(STDOUT,'?↑FILE %S IS NOT LOCAL.\N',FILESTR,0,0,0)
         RETURN
      ENDIF
C
```

```
C     DELAY THE FIRST PACKET
C
      IF (DELAYFP .GT. 0) CALL SLEEP(DELAYFP)
      CALL STTY('BINARY',ON)
      SAVEDPX = GTTY('DUPLEX')
      CALL STTY('DUPLEX',HALF)
C
C     START SENDING PACKETS
C
      PACKNUM = 0
      IF (SEND() .EQ. OK) THEN
         CALL FPRINTF(STDOUT,'↑SEND COMPLETE.\N',0,0,0,0)
      ELSE
         CALL FPRINTF(STDOUT,'↑SEND FAILED.\N',0,0,0,0)
      ENDIF
      CALL STTY('BINARY',OFF)
      IF (SAVEDPX .NE. HALF) CALL STTY('DUPLEX',FULL)
      RETURN
      END
      SUBROUTINE SET

CCC   SET - SET SOME ATTRIBUTES.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
```

```
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
C-----FILE: C167,EMSGCOM
C
C     MESSAGE COMMON BLOCK
C
      COMMON /MESSAGE/ VERSION, AMBIG, NOMATCH, FOLLOW, NODIGIT,MISSING,
     +   CONFMSG, NOTCONF,
     +   HLPASCH, HLPDLFP, HLPDBFN, HLPPLEN, HLPPADL, HLPIPRC, HLPPRTR,
     +   HLPTIMO, HLPSNFN
      CHARACTER*37 VERSION
      CHARACTER*15 AMBIG
      CHARACTER*38 NOMATCH
      CHARACTER*24 FOLLOW
      CHARACTER*53 NODIGIT
      CHARACTER*31 MISSING
      CHARACTER*33 CONFMSG
      CHARACTER*19 NOTCONF
      CHARACTER*42 HLPDLFP
      CHARACTER*37 HLPDBFN
      CHARACTER*24 HLPPLEN
      CHARACTER*34 HLPPADL
      CHARACTER*74 HLPASCH
      CHARACTER*29 HLPIPRC
      CHARACTER*21 HLPPRTR
      CHARACTER*43 HLPTIMO
      CHARACTER*19 HLPSNFN
C-----END OF FILE: EMSGCOM
      PARAMETER (TSIZE=8)
```

```
      CHARACTER*10 SETTYP(TSIZE)
      DATA SETTYP / 'DEBUG', 'DELAY', 'DUPLEX', 'INIT-RETRY', 'PARITY',
     +    'RECEIVE', 'RETRY', 'SEND' /

      INDX = MATCH(SETTYP,TSIZE,.FALSE.)
      IF (INDX .LE. 0) RETURN
      GO TO (10, 20, 30, 40, 50, 60, 70, 80), INDX
C
C     SET DEBUGGING MODES
C
10    CALL DBUGCMD
      RETURN
C
C     SET FIRST PACKET DELAY
C
20    CALL SETVAL(DELAYFP,'I',0,60,0,60,HLPDLFP,.TRUE.)
      RETURN
C
C     SET THE DUPLEX
C
30    CALL DPLXCMD
      RETURN
C
C     SET INTIAL PACKET RETRY COUNT
C
40    CALL SETVAL(MAXRINI,'I',1,50,1,50,HLPIPRC,.TRUE.)
      RETURN
C
C     SET PARITY
C
50    CALL PARCMD
      RETURN
C
C     SET RECEIVE PACKET ATTRIBUTES
C
60    CALL SETPACK(PACKSIZ)
      RETURN
C
C     SET PACKET RETRY COUNT
C
70    CALL SETVAL(MAXRTRY,'I',1,50,1,50,HLPPRTR,.TRUE.)
      RETURN
C
C     SET SEND PACKET ATTRIBUTES
C
80    CALL SETPACK(SPKSIZ)
      RETURN
      END
      SUBROUTINE SHOW

CCC   SHOW THE CURRENT PROGRAM SETTINGS
C
      IMPLICIT INTEGER (A-Z)
      LOGICAL CONFIRM
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
```

```
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
```

```
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,EMSGCOM
C
C     MESSAGE COMMON BLOCK
C
      COMMON /MESSAGE/ VERSION, AMBIG, NOMATCH, FOLLOW, NODIGIT,MISSING,
     +   CONFMSG, NOTCONF,
     +   HLPASCH, HLPDLFP, HLPDBFN, HLPPLEN, HLPPADL, HLPIPRC, HLPPRTR,
     +   HLPTIMO, HLPSNFN
      CHARACTER*37 VERSION
      CHARACTER*15 AMBIG
      CHARACTER*38 NOMATCH
      CHARACTER*24 FOLLOW
      CHARACTER*53 NODIGIT
      CHARACTER*31 MISSING
      CHARACTER*33 CONFMSG
      CHARACTER*19 NOTCONF
      CHARACTER*42 HLPDLFP
      CHARACTER*37 HLPDBFN
      CHARACTER*24 HLPPLEN
      CHARACTER*34 HLPPADL
      CHARACTER*74 HLPASCH
      CHARACTER*29 HLPIPRC
      CHARACTER*21 HLPPRTR
      CHARACTER*43 HLPTIMO
      CHARACTER*19 HLPSNFN
C-----END OF FILE: EMSGCOM
C
C     CONFIRM THE COMMAND
C
      IF (.NOT. CONFIRM(STDIN)) RETURN
      CALL FPRINTF(STDOUT,VERSION,0,0,0,0)
C
C     DISPLAY THE CURRENT DATE AND TIME
C
      CALL GETNOW(MM,DD,YY,HR,MIN,SEC)
      CALL PUTDAY(STDOUT,MM,DD,YY)
      CALL FPRINTF(STDOUT,', ',0,0,0,0)
      CALL PUTMNTH(STDOUT,MM)
      CALL FPRINTF(STDOUT,' %D, %D ',DD,YY,0,0)
      IF (HR .LT. 10) CALL PUTC(ASC('0'),STDOUT)
      CALL FPRINTF(STDOUT,'%D:',HR,0,0,0)
      IF (MIN .LT. 10) CALL PUTC(ASC('0'),STDOUT)
      CALL FPRINTF(STDOUT,'%D:',MIN,0,0,0)
      IF (SEC .LT. 10) CALL PUTC(ASC('0'),STDOUT)
      CALL FPRINTF(STDOUT,'%D\N\N',SEC,0,0,0)
C
C     DISPLAY KNOWN PARITY
C
      CALL FPRINTF(STDOUT,'  ↑PARITY:    ',0,0,0,0)
      PARITY = GTTY('PARITY')
```

```
      IF (PARITY .EQ. NONE) THEN
         CALL FPRINTF(STDOUT,'↑NONE\N',0,0,0,0)
      ELSE IF (PARITY .EQ. EVEN) THEN
         CALL FPRINTF(STDOUT,'↑EVEN\N',0,0,0,0)
      ELSE IF (PARITY .EQ. ODD) THEN
         CALL FPRINTF(STDOUT,'↑ODD\N',0,0,0,0)
      ELSE IF (PARITY .EQ. MARK) THEN
         CALL FPRINTF(STDOUT,'↑MARK\N',0,0,0,0)
      ELSE IF (PARITY .EQ. SPACE) THEN
         CALL FPRINTF(STDOUT,'↑SPACE\N',0,0,0,0)
      ELSE
         CALL FPRINTF(STDOUT,'↑UNKNOWN\N',0,0,0,0)
      ENDIF
C
C     DISPLAY THE CURRENT DUPLEX
C
      CALL FPRINTF(STDOUT,'  ↑DUPLEX:    ',0,0,0,0)
      DUPLEX = GTTY('DUPLEX')
      IF (DUPLEX .EQ. FULL) THEN
         CALL FPRINTF(STDOUT,'↑FULL\N',0,0,0,0)
      ELSE IF (DUPLEX .EQ. HALF) THEN
         CALL FPRINTF(STDOUT,'↑HALF\N',0,0,0,0)
      ELSE
         CALL FPRINTF(STDOUT,'↑UNKNOWN\N',0,0,0,0)
      ENDIF
C
C     DISPLAY CURRENT DEBUG MODES
C
      CALL FPRINTF(STDOUT,'  ↑DEBUGGING: ',0,0,0,0)
      IF ((DEBUG.AND.DBGSTAT).NE.0) CALL FPRINTF(STDOUT,'↑STATES ',
     +   0,0,0,0)
      IF ((DEBUG.AND.DBGPACK).NE.0) CALL FPRINTF(STDOUT,'↑PACKETS',
     +   0,0,0,0)
      IF (DEBUG.EQ.DBGOFF) CALL FPRINTF(STDOUT,'↑OFF',0,0,0,0)
      CALL PUTC(NEL,STDOUT)
      IF (DEBUG .NE. DBGOFF) THEN
         CALL FPRINTF(STDOUT,'   ↑LOG FILE: %S\N',DEBUGFN,0,0,0)
      ENDIF
C
C     DISPLAY PACKET SETTINGS
C
      CALL FPRINTF(STDOUT,'\N↑PACKET ↑PARAMETERS\N',0,0,0,0)
      CALL FPRINTF(STDOUT,
     +   '                    ↑RECEIVE   ↑SEND\N',0,0,0,0)
      CALL FPRINTF(STDOUT,'  ↑SIZE:             %D        %D\N',
     +   PACKSIZ,SPKSIZ,0,0)
      CALL FPRINTF(STDOUT,'  ↑TIMEOUT:          %D        %D\N',
     +   TIMEOUT,STIMOUT,0,0)
      CALL FPRINTF(STDOUT,'  ↑PADDING:          %D',NPAD,0,0,0)
      IF (NPAD .LT. 10) CALL PUTC(BLANK,STDOUT)
      CALL FPRINTF(STDOUT,'        %D\N',SPAD,0,0,0)
      CALL FPRINTF(STDOUT,'  ↑PAD CHARACTER:    \↑%C        \↑%C\N',
     +   O"100".XOR.(PADCH),O"100".XOR.(SPADCH),0,0)
      CALL FPRINTF(STDOUT,'  ↑END-OF-↑LINE:      \↑%C        \↑%C\N',
     +   O"100".XOR.(EOLCH),O"100".XOR.(SPEOL),0,0)
      CALL FPRINTF(STDOUT,'  ↑CONTROL QUOTE:    %C         %C\N',
     +   QUOTECH,SPQUOTE,0,0)
      CALL FPRINTF(STDOUT,'  ↑START-OF-↑PACKET:  \↑%C        \↑%C\N',
     +   O"100".XOR.(SYNC),O"100".XOR.(SNDSYNC),0,0)
C
```

```
C     DISPLAY PROTOCOL STUFF
C
      CALL FPRINTF(STDOUT,'\N↑DELAY BEFORE SENDING FIST PACKET: %D\N',
     +   DELAYFP,0,0,0)
      CALL FPRINTF(STDOUT,'↑INIT PACKET RETRY COUNT: %D\N',MAXRINI,0,0,
     +   0)
      CALL FPRINTF(STDOUT,'↑PACKET RETRY COUNT: %D\N\N',MAXRTRY,0,0,0)
      RETURN
      END
      SUBROUTINE STATUS

CCC   STATUS - TELL HOW LONG LAST TRANSFER TOOK.
C
      IMPLICIT INTEGER (A-Z)
      LOGICAL CONFIRM
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
```

```
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
C
C     CONFIRM THE COMMAND
C
      IF (.NOT. CONFIRM(STDIN)) RETURN
      CALL FPRINTF(STDOUT,
     +   '↑MAX CHARACTERS IN PACKET: %D RECEIVED; %D SENT\N',PACKSIZ,
     +   SPKSIZ,0,0)
      IF (ENDTIM .LT. STARTIM) ENDTIM = ENDTIM + 86400
      NSEC = ENDTIM - STARTIM
      HR = NSEC / 3600
      NSEC = NSEC - (HR * 3600)
      MIN = NSEC / 60
      NSEC = NSEC - (MIN * 60)
      CALL FPRINTF(STDOUT,'↑NUMBER OF CHARACTERS TRANSMITTED IN ',
     +   0,0,0,0)
      IF (HR .GT. 0) CALL FPRINTF(STDOUT,'%D HOURS ',HR,0,0,0)
      IF (MIN .GT. 0) CALL FPRINTF(STDOUT,'%D MINUTES ',MIN,0,0,0)
      CALL FPRINTF(STDOUT,'%D SECONDS\N\N',NSEC,0,0,0)
      CALL FPRINTF(STDOUT,'              ↑SENT:  %20D',SCHCNT,0,0,0)
      CALL FPRINTF(STDOUT,' ↑OVERHEAD:  %D\N',SCHOVRH,0,0,0)
      CALL FPRINTF(STDOUT,'          ↑RECEIVED:  %20D',RCHCNT,0,0,0)
      CALL FPRINTF(STDOUT,' ↑OVERHEAD:  %D\N',RCHOVRH,0,0,0)
      CALL FPRINTF(STDOUT,'↑TOTAL TRANSMITTED:  %20D',SCHCNT+RCHCNT,0,0,
     +   0)
      CALL FPRINTF(STDOUT,' ↑OVERHEAD:  %D\N',SCHOVRH+RCHOVRH,0,0,0)
      CALL FPRINTF(STDOUT,
     +   '↑TOTAL CHARACTERS TRANSMITTED PER SEC: %D\N',
     +   (SCHCNT+RCHCNT) / (ENDTIM-STARTIM),0,0,0)
      CALL FPRINTF(STDOUT,
     +   '↑EFFECTIVE DATA RATE: %D BAUD\N', ((SCHCNT+RCHCNT) -
     +   (SCHOVRH+RCHOVRH)) / (ENDTIM-STARTIM) * 10,0,0,0)
      IF (STATE .NE. C) THEN
         CALL GETEMSG(PACKET)
         CALL FPRINTF(STDOUT,'?↑KERMIT:  %S\N',PACKET,0,0,0)
      ENDIF
      RETURN
```

```
      END
      SUBROUTINE DBUGCMD

CCC   DBUGCMD - SET THE DEBUGGING MODES.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,EMSGCOM
C
C     MESSAGE COMMON BLOCK
C
      COMMON /MESSAGE/ VERSION, AMBIG, NOMATCH, FOLLOW, NODIGIT,MISSING,
     +   CONFMSG, NOTCONF,
     +   HLPASCH, HLPDLFP, HLPDBFN, HLPPLEN, HLPPADL, HLPIPRC, HLPPRTR,
     +   HLPTIMO, HLPSNFN
```

```
      CHARACTER*37 VERSION
      CHARACTER*15 AMBIG
      CHARACTER*38 NOMATCH
      CHARACTER*24 FOLLOW
      CHARACTER*53 NODIGIT
      CHARACTER*31 MISSING
      CHARACTER*33 CONFMSG
      CHARACTER*19 NOTCONF
      CHARACTER*42 HLPDLFP
      CHARACTER*37 HLPDBFN
      CHARACTER*24 HLPPLEN
      CHARACTER*34 HLPPADL
      CHARACTER*74 HLPASCH
      CHARACTER*29 HLPIPRC
      CHARACTER*21 HLPPRTR
      CHARACTER*43 HLPTIMO
      CHARACTER*19 HLPSNFN
C-----END OF FILE: EMSGCOM
      CHARACTER*10 FN
      LOGICAL CONFIRM
      PARAMETER (TSIZE=5)
      CHARACTER*10 DBGTYP(TSIZE)
      DATA DBGTYP / 'ALL', 'LOG-FILE', 'OFF', 'PACKETS', 'STATES' /

      INDX = MATCH(DBGTYP,TSIZE,.FALSE.)
      IF (INDX .LE. 0) RETURN
      GO TO (10, 20, 30, 40, 50), INDX
C
C     SET ALL DEBUG MODES
C
10    IF (.NOT. CONFIRM(STDIN)) RETURN
      DEBUG = DBGALL
      GO TO 100
C
C     SET DEBUG LOGFILE
C
20    CALL SETVAL(DEBUGFN,'S',IRET,7,0,0,HLPDBFN,.TRUE.)
      IF (IRET .EQ. OK) THEN
         IF (DEBUGFD .NE. 0) THEN
            CALL FCLOSE(DEBUGFD)
            DEBUGFD = 0
         ENDIF
         GO TO 100
      ENDIF
      RETURN
C
C     TURN OFF ALL DEBUGGING
C
30    IF (.NOT. CONFIRM(STDIN)) RETURN
      DEBUG = DBGOFF
      IF (DEBUGFD .NE. 0) THEN
         CALL FCLOSE(DEBUGFD)
         DEBUGFD = 0
      ENDIF
      RETURN
C
C     TOGGLE DEBUG PACKETS
C
40    IF (.NOT. CONFIRM(STDIN)) RETURN
      DEBUG = DEBUG .XOR. DBGPACK
```

```
      GO TO 100
C
C     TOGGLE DEBUG STATES
C
50    IF (.NOT. CONFIRM(STDIN)) RETURN
      DEBUG = DEBUG .XOR. DBGSTAT
      GO TO 100
C
C     OPEN THE DEBUG FILE IF NOT DONE ALREADY
C
100   IF (DEBUGFD .EQ. 0) THEN
         CALL AS2DPC(DEBUGFN,FN)
         DEBUGFD = FOPEN(FN,2)
      ENDIF
      RETURN
      END
      SUBROUTINE SETPACK(ATTR)

CCC   SET PACKET SEND OR RECEIVE ATTRIBUTES.
C
C     SETPACK WILL WET THE ATTRIBUTES OF THE PASSED ATTRIBUTE
C     LIST.  THIS SUBROUTINE WILL SET THE APPROPRIATE PACKET
C     PARAMETER.  THE PARAMETER TO SET IS PASSED IN AN ARRAY
C     AND IS VERY ORDER DEPENDENT.  SEE COMMON BLOCK /PACKET/
C     FOR THE ORDERING.  NOTE THAT SEND AND RECEIVE PARAMETER
C     ORDERING AND STORAGE SIZE IN THE COMMON BLOCK ARE
C     IDENTICAL.  KEEP IT THAT WAY!
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
```

```
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
C-----FILE: C167,EMSGCOM
C
C     MESSAGE COMMON BLOCK
C
      COMMON /MESSAGE/ VERSION, AMBIG, NOMATCH, FOLLOW, NODIGIT,MISSING,
     +   CONFMSG, NOTCONF,
     +   HLPASCH, HLPDLFP, HLPDBFN, HLPPLEN, HLPPADL, HLPIPRC, HLPPRTR,
     +   HLPTIMO, HLPSNFN
      CHARACTER*37 VERSION
      CHARACTER*15 AMBIG
      CHARACTER*38 NOMATCH
      CHARACTER*24 FOLLOW
      CHARACTER*53 NODIGIT
      CHARACTER*31 MISSING
      CHARACTER*33 CONFMSG
      CHARACTER*19 NOTCONF
      CHARACTER*42 HLPDLFP
      CHARACTER*37 HLPDBFN
      CHARACTER*24 HLPPLEN
      CHARACTER*34 HLPPADL
      CHARACTER*74 HLPASCH
      CHARACTER*29 HLPIPRC
      CHARACTER*21 HLPPRTR
      CHARACTER*43 HLPTIMO
      CHARACTER*19 HLPSNFN
C-----END OF FILE: EMSGCOM
      INTEGER ATTR(12)
      PARAMETER (TSIZE=7)
      CHARACTER*10 ATTRTYP(TSIZE)
      DATA ATTRTYP / 'EOL', 'PACKCHR', 'PADCHR', 'PADLEN',
     +               'QUOTECHR', 'SYNCCHR', 'TIMEOUT' /

      INDX = MATCH(ATTRTYP,TSIZE,.FALSE.)
      IF (INDX .LE. 0) RETURN
      GO TO (10, 20, 30, 40, 50, 60, 70), INDX
```

```
C
C     SET EOL CHARACTER
C
10    CALL SETVAL(ATTR(5),'I',1,31,127,127,HLPASCH,.TRUE.)
      RETURN
C
C     SET MAXIMUM PACKET LENGTH
C
20    CALL SETVAL(ATTR(1),'I',20,94,20,94,HLPPLEN,.TRUE.)
      RETURN
C
C     SET PAD LENGTH
C
30    CALL SETVAL(ATTR(3),'I',0,94,0,94,HLPPADL,.TRUE.)
      RETURN
C
C     SET PAD CHARACTER
C
40    CALL SETVAL(ATTR(4),'I',0,31,127,127,HLPASCH,.TRUE.)
      RETURN
C
C     SET QUOTE CHARACTER
C
50    CALL SETVAL(ATTR(6),'I',33,62,97,126,HLPASCH,.TRUE.)
      RETURN
C
C     SET SYNC CHARACTER
C
60    CALL SETVAL(ATTR(12),'I',0,127,0,127,HLPASCH,.TRUE.)
      RETURN
C
C     SET TIMEOUT VALUE
C
70    CALL SETVAL(ATTR(2),'I',0,94,0,94,HLPTIMO,.TRUE.)
      RETURN
      END
      SUBROUTINE DPLXCMD

CCC   DPLXCMD - PERFORM A SET DUPLEX XXXX COMMAND
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
```

```
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
      LOGICAL CONFIRM
      PARAMETER (TSIZE=2)
      CHARACTER*10 DUPTYP(TSIZE)
      DATA DUPTYP / 'HALF', 'FULL' /
C
C     MATCH THE PARAMETER
C
      INDX = MATCH(DUPTYP,TSIZE,.FALSE.)
      IF (INDX .LE. 0) RETURN
      IF (.NOT. CONFIRM(STDIN)) RETURN
C
C     TAKE THE APPROPRIATE ACTION
C
      GO TO (10, 20), INDX
C
C     SET HALF DUPLEX
C
10    CALL STTY('DUPLEX',HALF)
      RETURN
C
C     SET FULL DUPLEX
C
20    CALL STTY('DUPLEX',FULL)
      RETURN
      END
      SUBROUTINE PARCMD

CCC   PARCMD - SET THE PARITY FOR TERMINAL I/O.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
```

```
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
      LOGICAL CONFIRM
      PARAMETER (TSIZE=5)
      CHARACTER*10 PARTYP(TSIZE)
      DATA PARTYP / 'EVEN', 'MARK', 'NONE', 'ODD', 'SPACE' /
C
C     MATCH THE PARAMETER
C
      INDX = MATCH(PARTYP,TSIZE,.FALSE.)
      IF (INDX .LE. 0) RETURN
      IF (.NOT. CONFIRM(STDIN)) RETURN
C
C     SET THE PROPER PARITY
C
      GO TO (10, 20, 30, 40, 50), INDX
10    CALL STTY('PARITY',EVEN)
      RETURN
20    CALL STTY('PARITY',MARK)
      RETURN
30    CALL STTY('PARITY',NONE)
      RETURN
40    CALL STTY('PARITY',ODD)
      RETURN
50    CALL STTY('PARITY',SPACE)
      RETURN
      END
      INTEGER FUNCTION MATCH(TABLE,TABLEN,NELOK)

CCC   MATCH - MATCH INPUT WITH A TABLE OF POSSIBILITIES.
C
C     TABLE SHOULD BE AN ARRAY OF CHARACTER STRINGS DEFINING WHAT
C     IS REASONABLE INPUT.  MATCH WILL READ INPUT AND RETURN THE
```

```
C     INDEX OF THE TABLE ENTRY THAT MATCHES OR "ERROR" IF A PROPER
C     MATCH COULDN'T BE MADE.  MATCHS WILL FAIL IF THE INPUT MATCH
C     IS AMBIGUOUS OR DOESN'T MATCH AT ALL.  A QUESTION MARK IN THE
C     INPUT WILL OUTPUT THE POSSIBLE MATCHES ACCORDING TO THE INPUT
C     PREVIOUSLY READ AND THEN RETURN AS IF NO MATCH WAS MADE.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,EMSGCOM
C
C     MESSAGE COMMON BLOCK
C
      COMMON /MESSAGE/ VERSION, AMBIG, NOMATCH, FOLLOW, NODIGIT,MISSING,
     +   CONFMSG, NOTCONF,
     +   HLPASCH, HLPDLFP, HLPDBFN, HLPPLEN, HLPPADL, HLPIPRC, HLPPRTR,
     +   HLPTIMO, HLPSNFN
      CHARACTER*37 VERSION
      CHARACTER*15 AMBIG
      CHARACTER*38 NOMATCH
      CHARACTER*24 FOLLOW
      CHARACTER*53 NODIGIT
      CHARACTER*31 MISSING
```

```
      CHARACTER*33 CONFMSG
      CHARACTER*19 NOTCONF
      CHARACTER*42 HLPDLFP
      CHARACTER*37 HLPDBFN
      CHARACTER*24 HLPPLEN
      CHARACTER*34 HLPPADL
      CHARACTER*74 HLPASCH
      CHARACTER*29 HLPIPRC
      CHARACTER*21 HLPPRTR
      CHARACTER*43 HLPTIMO
      CHARACTER*19 HLPSNFN
C-----END OF FILE: EMSGCOM
      CHARACTER*(*) TABLE(TABLEN)
      LOGICAL NELOK
      CHARACTER*40 WORD
      INTEGER ASTR(41)
C
C     SET UP COLLATING SEQUENCE FOR MATCHING
C
      CALL COLSEQ('DISPLAY')
C
C     GET THE WORD TO MATCH
C
      LEN = GETWORD(STDIN,ASTR,40)
      IF (LEN .EQ. 0 .OR. LEN .EQ. EOF) THEN
         MATCH = LEN
         IF (LEN .EQ. 0 .AND. .NOT. NELOK) THEN
            MATCH = ERROR
            CALL FPRINTF(STDOUT,'?↑NULL SWITCH OR KEYWORD GIVEN\N',0,0,
     +         0,0)
         ENDIF
         CALL FFLUSH(STDIN)
         RETURN
      ENDIF
      CALL AS2DPC(ASTR,WORD)
C
C     BEGIN THE MATCHING HERE; TABLES MUST BE IN ALPHABETICAL ORDER
C
      T1 = 1
      T2 = TABLEN
      CHP = 1
10    IF (CHP .LE. LEN) THEN
C
C        IF WE FIND A "?", THEN GIVE THE POSSIBILITIES
C
         IF (WORD(CHP:CHP) .EQ. '?') THEN
            CALL FPRINTF(STDOUT,FOLLOW,0,0,0,0)
            CALL OUTTBL(TABLE,T1,T2)
            CALL FFLUSH(STDIN)
            MATCH = ERROR
            RETURN
         ENDIF
C
C        WHILE WORD IS LESS THAN LOWER TABLE ENTRY
C
20       IF (WORD(CHP:CHP) .GT. TABLE(T1)(CHP:CHP) .AND.
     +       T1 .LE. T2) THEN
            T1 = T1 + 1
            GO TO 20
         ENDIF
```

```
C
C        WHILE WORD IS GREATER THAN UPPER TABLE ENTRY
C
30       IF (WORD(CHP:CHP) .LT. TABLE(T2)(CHP:CHP) .AND.
     +       T2 .GE. T1) THEN
            T2 = T2 - 1
            GO TO 30
         ENDIF
C
C        IF WE KNOW WE HAVE A MISMATCH
C
         IF (T2 .LT. T1) THEN
            CALL FPRINTF(STDOUT,NOMATCH,0,0,0,0)
            CALL PUTSTR(STDOUT,ASTR)
            CALL FPRINTF(STDOUT,'"\N',0,0,0,0)
            CALL FFLUSH(STDIN)
            MATCH = ERROR
            RETURN
         ENDIF
         CHP = CHP + 1
         GO TO 10
      ENDIF
C
C     AFTER LOOKING AT THE WHOLE WORD, IS IT STILL AMBIGUOUS?
C
      IF (T1 .NE. T2) THEN
         CALL FPRINTF(STDOUT,AMBIG,0,0,0,0)
         CALL PUTSTR(STDOUT,ASTR)
         CALL FPRINTF(STDOUT,'"\N',0,0,0,0)
         CALL FFLUSH(STDIN)
         MATCH = ERROR
      ELSE
         MATCH = T1
      ENDIF
      RETURN
      END
      SUBROUTINE OUTTBL(TABLE,START,FIN)

CCC   OUTTBL - OUTPUT A STRING ARRAY IN TABULAR FORMAT.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
```

```
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
      CHARACTER*(*) TABLE(FIN)
      INTEGER START, FIN

      CHARACTER*80 LINE
      INTEGER ASTR(81)
      INTEGER COLWID, NCOLS

      COLWID = LEN(TABLE(1)) + 2
      NCOLS = 80 / COLWID
      LINE = ' '
      ICOL = 1
      DO 100 I = START,FIN
         IPOS = (ICOL-1)*COLWID + 1
         LINE(IPOS:) = TABLE(I)
         ICOL = ICOL + 1
         IF (ICOL .GT. NCOLS .OR. I .EQ. FIN) THEN
            CALL DPC2AS(LINE,ASTR,LEN(LINE))
C
C           DELETE TRAILING BLANKS
C
            J = LEN(LINE)
10          IF (LINE(J:J) .EQ. ' ') THEN
               ASTR(J) = 0
               J = J - 1
               GO TO 10
            ENDIF
            CALL PUTSTR(STDOUT,ASTR)
            CALL PUTC(NEL,STDOUT)
            LINE = ' '
            ICOL = 1
         ENDIF
100   CONTINUE
      RETURN
      END
      SUBROUTINE SETVAL(VAR,VTYP,MN1,MX1,MN2,MX2,HLPMSG,CONFRM)

CCC   SETVAL - SET A VARIABLE VALUE.
C
C     SETVAL WILL READ A TOKEN FROM INPUT AND SET A VARIABLE TO
C     THAT VALUE.  IF THE TOKEN IS A QUESTION MARK THEN THE
```

```
C     HELP MESSAGE WILL BE DISPLAYED AND SETVAL WILL RETURN
C     WITHOUT SETTING A VALUE.
C
C     ENTRY:   (VTYP) = CHARACTER 'S' FOR STRING VARIABLE.
C                     = CHARACTER 'I' FOR INTEGER VARIABLE.
C              (MN1-MX1) = RANGE #1 FOR VAR TO FIT IN IF INTEGER.
C                        = MN1 IS RETURN CODE FOR ERROR AND MX1 IS
C                          MAX SIZE OF STRING IF STRING VAR.
C              (MN2-MX2) = SECONDARY RANGE FOR VAR TO FIT IN IF
C                          INTEGER VAR.
C                        = UNUSED FOR STRING VAR.
C              (HLPMSG) = FPRINTF MESSAGE FORMAT TO DISPLAY IF
C                       A QUESTION MARK IS READ.
C
C     EXIT:    (VAR) = INT VALUE READ IF INTEGER VAR. OR STRING
C                    VALUE READ IF STRING VAR.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,EMSGCOM
C
C     MESSAGE COMMON BLOCK
```

```
C
      COMMON /MESSAGE/ VERSION, AMBIG, NOMATCH, FOLLOW, NODIGIT,MISSING,
     +   CONFMSG, NOTCONF,
     +   HLPASCH, HLPDLFP, HLPDBFN, HLPPLEN, HLPPADL, HLPIPRC, HLPPRTR,
     +   HLPTIMO, HLPSNFN
      CHARACTER*37 VERSION
      CHARACTER*15 AMBIG
      CHARACTER*38 NOMATCH
      CHARACTER*24 FOLLOW
      CHARACTER*53 NODIGIT
      CHARACTER*31 MISSING
      CHARACTER*33 CONFMSG
      CHARACTER*19 NOTCONF
      CHARACTER*42 HLPDLFP
      CHARACTER*37 HLPDBFN
      CHARACTER*24 HLPPLEN
      CHARACTER*34 HLPPADL
      CHARACTER*74 HLPASCH
      CHARACTER*29 HLPIPRC
      CHARACTER*21 HLPPRTR
      CHARACTER*43 HLPTIMO
      CHARACTER*19 HLPSNFN
C-----END OF FILE: EMSGCOM
      CHARACTER*(*) VTYP, HLPMSG
      INTEGER VAR(41), STR(41)
      LOGICAL CONFRM, CONFIRM
C
C     CHECK VAR TYPE
C
      IF (VTYP .NE. 'S' .AND. VTYP .NE. 'I') THEN
         CALL FPRINTF(STDOUT,'SETVAL - INVALID VAR TYPE %C\N',ASC(VTYP),
     +      0,0,0)
         RETURN
      ENDIF
      IF (VTYP .EQ. 'S' .AND. MX1 .GT. 40) THEN
         CALL FPRINTF(STDOUT,'SETVAL - STRING MAX OF %D IS TOO LARGE\N',
     +      MX1,0,0,0)
         RETURN
      ENDIF
      LEN = GETWORD(STDIN,STR,MX1)
      IF (LEN .EQ. 0 .OR. LEN .EQ. EOF) THEN
         IF (VTYP .EQ. 'I') THEN
            CALL FPRINTF(STDOUT,NODIGIT,0,0,0,0)
         ELSE
            CALL FPRINTF(STDOUT,MISSING,0,0,0,0)
            MN1 = ERROR
         ENDIF
         RETURN
      ENDIF
      IF (STR(1) .EQ. QMARK) THEN
         CALL FPRINTF(STDOUT,HLPMSG,0,0,0,0)
         CALL FFLUSH(STDIN)
         IF (VTYP .EQ. 'S') MN1 = ERROR
         RETURN
      ENDIF
C
C     CONFIRM THE REQUEST IF NECESSARY
C
      IF (CONFRM) THEN
         IF (.NOT. CONFIRM(STDIN)) THEN
```

```
            IF (VTYP .EQ. 'S') MN1 = ERROR
            RETURN
         ENDIF
      ENDIF
C
C     GO AHEAD AND SET THE VARIABLE
C
      IF (VTYP .EQ. 'I') THEN
         I = CTOI(STR)
         IF (I .GE. MN1 .AND. I .LE. MX1) THEN
            VAR(1) = I
         ELSE IF (I .GE. MN2 .AND. I .LE. MX2) THEN
            VAR(2) = I
         ELSE
            CALL FPRINTF(STDOUT,
     +         '?↑VALUE IS NOT WITHIN RANGE OF %D - %D',
     +         MN1,MX1,0,0)
            IF (MN1 .NE. MN2 .OR. MX1 .NE. MX2) CALL FPRINTF(STDOUT,
     +         ' OR %D - %D',MN2,MX2,0,0)
            CALL PUTC(NEL,STDOUT)
         ENDIF
      ELSE
         DO 100 I = 1,LEN
            VAR(I) = STR(I)
100      CONTINUE
         VAR(LEN+1) = 0
         MN1 = OK
      ENDIF
      RETURN
      END
      LOGICAL FUNCTION CONFIRM(FD)

CCC   CONFIRM - LOOK FOR A NEWLINE.
C
C     CONFIRM WILL EXPECT THAT THE NEXT TOKEN OF INPUT BE A
C     NEWLINE FOR CONFIRMATION TO BE TRUE.  IF THE NEXT TOKEN
C     IS A QUESTION MARK, THEN CONFIRMATION IS FALSE AND
C     A "CONFIRM WITH A CARRIAGE RETURN" MESSAGE WILL BE DISPLAYED.
C     ANY OTHER TEXT WILL CAUSE A 'NOT CONFIRMED "TEXT"' MESSAGE
C     TO BE DISPLAYED AND CONFIRM WILL RETURN FALSE.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
```

```
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,EMSGCOM
C
C     MESSAGE COMMON BLOCK
C
      COMMON /MESSAGE/ VERSION, AMBIG, NOMATCH, FOLLOW, NODIGIT,MISSING,
     +   CONFMSG, NOTCONF,
     +   HLPASCH, HLPDLFP, HLPDBFN, HLPPLEN, HLPPADL, HLPIPRC, HLPPRTR,
     +   HLPTIMO, HLPSNFN
      CHARACTER*37 VERSION
      CHARACTER*15 AMBIG
      CHARACTER*38 NOMATCH
      CHARACTER*24 FOLLOW
      CHARACTER*53 NODIGIT
      CHARACTER*31 MISSING
      CHARACTER*33 CONFMSG
      CHARACTER*19 NOTCONF
      CHARACTER*42 HLPDLFP
      CHARACTER*37 HLPDBFN
      CHARACTER*24 HLPPLEN
      CHARACTER*34 HLPPADL
      CHARACTER*74 HLPASCH
      CHARACTER*29 HLPIPRC
      CHARACTER*21 HLPPRTR
      CHARACTER*43 HLPTIMO
      CHARACTER*19 HLPSNFN
C-----END OF FILE: EMSGCOM
C
C     GET LEADING BLANKS TIL A TOKEN IS FOUND
C
      CONFIRM = .FALSE.
10    IF (GETC(FD,CH) .EQ. NEL) THEN
         CONFIRM = .TRUE.
      ELSE IF (CH .EQ. EOF) THEN
         RETURN
      ELSE IF (CH .EQ. BLANK .OR. CH .EQ. TAB) THEN
         GO TO 10
      ELSE IF (CH .EQ. QMARK) THEN
         CALL FPRINTF(STDOUT,CONFMSG,0,0,0,0)
      ELSE
```

```
         CALL FPRINTF(STDOUT,NOTCONF,0,0,0,0)
20       CALL PUTC(CH,STDOUT)
         CH = GETC(FD,CH)
         IF (CH .NE. NEL .AND. CH .NE. EOF) GO TO 20
         CALL FPRINTF(STDOUT,'"\N',0,0,0,0)
      ENDIF
      RETURN
      END
      SUBROUTINE SERVER

CCC   SERVER - START KERMIT SERVER
C
C     THE SERVER CURRENTLY KNOWS ABOUT THE SEND AND RECEIVE PACKETS
C     AND ALSO THE GENERIC KERMIT PACKETS LOGOUT AND FINISH.  USING
C     LOGOUT CAN CAUSE PROBLEM DUE TO FILES NOT BEING MADE PERMANENT
C     BEFORE LEAVING.  I SUPPOSE IMPLEMENTING SYSTEM COMMAND PACKETS
C     WOULD ALLOW FILES TO BE SAVED BUT WHAT OTHER KERMIT ALLOWS
C     SYSTEM COMMAND PACKETS AND IS THERE A STANDARD WHAT TO
C     CHECKPOINT PROGRAMS ON THE CYBER?
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
```

```
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
      LOGICAL CONFIRM
C
C     CONFIRM THE COMMAND
C
      IF (.NOT. CONFIRM(STDIN)) RETURN
C
C     INITIALIZE MSG #, SAY NO TRIES YET
C
      PACKNUM = 0
      NUMTRY = 0
      CALL FPRINTF(STDOUT,'[↑KERMIT SERVER RUNNING ON ↑CYBER HOST.  ↑PLE
     +ASE TYPE YOUR ESCAPE SEQUENCE TO\N RETURN TO YOUR LOCAL MACHINE. ↑
     +SHUT DOWN THE SERVER BY TYPING THE ↑KERMIT ↑B↑Y↑E \N COMMAND ON YO
     +UR LOCAL MACHINE.]\N',0,0,0,0)
      CALL STTY('BINARY',ON)
      SAVEDPX = GTTY('DUPLEX')
      CALL STTY('DUPLEX',HALF)
    1 PTYP = RDPACK(LEN,NUM,RECPACK)
      IF (PTYP .EQ. S) THEN
         PACKNUM = NUM
         CALL RDPARAM(RECPACK)
         I = SNDPAR(PACKET)
         CALL SNDPACK(Y,PACKNUM,I,PACKET)
         NUMTRY = 0
         PACKNUM = MOD(PACKNUM+1,64)
         RECSTAT = RECEIVE(F)
         IF (DEBUG .NE. 0) THEN
            IF (RECSTAT .EQ. ERROR) THEN
               CALL FPRINTF(DEBUGFD,'↑RECEIVE FAILED.\N',0,0,0,0)
            ELSE
               CALL FPRINTF(DEBUGFD,'↑RECEIVE COMPLETED.\N',0,0,0,0)
            ENDIF
         ENDIF
      ELSE IF (PTYP .EQ. R) THEN
            I = 0
            CALL STRCPY(RECPACK,FILESTR)
            SNDSTAT = SEND()
            PACKNUM = 0
            IF (DEBUG .NE. 0) THEN
               IF (SNDSTAT .EQ. ERROR) THEN
```

```
                CALL FPRINTF(DEBUGFD,'↑SEND FAILED.\N',0,0,0,0)
             ELSE
                CALL FPRINTF(DEBUGFD,'↑SEND COMPLETED.\N',0,0,0,0)
             ENDIF
          ENDIF
      ELSE IF (PTYP .EQ. G) THEN
          IF (RECPACK(1) .EQ. L) THEN
             CALL SNDPACK(Y,NUM,0,0)
             CALL LOGOUT
          ELSE
             IF (RECPACK(1) .EQ. F) THEN
                CALL SNDPACK(Y,NUM,0,0)
                CALL EXITPGM
             ENDIF
          ENDIF
      ELSE
         IF (DEBUG .NE. 0) CALL FPRINTF
     +      (DEBUGFD,'SERVER: INVALID PACKET TYPE: %D\N',PTYP,0,0,0)
      ENDIF
      GO TO 1
      END
      SUBROUTINE LOGOUT

CCC   LOGOUT - LOG OUT THE JOB
C
C     THIS IS SITE DEPENDENT.
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMSIT
C
C     KERMIT SITE DEFINITIONS:
C
      PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
```

```
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C$    IF (UT2D .EQ. ON)
      CALL BELLC(L"LOGOUT",0,IRET)
C$    ELSE
      IRET = ERROR
C$    ENDIF
      IF (IRET .NE. 0) CALL DISPLA('LOGOUT ERROR',IRET)
      RETURN
      END
      INTEGER FUNCTION RECEIVE(ISTATE)

CCC   RECEIVE - RECEIVE FILE STATE SWITCHING ROUTINE.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
```

```
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
      INTEGER MSG(MAXPACK)
      DATA (MSG(I),I=1,14) / 63, 75, 101, 114, 109, 105, 116, 45, 49,
C                             ?   K    E    R    M    I    T   -   1
     +                       55, 48, 58, 2*32 /
C                             7   0   :
C
C     INITIALIZE STATISTICS VARIABLES
C
      CALL GETNOW(MM,DD,YY,HR,MIN,SEC)
      STARTIM = HR * 3600 + MIN * 60 + SEC
      SCHCNT = 0
      RCHCNT = 0
      SCHOVRH = 0
      RCHOVRH = 0
C
C     SET PACKET RETRY COUNT & CURRENT STATE
C
      NUMTRY = 0
      STATE = ISTATE
C
C     TAKE APPROPRIATE ACTION FOR THE CURRENT STATE
C
10    IF (STATE .EQ. D) THEN
         STATE = RDATA()
      ELSE IF (STATE .EQ. F) THEN
         STATE = RFILE()
      ELSE IF (STATE .EQ. R) THEN
         STATE = RINIT()
      ELSE IF (STATE .EQ. C) THEN
         CALL GETNOW(MM,DD,YY,HR,MIN,SEC)
         ENDTIM = HR * 3600 + MIN * 60 + SEC
         RECEIVE = OK
         RETURN
      ELSE IF (STATE .EQ. A) THEN
         CALL GETNOW(MM,DD,YY,HR,MIN,SEC)
         ENDTIM = HR * 3600 + MIN * 60 + SEC
```

```
         RECEIVE = ERROR
         IF (FFD .NE. 0) CALL FCLOSE(FFD)
         IF ((ABORTYP.AND.LCLFILE) .EQ. 0) CALL REMOVE(FILESTR)
         CALL GETEMSG(MSG(15))
         CALL SNDPACK(E,PACKNUM,SLEN(MSG),MSG)
         RETURN
      ELSE
         CALL DISPLA(' RECEIVE - STATE ERROR = ',STATE)
         IF (FFD .NE. 0) CALL FCLOSE(FFD)
         RECEIVE = ERROR
         RETURN
      ENDIF
      IF ((DEBUG.AND.DBGSTAT).NE.0) THEN
         CALL FPRINTF(DEBUGFD,'%C%2D ',STATE,PACKNUM,0,0)
         IF (MOD(PACKNUM+1,16) .EQ. 0) CALL PUTC(NEL,DEBUGFD)
      ENDIF
      GO TO 10
      END
      INTEGER FUNCTION RINIT()

CCC   RINIT - RECEIVE A SEND-INIT PACKET.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
```

```
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C
C     CHECK RETRY COUNT
C
      NUMTRY = NUMTRY + 1
      IF (NUMTRY .GT. MAXRINI) THEN
         RINIT = A
         ABORTYP = TOOMANY.OR.READING.OR.INITERR
         RETURN
      ENDIF
C
C     READ A PACKET AND HOPE FOR THE BEST
C
      PTYP = RDPACK(LEN,NUM,PACKET)
C
C     IS IT A VALID PACKET TYPE?
C
      IF (PTYP .EQ. S) THEN
         PACKNUM = NUM
         CALL RDPARAM(PACKET)
         LEN = SNDPAR(PACKET)
         CALL SNDPACK(Y,NUM,LEN,PACKET)
         NUMTRY = 0
         PACKNUM = MOD(PACKNUM+1,64)
         RINIT = F
C
C     DID WE GET A CHECKSUM ERROR
C
      ELSE IF (PTYP .EQ. ERROR) THEN
         RINIT = STATE
         CALL SNDPACK(N,NUM,0,0)
      ELSE
         RINIT = A
         ABORTYP = INVALID.OR.READING.OR.INITERR
      ENDIF
      RETURN
      END
      INTEGER FUNCTION RFILE()
```

```
CCC   RFILE  READ A FILENAME PACKET.
C
C     RFILE EXPECTS TO SEE A FILENAME (TYPE F) PACKET.  HOWEVER, IT MAY
C     FIND A SEND-INIT RETRY, END-OF-FILE RETRY OR BREAK PACKET.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
```

```
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
      NUMTRY = NUMTRY + 1
      IF (NUMTRY .GT. MAXRTRY) THEN
         RFILE = A
         ABORTYP = TOOMANY.OR.READING.OR.FILERR
         RETURN
      ENDIF
C
C     READ A PACKET
C
      PTYP = RDPACK(LEN,NUM,PACKET)
C
C     IS IT A FILENAME PACKET?
C
      IF (PTYP .EQ. F) THEN
         IF (NUM .NE. PACKNUM) THEN
            RFILE = A
            ABORTYP = SEQERR.OR.READING.OR.FILERR
            RETURN
         ENDIF
         IF (DEBUG .NE. 0) CALL FPRINTF(DEBUGFD,'↑RECEIVING FILE %S\N',
     +      PACKET,0,0,0)
         FFD = GETFILE(PACKET)
         IF (FFD .EQ. ERROR) THEN
            FFD = 0
            RFILE = A
            ABORTYP = LCLFILE.OR.READING.OR.FILERR
         ELSE
            CALL STRCPY(PACKET,FILESTR)
            CALL SNDPACK(Y,NUM,0,0)
            NUMTRY = 0
            PACKNUM = MOD(PACKNUM+1,64)
            RFILE = D
         ENDIF
C
C     IS IT AN OLD SEND-INIT PACKET
C
      ELSE IF (PTYP .EQ. S) THEN
         IF (MOD(NUM+1,64) .EQ. PACKNUM) THEN
            LEN = SNDPAR(PACKET)
            CALL SNDPACK(Y,NUM,LEN,PACKET)
            NUMTRY = 0
            RFILE = STATE
         ELSE
            RFILE = A
            ABORTYP = SEQERR.OR.READING.OR.INITERR
         ENDIF
C
C     IS IT AN OLD EOF PACKET?
C
      ELSE IF (PTYP .EQ. Z) THEN
         IF (MOD(NUM+1,64) .EQ. PACKNUM) THEN
            CALL SNDPACK(Y,NUM,0,0)
            NUMTRY = 0
            RFILE = STATE
         ELSE
```

```
            RFILE = A
            ABORTYP = SEQERR.OR.READING.OR.EOFERR
         ENDIF
C
C     IS IT A BREAK PACKET?
C
      ELSE IF (PTYP .EQ. B) THEN
         IF (NUM .NE. PACKNUM) THEN
            RFILE = A
            ABORTYP = SEQERR.OR.READING.OR.BRKERR
         ELSE
            CALL SNDPACK(Y,PACKNUM,0,0)
            RFILE = C
         ENDIF
C
C     DID WE GET A CHECKSUM ERROR?
C
      ELSE IF (PTYP .EQ. ERROR) THEN
         RFILE = STATE
         CALL SNDPACK(N,NUM,0,0)
C
C     INVALID PACKET TYPE, SO ABORT
C
      ELSE
         RFILE = A
         ABORTYP = INVALID.OR.READING.OR.FILERR
      ENDIF
      RETURN
      END
      INTEGER FUNCTION RDATA()

CCC   RDATA - READ A DATA PACKET.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
```

```
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C
C     CHECK RETRY COUNT
C
      NUMTRY = NUMTRY + 1
      IF (NUMTRY .GT. MAXRTRY) THEN
         RDATA = A
         ABORTYP = TOOMANY.OR.READING.OR.DATAERR
         RETURN
      ENDIF
C
C     READ A PACKET
C
      PTYP = RDPACK(LEN,NUM,PACKET)
C
C     DID WE GET A DATA PACKET?
C
      IF (PTYP .EQ. D) THEN
         IF (NUM .NE. PACKNUM) THEN
            IF (MOD(NUM+1,64) .EQ. PACKNUM) THEN
               CALL SNDPACK(Y,NUM,0,0)
               RDATA = STATE
            ELSE
               RDATA = A
               ABORTYP = SEQERR.OR.READING.OR.DATAERR
            ENDIF
         ELSE
            CALL BUFEMP(PACKET,FFD,LEN)
```

```
            CALL SNDPACK(Y,PACKNUM,0,0)
            NUMTRY = 0
            PACKNUM = MOD(PACKNUM+1,64)
            RDATA = STATE
         ENDIF
C
C     IS IT AN OLD FILENAME PACKET?
C
      ELSE IF (PTYP .EQ. F) THEN
         IF (MOD(NUM+1,64) .EQ. PACKNUM) THEN
            CALL SNDPACK(Y,NUM,0,0)
            NUMTRY = 0
            RDATA = STATE
         ELSE
            RDATA = A
            ABORTYP = SEQERR.OR.READING.OR.FILERR
         ENDIF
C
C     IS IT AN EOF PACKET?
C
      ELSE IF (PTYP .EQ. Z) THEN
         IF (NUM .NE. PACKNUM) THEN
            RDATA = A
            ABORTYP = SEQERR.OR.READING.OR.EOFERR
         ELSE
            CALL SNDPACK(Y,PACKNUM,0,0)
            CALL FCLOSE(FFD)
            FFD = 0
            PACKNUM = MOD(PACKNUM+1,64)
            RDATA = F
         ENDIF
      ELSE IF (PTYP .EQ. ERROR) THEN
         RDATA = STATE
         CALL SNDPACK(N,NUM,0,0)
      ELSE
         RDATA = A
         ABORTYP = INVALID.OR.READING.OR.DATAERR
      ENDIF
      RETURN
      END
      INTEGER FUNCTION SEND()

CCC   SEND - SEND FILE STATE SWITCHING ROUTINE
C
C     THE FILENAME TO SEND IS ASSUMED TO HAVE ALREADY BEEN
C     OBTAINED AND SET IN ASCII STRING BUFFER FILESTR.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
```

```
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
      INTEGER MSG(MAXPACK)
      DATA (MSG(I),I=1,14) / 63, 75, 101, 114, 109, 105, 116, 45, 49,
C                             ?   K    E    R    M    I    T   -   1
     +                       55, 48, 58, 2*32 /
C                             7   0   :
C
C     INITIALIZE STATICS VARIABLES
C
      CALL GETNOW(MM,DD,YY,HR,MIN,SEC)
      STARTIM = HR * 3600 + MIN * 60 + SEC
      SCHCNT = 0
      RCHCNT = 0
      SCHOVRH = 0
```

```
      RCHOVRH = 0
      STATE = S
      NUMTRY = 0
C
C     TAKE APPROPRIATE ACTION FOR THE CURRENT STATE
C
10    IF (STATE .EQ. D) THEN
         STATE = SDATA()
      ELSE IF (STATE .EQ. F) THEN
         STATE = SFILE()
      ELSE IF (STATE .EQ. Z) THEN
         STATE = SEOF()
      ELSE IF (STATE .EQ. S) THEN
         STATE = SINIT()
      ELSE IF (STATE .EQ. B) THEN
         STATE = SBREAK()
      ELSE IF (STATE .EQ. C) THEN
         CALL GETNOW(MM,DD,YY,HR,MIN,SEC)
         ENDTIM = HR * 3600 + MIN * 60 + SEC
         SEND = OK
         RETURN
      ELSE IF (STATE .EQ. A) THEN
         CALL GETNOW(MM,DD,YY,HR,MIN,SEC)
         ENDTIM = HR * 3600 + MIN * 60 + SEC
         SEND = ERROR
         IF (FFD .NE. 0) CALL FCLOSE(FFD)
         CALL GETEMSG(MSG(15))
         CALL SNDPACK(E,PACKNUM,SLEN(MSG),MSG)
         RETURN
      ELSE
         CALL DISPLA(' SEND - STATE ERROR = ',STATE)
         SEND = ERROR
         IF (FFD .NE. 0) CALL FCLOSE(FFD)
         RETURN
      ENDIF
      IF ((DEBUG.AND.DBGSTAT).NE.0) THEN
         CALL FPRINTF(DEBUGFD,'%C%2D ',STATE,PACKNUM,0,0)
         IF (MOD(PACKNUM+1,16) .EQ. 0) CALL PUTC(NEL,DEBUGFD)
      ENDIF
      GO TO 10
      END
      INTEGER FUNCTION SINIT()

CCC   SINIT - SEND THE SEND-INIT PACKET AND WAIT FOR REPLY.
C
      IMPLICIT INTEGER (A-Z)
      CHARACTER*10 FILENAM
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
```

```
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C
C     CHECK NUMBER OF RETRIES
C
      NUMTRY = NUMTRY + 1
      IF (NUMTRY .GT. MAXRINI) THEN
         SINIT = A
         ABORTYP = TOOMANY.OR.SENDING.OR.INITERR
         RETURN
      ENDIF
C
C     SEND THE SEND-INIT PACKET WITH THE RIGHT INFO
C
      LEN = SNDPAR(PACKET)
      CALL SNDPACK(S,PACKNUM,LEN,PACKET)
```

```
C
C     PICK UP AND PROCESS THE REPLY
C
      PTYP = RDPACK(LEN,NUM,RECPACK)
      IF (PTYP .EQ. N) THEN
         SINIT = STATE
         RETURN
      ELSE IF (PTYP .EQ. Y) THEN
         IF (PACKNUM .NE. NUM) THEN
            SINIT = STATE
            RETURN
         ENDIF
         CALL RDPARAM(RECPACK)
         NUMTRY = 0
         PACKNUM = MOD(PACKNUM+1,64)
         CALL AS2DPC(FILESTR,FILENAM)
         FFD = FOPEN(FILENAM,1)
         IF (FFD .EQ. ERROR) THEN
            SINIT = A
         ELSE
            SINIT = F
         ENDIF
      ELSE IF (PTYP .EQ. ERROR) THEN
         SINIT = STATE
      ELSE
         SINIT = A
         ABORTYP = INVALID.OR.SENDING.OR.INITERR
      ENDIF
      RETURN
      END
      INTEGER FUNCTION SFILE()

CCC   SFILE - SEND A FILENAME PACKET AND WAIT FOR REPLY.
C
C     THE FILENAME IS ASSUMED TO HAVE BEEN PREVIOUSLY OBTAINED
C     AND STORED IN THE ASCII STRING BUFFER FILESTR.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
```

```
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C
C     HAVE WE TRIED THIS TOO MANY TIMES?
C
      NUMTRY = NUMTRY + 1
      IF (NUMTRY .GT. MAXRTRY) THEN
         SFILE = A
         ABORTYP = TOOMANY.OR.SENDING.OR.FILERR
         RETURN
      ENDIF
C
C     SEND THE FILENAME PACKET
C
      CALL SNDPACK(F,PACKNUM,SLEN(FILESTR),FILESTR)
C
C     CHECK ON THE REPLY
C
      PTYP = RDPACK(LEN,NUM,RECPACK)
      IF (PTYP .EQ. N) THEN
         IF (MOD(PACKNUM+1,64) .NE. NUM) THEN
            SFILE = STATE
            RETURN
         ELSE
            PTYP = Y
```

```
            NUM = NUM - 1
         ENDIF
      ENDIF
      IF (PTYP .EQ. Y) THEN
         IF (PACKNUM .NE. NUM) THEN
            SFILE = STATE
            RETURN
         ENDIF
         NUMTRY = 0
         PACKNUM = MOD(PACKNUM+1,64)
C
C        GET FIRST PACKET OF DATA FROM THE FILE
C
         PSIZE = BUFFILL(FFD,PACKET)
         SFILE = D
      ELSE IF (PTYP .EQ. ERROR) THEN
         SFILE = STATE
      ELSE
         SFILE = A
         ABORTYP = INVALID.OR.SENDING.OR.FILERR
      ENDIF
      RETURN
      END
      INTEGER FUNCTION SDATA()

CCC   SDATA - SEND A DATA PACKET AND WAIT FOR REPLY.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
```

```
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C
C     HAVE WE TRIED THIS TOO MANY TIMES?
C
      NUMTRY = NUMTRY + 1
      IF (NUMTRY .GT. MAXRTRY) THEN
         SDATA = A
         ABORTYP = TOOMANY.OR.SENDING.OR.DATAERR
         RETURN
      ENDIF
C
C     SEND THE CURRENT DATA BUFFER
C
      IF (PSIZE .EQ. EOF) THEN
         SDATA = Z
         RETURN
      ENDIF
      CALL SNDPACK(D,PACKNUM,PSIZE,PACKET)
C
C     CHECK ON THE REPLY
C
      PTYP = RDPACK(LEN,NUM,RECPACK)
      IF (PTYP .EQ. N) THEN
         IF (MOD(PACKNUM+1,64) .NE. NUM) THEN
            SDATA = STATE
            RETURN
         ELSE
            PTYP = Y
            NUM = NUM - 1
         ENDIF
      ENDIF
      IF (PTYP .EQ. Y) THEN
         IF (PACKNUM .NE. NUM) THEN
            SDATA = STATE
```

```
            RETURN
         ENDIF
         NUMTRY = 0
         PACKNUM = MOD(PACKNUM+1,64)
         PSIZE = BUFFILL(FFD,PACKET)
         IF (PSIZE .EQ. EOF) THEN
            SDATA = Z
         ELSE
            SDATA = STATE
         ENDIF
      ELSE IF (PTYP .EQ. ERROR) THEN
         SDATA = STATE
      ELSE
         SDATA = A
         ABORTYP = INVALID.OR.SENDING.OR.DATAERR
      ENDIF
      RETURN
      END
      INTEGER FUNCTION SEOF()

CCC   SEOF - SEND AN EOF PACKET AND WAIT FOR THE REPLY.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
```

```
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C
C     HAVE WE TRIED THIS TOO MANY TIMES?
C
      NUMTRY = NUMTRY + 1
      IF (NUMTRY .GT. MAXRTRY) THEN
         SEOF = A
         ABORTYP = TOOMANY.OR.SENDING.OR.EOFERR
         RETURN
      ENDIF
C
C     SEND THE EOF PACKET
C
      CALL SNDPACK(Z,PACKNUM,0,0)
C
C     CHECK THE REPLY
C
      PTYP = RDPACK(LEN,NUM,RECPACK)
      IF (PTYP .EQ. N) THEN
         IF (MOD(PACKNUM+1,64) .NE. NUM) THEN
            SEOF = STATE
            RETURN
         ELSE
            PTYP = Y
            NUM = NUM - 1
         ENDIF
      ENDIF
      IF (PTYP .EQ. Y) THEN
         IF (PACKNUM .NE. NUM) THEN
            SEOF = STATE
            RETURN
         ENDIF
         NUMTRY = 0
         PACKNUM = MOD(PACKNUM+1,64)
         CALL FCLOSE(FFD)
         SEOF = B
      ELSE IF (PTYP .EQ. ERROR) THEN
         SEOF = STATE
      ELSE
```

```
         SEOF = A
         ABORTYP = INVALID.OR.SENDING.OR.EOFERR
      ENDIF
      RETURN
      END
      INTEGER FUNCTION SBREAK()

CCC   SBREAK - SEND THE BREAK PACKET AND WAIT FOR REPLY.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
```

```
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C
C     HAVE WE TRIED THIS TOO MANY TIMES?
C
      NUMTRY = NUMTRY + 1
      IF (NUMTRY .GT. MAXRTRY) THEN
         SBREAK = A
         ABORTYP = TOOMANY.OR.SENDING.OR.BRKERR
         RETURN
      ENDIF
C
C     SEND THE BREAK PACKET
C
      CALL SNDPACK(B,PACKNUM,0,0)
C
C     CHECK ON THE REPLY
C
      PTYP = RDPACK(LEN,NUM,RECPACK)
      IF (PTYP .EQ. N) THEN
         IF (MOD(PACKNUM+1,64) .NE. NUM) THEN
            SBREAK = STATE
            RETURN
         ELSE
            PTYP = Y
            NUM = NUM - 1
         ENDIF
      ENDIF
      IF (PTYP .EQ. Y) THEN
         IF (PACKNUM .NE. NUM) THEN
            SBREAK = STATE
            RETURN
         ENDIF
         NUMTRY = 0
         PACKNUM = MOD(PACKNUM+1,64)
         SBREAK = C
      ELSE IF (PTYP .EQ. ERROR) THEN
         SBREAK = STATE
      ELSE
         SBREAK = A
         ABORTYP = INVALID.OR.SENDING.OR.BRKERR
      ENDIF
      RETURN
      END
      SUBROUTINE SNDPACK(TYPE,NUM,LEN,DATA)

CCC   SNDPACK - SEND A PACKET DOWN AN OUTPUT STREAM
C
C     SNDPACK WILL SEND A PACKET OF INFORMATION AND LOG IT
C     IF DEBUG IS TURNED ON.  THIS SUBROUTINE COULD BE MADE
C     MORE EFFICIENT BY NOT CALLING A SUBROUTINE FOR EACH
C     CHARACTER, BUT THAT MIGHT CAUSE PORTABILITY PROBLEMS.
C
```

```
      IMPLICIT INTEGER (A-Z)
      INTEGER DATA(200)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
```

```
C-----END OF FILE: PROTCOM
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
C$    DO(OT=0)
C
C     DEFINE THE TOCHAR STATEMENT FUNCTION
C
      TOCHAR(ASCCH) = ASCCH + BLANK

      IF ((DEBUG.AND.DBGPACK).NE.0) CALL FPRINTF(DEBUGFD,'↑SENDING...',
     +   0,0,0,0)
C
C     PUT OUT PAD CHARS
C
      DO 100 I = 1,SPAD
         CALL PUTC(SPADCH,OFD)
         IF ((DEBUG.AND.DBGPACK).NE.0) THEN
            CALL PUTC(SPADCH,DEBUGFD)
         ENDIF
100   CONTINUE
      CALL PUTC(SNDSYNC,OFD)
C
C     PACKET LEN ASSUMES ONE CHARACTER CHECKSUMS
C
      CHKSUM = TOCHAR(LEN+3)
      CALL PUTC(CHKSUM,OFD)
      TMP = TOCHAR(NUM)
      CHKSUM = CHKSUM + TMP
      CALL PUTC(TMP,OFD)
      CHKSUM = CHKSUM + TYPE
      CALL PUTC(TYPE,OFD)
      DO 110 I = 1,LEN
         CHKSUM = CHKSUM + (DATA(I) .AND. O"377")
         CALL PUTC(DATA(I),OFD)
110   CONTINUE
      CHKSUM = (CHKSUM + (CHKSUM.AND.O"300") / O"100") .AND. O"77"
      CALL PUTC(TOCHAR(CHKSUM),OFD)
      CALL PUTC(SPEOL,OFD)
      IF ((DEBUG.AND.DBGPACK).NE.0) THEN
         CALL PUTC(SNDSYNC,DEBUGFD)
         CALL PUTC(TOCHAR(LEN+3),DEBUGFD)
         CALL PUTC(TOCHAR(NUM),DEBUGFD)
         CALL PUTC(TYPE,DEBUGFD)
         IF (LEN .GT. 0) CALL PUTSTR(DEBUGFD,DATA)
         CALL PUTC(TOCHAR(CHKSUM),DEBUGFD)
```

```
         CALL PUTC(SPEOL,DEBUGFD)
         CALL PUTC(NEL,DEBUGFD)
      ENDIF
C
C     FORCE BUFFER FLUSH SINCE DESIRED EOL CHAR WON'T
C
      CALL FFLUSH(OFD)
C
C     UPDATE THE STATISTICS
C
      NCH = SPAD + 5 + LEN + 1
      SCHCNT = SCHCNT + NCH
      SCHOVRH = SCHOVRH + NCH - LEN
      RETURN
      END
      INTEGER FUNCTION RDPACK(LEN,NUM,DATA)

CCC   RDPACK - READ A PACKET OF INFORMATION.
C
C     RDPACK WILL READ A PACKET OF DATA AND RETURN THE PACKET TYPE
C     AS A RESULT.  IF THE PACKET CONTAINS AN ERROR (CHECKSUM) THEN
C     ERROR WILL BE RETURNED.  LEN, NUM, AND DATA WILL BE SET ACCORDING
C     TO THE FIELDS OF THE PACKET.
C
      IMPLICIT INTEGER (A-Z)
      INTEGER DATA(200)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
```

```
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
C$    DO(OT=0)
C
C     DEFINE THE UNCHAR STATEMENT FUNCTION
C
      UNCHAR(ASCCH) = ASCCH - BLANK
C
C     IS DEBUG PACKETS TURNED ON?
C
      IF ((DEBUG.AND.DBGPACK).NE.0) THEN
         CALL FPRINTF(DEBUGFD,'↑READING...',0,0,0,0)
      ENDIF
      NCH = 0
C
C     HUNT FOR THE START OF PACKET
C
10    CH = GETC(IFD,CH)
      NCH = NCH + 1
      IF ((DEBUG.AND.DBGPACK).NE.0) CALL PUTC(CH,DEBUGFD)
```

```
      IF (CH .NE. SYNC) GO TO 10
      CHKSUM = 0
      LEN = 0
C
C     PARSE EACH FIELD OF THE PACKET
C
C     FOR (FIELD=1; FIELD <= 5; FIELD++)
      FIELD = 1
20    IF (FIELD .LE. 5) THEN
C
C        A CHARACTER READ IN FIELD 4 HERE IS THE FIRST CHAR OF THE
C        DATA FIELD OR THE CHECKSUM CHARACTER IF THE DATA FIELD IS EMPTY
C
         IF (FIELD .NE. 5 .OR. LEN .GT. 0) THEN
            IF (GETC(IFD,CH) .EQ. SYNC) FIELD = 0
            NCH = NCH + 1
            IF ((DEBUG.AND.DBGPACK).NE.0) CALL PUTC(CH,DEBUGFD)
         ENDIF
         IF (FIELD .LE. 3) CHKSUM = CHKSUM + CH
C
C        IF RESYNC
         IF (FIELD .EQ. 0) THEN
            CHKSUM = 0
            IF ((DEBUG.AND.DBGPACK).NE.0) THEN
               CALL FPRINTF(DEBUGFD,'\N↑READING...%C',SYNC,0,0,0)
            ENDIF
C
C        IF DATA LENGTH
         ELSE IF (FIELD .EQ. 1) THEN
            LEN = UNCHAR(CH-3)
C
C        IF PACKET NUMBER
         ELSE IF (FIELD .EQ. 2) THEN
            NUM = UNCHAR(CH)
C
C        IF PACKET TYPE
         ELSE IF (FIELD .EQ. 3) THEN
            TYPE = CH
C
C        IF DATA FIELD IS NOT EMPTY
         ELSE IF (FIELD .EQ. 4 .AND. LEN .GT. 0) THEN
C
C           READ 2ND-LEN CHARS OF DATA & CHECKSUM CHAR
C
            DO 100 I = 1,LEN
               IF (I .GT. 1) THEN
                  CH = GETC(IFD,CH)
                  NCH = NCH + 1
                  IF (CH .EQ. SYNC) THEN
                     FIELD = 0
                     GO TO 20
                  ENDIF
                  IF ((DEBUG.AND.DBGPACK).NE.0) CALL PUTC(CH,DEBUGFD)
               ENDIF
               CHKSUM = CHKSUM + CH
               DATA(I) = CH
100         CONTINUE
C
C        IF CHKSUM CHAR
         ELSE IF (FIELD .EQ. 5) THEN
```

```
            DATA(LEN+1) = 0
            CHKSUM = (CHKSUM + ((CHKSUM .AND. O"300") / O"100"))
     +                .AND. O"77"
         ENDIF
C
C     PROCESS NEXT PACKET FIELD
C
         FIELD = FIELD + 1
         GO TO 20
      ENDIF
      IF ((DEBUG.AND.DBGPACK).NE.0) CALL PUTC(NEL,DEBUGFD)
C
C     DOES THE CHECKSUM MATCH?
C
      IF (CHKSUM .NE. UNCHAR(CH)) THEN
         RDPACK = ERROR
         RCHOVRH = RCHOVRH + NCH
         IF (DEBUG .NE. 0) THEN
            CALL FPRINTF(DEBUGFD,'CHKSUM ERROR, FOUND %D NEEDED %D\N',
     +                   UNCHAR(CH),CHKSUM,0,0)
         ENDIF
      ELSE
         RDPACK = TYPE
         RCHOVRH = RCHOVRH + NCH - LEN
      ENDIF
      RCHCNT = RCHCNT + NCH
C
C     FLUSH ANY END-OF-LINE CHARACTERS AND OTHER GARBAGE
C
      CALL FFLUSH(IFD)
      RETURN
      END
      INTEGER FUNCTION BUFFILL(FD,BUFFER)

CCC   BUFFILL - GET SOME DATA TO SEND.
C
C     BUFFILL READS FROM THE FILE TO SEND AND PERFORMS ALL
C     THE PROPER ESCAPING OF CONTROL CHARACTERS AND MAPPING
C     NEWLINES INTO CRLF SEQUENCES.  IF IT EVER GETS SMART
C     ENOUGH, IT WILL ALSO DO THE 8 BIT QUOTING AND REPEAT
C     COUNTS.
C
C *** NOTE: THIS ALGORITHM ASSUMES 5 OVERHEAD CHARACTERS FOR THE
C     PACKET AND LEAVES 3 CHARACTERS IN CASE THE LAST CHARACTER
C     TO BUFFER IS A NEL (EXPANDS TO 4 CHARACTERS).
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
```

```
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
      INTEGER BUFFER(100)
C
C     DEFINE CTL STATEMENT FUNCTION
C
      CTL(ASCCH) = ASCCH .XOR. O"100"
C
C     GET A PACKET WORTH OF DATA
C
      I = 0
10    IF (GETC(FD,CH) .NE. EOF) THEN
```

```
         IF (CH .LT. BLANK .OR. CH .EQ. DEL .OR. CH .EQ. NEL .OR.
     +       CH .EQ. SPQUOTE) THEN
            IF (CH .EQ. NEL) THEN
               BUFFER(I+1) = SPQUOTE
               BUFFER(I+2) = CTL(CR)
               I = I + 2
               CH = LF
            ENDIF
            I = I + 1
            BUFFER(I) = SPQUOTE
            IF (CH .NE. SPQUOTE) CH = CTL(CH)
         ENDIF
         I = I + 1
         BUFFER(I) = CH
         IF (I .GE. SPKSIZ-8) THEN
            BUFFILL = I
            GO TO 99
         ENDIF
         GO TO 10
      ENDIF
      IF (I .EQ. 0) THEN
         BUFFILL = EOF
      ELSE
         BUFFILL = I
      ENDIF
99    BUFFER(I+1) = 0
      RETURN
      END
      SUBROUTINE BUFEMP(BUFFER,FD,LEN)

CCC   BUFEMP - DUMP A BUFFER TO A FILE.
C
      IMPLICIT INTEGER (A-Z)
      INTEGER BUFFER(100)
      INTEGER PREVCH
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
```

```
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
C
C     DEFINE CTL STATEMENT FUNCTION
C
      CTL(ASCCH) = ASCCH .XOR. O"100"
C
C     WRITE THE PACKET DATA TO THE FILE
C
      I = 1
10    IF (I .LE. LEN) THEN
         CH = BUFFER(I)
         IF (CH .EQ. QUOTECH) THEN
            I = I + 1
            CH = BUFFER(I)
            IF (CH .NE. QUOTECH) CH = CTL(CH)
         ENDIF
C
C     CONVERT CR/LF PAIR TO NEL (205B)
C
         IF (CH .EQ. LF .AND. PREVCH .EQ. CR) THEN
            CH = NEL
C
C     JUST A LONE CR
```

```
C
         ELSE IF (PREVCH .EQ. CR) THEN
            CALL PUTC(PREVCH,FD)
         ENDIF
         IF (CH .NE. CR) CALL PUTC(CH,FD)
         PREVCH = CH
         I = I + 1
         GO TO 10
      ENDIF
      RETURN
      END
      INTEGER FUNCTION FOPEN(FN,MODE)

CCC   FOPEN - PRETEND TO OPEN A FILE FOR I/O.
C
C     FOPEN JUST ASSIGNS A FILE DESCIPTOR (INTEGER INDEX) TO
C     A FILE NAME.  NO OPENING OF THE FILE IS REALLY PERFORMED
C     SINCE THIS IS DONE AUTOMATICALLY BY IOP.
C
      IMPLICIT INTEGER (A-Z)
      CHARACTER*10 FN
      LOGICAL LOCAL
C-----FILE: C167,KERMSIT
C
C     KERMIT SITE DEFINITIONS:
C
      PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
C-----FILE: C167,FILECOM
C
C     FILE I/O COMMON BLOCK DEFINITIONS
C
C     THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C     AND BEFORE ANY DATA DECLARATIONS.
C
      PARAMETER (MAXFILE=5, MAXWD=512)
      PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
      PARAMETER (ERROR=-2)
      PARAMETER (TAB=9, LF=10, CR=13, NEL=0"205")
      PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
      PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
      COMMON /FILEIOC/ FNAME(MAXFILE)
      CHARACTER*10 FNAME
      COMMON /FILEIO/ FMODE(MAXFILE),
     +   FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +   FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +   BINMODE, PARITY, DUPLEX
      INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
      LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
      DATA FMODE / MAXFILE*CLOSED /
      DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
      DATA RAWMODE, BINMODE / 2*.FALSE. /
      DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM
C
C     CHECK FOR VALID PARAMETERS
C
      IF (MODE .LT. RD .OR. MODE .GT. CREATE) THEN
         CALL DISPLA(' FOPEN - INVALID MODE ',MODE)
         CALL ABORT
```

```
      ENDIF
C
C     FIND THE NEXT UNUSED ENTRY
C
      DO 100 I = 1, MAXFILE
C
C        IF UNUSED TABLE ENTRY IS FOUND
C
         IF (FMODE(I) .EQ. CLOSED) THEN
            FNAME(I) = FN
            FWPTR(I) = 1
            FNWDS(I) = 0
            IF (MODE .EQ. RD) THEN
               FWSHFT(I) = 12
            ELSE
               FWSHFT(I) = 0
            ENDIF
            IF (MODE .EQ. CREATE) THEN
               INQUIRE(FILE=FN,EXIST=LOCAL)
               IF (LOCAL) THEN
                  FMODE(I) = CLOSED
                  FOPEN = ERROR
                  RETURN
               ENDIF
               FMODE(I) = WR
            ELSE
               FMODE(I) = MODE
            ENDIF
            FEOF(I) = .FALSE.
            CTDEV(I) = .FALSE.
            FOPEN = I
C
C           IF STANDARD I/O FILES, CONNECT THEM TO THE TERMINAL.
C
            IF (FN .EQ. 'STDIN' .OR. FN .EQ. 'STDOUT') THEN
               OPEN(UNIT=1,FILE=FN)
               CALL CONNEC(1,1)
               CTDEV(I) = .TRUE.
               CLOSE(UNIT=1)
            ELSE IF (FN .EQ. 'TTY') THEN
               CTDEV(I) = .TRUE.
            ENDIF
C
C        SET THE ASCII FLAG AND REWIND THE FILE.
C
C$          IF (UT2D .EQ. 1)
            CALL IOP('ASC',FN)
            CALL IOP('REW',FN)
C$          ENDIF
            RETURN
C
C        IF TABLE ENTRY FILE NAME MATCHES FN
C
         ELSE IF (FNAME(I) .EQ. FN) THEN
            CALL REMARK(' FOPEN - FILE ' // FN // ' ALREADY OPEN.')
            CALL ABORT
         ENDIF
100   CONTINUE
      CALL REMARK(' FOPEN - TOO MANY FILES OPEN.')
      CALL ABORT
```

```
      RETURN
      END
      SUBROUTINE FCLOSE(FD)

CCC   FCLOSE - REMOVE AN FD FROM THE ACTIVE LIST.
C
C     FCLOSE WILL REMOVE THE FD FROM THE ACTIVE LIST FOR
C     ALLOCATION AT A LATER DATE.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMSIT
C
C     KERMIT SITE DEFINITIONS:
C
      PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
C-----FILE: C167,FILECOM
C
C     FILE I/O COMMON BLOCK DEFINITIONS
C
C     THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C     AND BEFORE ANY DATA DECLARATIONS.
C
      PARAMETER (MAXFILE=5, MAXWD=512)
      PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
      PARAMETER (ERROR=-2)
      PARAMETER (TAB=9, LF=10, CR=13, NEL=O"205")
      PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
      PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
      COMMON /FILEIOC/ FNAME(MAXFILE)
      CHARACTER*10 FNAME
      COMMON /FILEIO/ FMODE(MAXFILE),
     +   FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +   FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +   BINMODE, PARITY, DUPLEX
      INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
      LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
      DATA FMODE / MAXFILE*CLOSED /
      DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
      DATA RAWMODE, BINMODE / 2*.FALSE. /
      DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM

      IF (FD .LT. 1 .OR. FD .GT. MAXFILE) THEN
         CALL DISPLA(' FCLOSE - INVALID FD ',FD)
         CALL ABORT
      ENDIF
      IF (FMODE(FD) .EQ. 0) THEN
         CALL DISPLA(' FCLOSE - FD NOT OPEN.',FD)
         RETURN
      ENDIF
C
C     FORCE EMPTYING OF THE BUFFER
C
      CALL FFLUSH(FD)
C
C     WRITE A FILE MARK
C
C$    IF (UT2D .EQ. 1)
      IF (FMODE(FD) .EQ. WR) CALL IOP('WF',FNAME(FD))
```

```
C$    ENDIF
      FMODE(FD) = CLOSED
      RETURN
      END
      SUBROUTINE FFLUSH(FD)

CCC   FFLUSH - FLUSH AN I/O BUFFER.
C
C     FFLUSH WILL FLUSH THE ASCII STRING BUFFER FOR A PARTICULAR
C     FILE DESCRIPTOR.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMSIT
C
C     KERMIT SITE DEFINITIONS:
C
      PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
C-----FILE: C167,FILECOM
C
C     FILE I/O COMMON BLOCK DEFINITIONS
C
C     THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C     AND BEFORE ANY DATA DECLARATIONS.
C
      PARAMETER (MAXFILE=5, MAXWD=512)
      PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
      PARAMETER (ERROR=-2)
      PARAMETER (TAB=9, LF=10, CR=13, NEL=O"205")
      PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
      PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
      COMMON /FILEIOC/ FNAME(MAXFILE)
      CHARACTER*10 FNAME
      COMMON /FILEIO/ FMODE(MAXFILE),
     +   FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +   FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +   BINMODE, PARITY, DUPLEX
      INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
      LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
      DATA FMODE / MAXFILE*CLOSED /
      DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
      DATA RAWMODE, BINMODE / 2*.FALSE. /
      DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM
C
C     IS THE FD VALID?
C
      IF (FD .LT. 1 .OR. FD .GT. MAXFILE) THEN
         CALL DISPLA(' FFLUSH - INVALID FILE DESCRIPTOR',FD)
         CALL ABORT
      ENDIF
      IF (FMODE(FD) .EQ. 0) THEN
         CALL DISPLA(' FFLUSH - FILE DESCRIPTOR NOT OPEN',FD)
         CALL ABORT
      ENDIF
C
C     IF FD WAS OPENED AS WRITE, THEN FLUSH TO THE FILE
C
      IF (FMODE(FD) .EQ. WR) THEN
C$       IF (UT2D .EQ. 1)
```

```
         CALL IOP('WB',FNAME(FD),FCHBUF(1,FD),FNWDS(FD))
C$       ENDIF
      ELSE IF (FMODE(FD) .EQ. RD) THEN
C$       IF (UT2D .EQ. 1)
         CALL IOP('REW',FNAME(FD),0,0)
C$       ENDIF
      ENDIF
C
C     RESET BUFFER CHARACTER COUNT
C
      FWPTR(FD) = 1
      FNWDS(FD) = 0
      IF (FMODE(FD) .EQ. RD) THEN
         FWSHFT(FD) = 12
      ELSE
         FWSHFT(FD) = 0
      ENDIF
      RETURN
      END
      INTEGER FUNCTION GETC(FD,CH)

CCC   GETC - RETURN NEXT CHARACTER FROM THE INPUT STREAM.
C
C     GETC WILL RETURN THE NEXT ASCII CHARACTER THAT WAS
C     READ FROM THE FILE DESCRIPTOR FD.  READS ARE BUFFERED
C     WITH 5 CHARACTERS PACKED TO A WORD.  O"0000" BYTES
C     ARE IGNORED.  NULS ARE O"4000" BYTES.  EOF (-1) IS
C     RETURNED WHEN EOF IS READ.
C
      IMPLICIT INTEGER (A-Z)
      LOGICAL CLREOR
C-----FILE: C167,KERMSIT
C
C     KERMIT SITE DEFINITIONS:
C
      PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
C-----FILE: C167,FILECOM
C
C     FILE I/O COMMON BLOCK DEFINITIONS
C
C     THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C     AND BEFORE ANY DATA DECLARATIONS.
C
      PARAMETER (MAXFILE=5, MAXWD=512)
      PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
      PARAMETER (ERROR=-2)
      PARAMETER (TAB=9, LF=10, CR=13, NEL=O"205")
      PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
      PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
      COMMON /FILEIOC/ FNAME(MAXFILE)
      CHARACTER*10 FNAME
      COMMON /FILEIO/ FMODE(MAXFILE),
     +   FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +   FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +   BINMODE, PARITY, DUPLEX
      INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
      LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
      DATA FMODE / MAXFILE*CLOSED /
      DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
```

```
      DATA RAWMODE, BINMODE / 2*.FALSE. /
      DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM
C
C     IS THE FD VALID?
C
      IF (FD .LT. 1 .OR. FD .GT. MAXFILE) THEN
         CALL DISPLA(' GETC - INVALID FILE DESCRIPTOR',FD)
         CALL ABORT
      ENDIF
      IF (FMODE(FD) .EQ. 0) THEN
         CALL DISPLA(' GETC - FILE DESCRIPTOR NOT OPEN',FD)
         CALL ABORT
      ENDIF
C
C     CHECK IF OK TO READ
C
      IF ((FMODE(FD).AND.RD) .NE. RD) THEN
         CALL DISPLA(' GETC - READ ON WRITE-ONLY FILE ',FD)
         CALL ABORT
      ENDIF
C
C     CHECK IF MORE DATA NEEDED
C
10    IF (FWPTR(FD) .GT. FNWDS(FD)) THEN
         IF (FEOF(FD)) THEN
            C = EOF
            GETC = C
            RETURN
         ENDIF
C
C        GET A BUFFER WORTH OF DATA
C
C$       IF (UT2D .EQ. 1)
         IF (CTDEV(FD) .AND. (.NOT. RAWMODE)) THEN
            NATTEMP = MAXWD
C
C           CLEAR THE INPUT BUFFER
C
            DO 100 NREAD = 1, NATTEMP
               FCHBUF(NREAD,FD) = 0
100         CONTINUE
            NREAD = NATTEMP - IOP('RD',FNAME(FD),FCHBUF(1,FD),NATTEMP)
            IF (NREAD .NE. NATTEMP) THEN
               CLREOR = .TRUE.
            ELSE
               CLREOR = .FALSE.
            ENDIF
C
C           FIND TRUE NUMBER OF WORDS IN BUFFER
C
            NREAD = 0
15          NREAD = NREAD + 1
            IF (FCHBUF(NREAD,FD).NE.0 .AND. NREAD.LT.NATTEMP) GO TO 15
            IF (FCHBUF(NREAD,FD).EQ.0 .AND. NREAD.GT.1) NREAD = NREAD-1
         ELSE
            NATTEMP = MAXWD
            NREAD = NATTEMP - IOP('RB',FNAME(FD),FCHBUF(1,FD),NATTEMP)
            IF (NREAD .LT. NATTEMP) THEN
               CLREOR = .TRUE.
```

```
            ELSE
               CLREOR = .FALSE.
            ENDIF
         ENDIF
         IF (CLREOR) THEN
            IF (IOP('RR',FNAME(FD),0,0) .NE. 0) THEN
               FOO = IOP('RF',FNAME(FD),0,0)
               IF ((.NOT.RAWMODE) .OR. (RAWMODE.AND.(.NOT.CTDEV(FD))))
     +            THEN
                  FEOF(FD) = .TRUE.
                  IF (NREAD .EQ. 0) THEN
                     C = EOF
                     GETC = C
                     RETURN
                  ENDIF
               ENDIF
            ENDIF
         ENDIF
C$       ENDIF
         FWPTR(FD) = 1
         FNWDS(FD) = NREAD
         FWSHFT(FD) = 12
      ENDIF
C
C     PICKUP CHAR TO RETURN AND CHECK FOR IGNORED O"0000" BYTE
C
      CH = SHIFT(FCHBUF(FWPTR(FD),FD),FWSHFT(FD)) .AND. O"7777"
      FWSHFT(FD) = FWSHFT(FD) + 12
      IF (FWSHFT(FD) .GT. 60) THEN
         FWSHFT(FD) = 12
         FWPTR(FD) = FWPTR(FD) + 1
      ENDIF
C
C     REMOVE PARITY BIT IF NECESSARY
C
      IF (BINMODE .OR. RAWMODE) THEN
         IF (PARITY .NE. NOPAR .AND. CTDEV(FD)) CH = CH .AND. O"0177"
         IF (CH .EQ. CR .OR. CH .EQ. LF) CH = NEL
      ENDIF
      IF (CH .EQ. 0) GO TO 10
      GETC = CH
      RETURN
      END
      SUBROUTINE UNGETC(FD,CH)

CCC   UNGETC - TRY TO PUT A CHARACTER BACK INTO THE INPUT STREAM.
C
C     UNGETC CAN ONLY PUT BACK CHARACTERS AS FAR AS THE BEGINNING
C     OF THE BUFFER.  HOPEFULLY, THIS IS OK, SINCE ONLY GETWORD
C     DOES THIS WITH AN NEL WHICH SHOULD BE WELL INTO THE BUFFER.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,FILECOM
C
C     FILE I/O COMMON BLOCK DEFINITIONS
C
C     THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C     AND BEFORE ANY DATA DECLARATIONS.
C
      PARAMETER (MAXFILE=5, MAXWD=512)
```

```
      PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
      PARAMETER (ERROR=-2)
      PARAMETER (TAB=9, LF=10, CR=13, NEL=O"205")
      PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
      PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
      COMMON /FILEIOC/ FNAME(MAXFILE)
      CHARACTER*10 FNAME
      COMMON /FILEIO/ FMODE(MAXFILE),
     +   FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +   FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +   BINMODE, PARITY, DUPLEX
      INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
      LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
      DATA FMODE / MAXFILE*CLOSED /
      DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
      DATA RAWMODE, BINMODE / 2*.FALSE. /
      DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM
C
C     IS IT OK TO BACK UP THE POINTER?
C
      IF (FWSHFT(FD) .EQ. 12 .AND. FWPTR(FD) .EQ. 1) THEN
         CALL DISPLA('UNGETC - CANNOT PUSH CHARACTER ',CH)
         RETURN
      ENDIF
C
C     BACK UP THE POINTER
C
      IF (FWSHFT(FD) .EQ. 12) THEN
         FWSHFT(FD) = 60
         FWPTR(FD) = FWPTR(FD) - 1
      ELSE
         FWSHFT(FD) = FWSHFT(FD) - 12
      ENDIF
      FCHBUF(FWPTR(FD),FD) = (FCHBUF(FWPTR(FD),FD) .AND. SHIFT(
     +   O"7777",60-FWSHFT(FD))) .OR. SHIFT(CH,60-FWSHFT(FD))
      RETURN
      END
      INTEGER FUNCTION GETWORD(FD,STR,MAXLEN)

CCC   GETWORD - GET A WORD FROM AN INPUT STREAM.
C
C     GETWORD CONSIDERS A WORD TO BE DELIMITED BY BLANKS.
C     IT WILL RETURN THE LENGTH OF THE WORD AS ITS VALUE.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
```

```
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
      INTEGER STR(200)

      LEN = 0
C
C     SKIP LEADING WHITE SPACES
C
10    IF (GETC(FD,CH) .EQ. EOF) THEN
         GETWORD = EOF
         RETURN
      ELSE IF (CH .EQ. NEL) THEN
         GETWORD = 0
         RETURN
      ENDIF
      IF (CH .EQ. BLANK .OR. CH .EQ. TAB) GO TO 10
C
C     FOUND THE FIRST CHARACTER, SO KEEP GOING
C
20    IF (LEN .LT. MAXLEN) THEN
         LEN = LEN + 1
         STR(LEN) = CH
      ENDIF
      CH = GETC(FD,CH)
      IF (CH .NE. EOF .AND. CH .NE. BLANK .AND. CH .NE. TAB .AND.
     +    CH .NE. NEL) GO TO 20
C
C     SAVE EOLS FOR NEXT GETWORD
C
      IF (CH .EQ. NEL) CALL UNGETC(FD,CH)
      STR(LEN+1) = 0
      GETWORD = LEN
      RETURN
      END
      SUBROUTINE PUTC(TCH,FD)

CCC   PUTC - PUT A CHARACTER INTO AN OUTPUT STREAM
```

```
C
C     PUTC OUTPUTS A CHARACTER WITH THE PARITY BIT SET TO THE
C     PROPER PARITY IF THE OUTPUT FILE IS CONVERSATIONAL.
C     THE FIVE TYPES OF PARITY ARE DEFINED FOR EACH CHARACTER
C     IN A TABLE.
C
      IMPLICIT INTEGER (A-Z)
      INTEGER CHPARTY(128)
C-----FILE: C167,FILECOM
C
C     FILE I/O COMMON BLOCK DEFINITIONS
C
C     THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C     AND BEFORE ANY DATA DECLARATIONS.
C
      PARAMETER (MAXFILE=5, MAXWD=512)
      PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
      PARAMETER (ERROR=-2)
      PARAMETER (TAB=9, LF=10, CR=13, NEL=O"205")
      PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
      PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
      COMMON /FILEIOC/ FNAME(MAXFILE)
      CHARACTER*10 FNAME
      COMMON /FILEIO/ FMODE(MAXFILE),
     +   FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +   FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +   BINMODE, PARITY, DUPLEX
      INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
      LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
      DATA FMODE / MAXFILE*CLOSED /
      DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
      DATA RAWMODE, BINMODE / 2*.FALSE. /
      DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM
      DATA (CHPARTY(I),I=1,38) /
     +              O"40000200020040004000", O"02010001020100010001",
C                           NUL                          SOH
     +              O"02020002020200020002", O"00030203020300030003",
C                           STX                          ETX
     +              O"02040004020400040004", O"00050205020500050005",
C                           EOT                          ENQ
     +              O"00060206020600060006", O"02070007020700070007",
C                           ACK                          BEL
     +              O"02100010021000100010", O"00110211021100110011",
C                           BS                           HT
     +              O"00120212021200120012", O"02130013021300130013",
C                           LF                           VT
     +              O"00140214021400140014", O"02150015021500150015",
C                           FF                           CR
     +              O"02160016021600160016", O"00170217021700170017",
C                           SO                           SI
     +              O"02200020022000200020", O"00210221022100210021",
C                           DLE                          DC1
     +              O"00220222022200220022", O"02230023022300230023",
C                           DC2                          DC3
     +              O"00240224022400240024", O"02250025022500250025",
C                           DC4                          NAK
     +              O"02260026022600260026", O"00270227022700270027",
C                           SYN                          ETB
     +              O"00300230023000300030", O"02310031023100310031",
```

```
C                                CAN                                EM
     +                   O"02320032023200320032", O"00330233023300330033",
C                                SUB                                ESC
     +                   O"02340034023400340034", O"00350235023500350035",
C                                FS                                 GS
     +                   O"00360236023600360036", O"02370037023700370037",
C                                RS                                 US
     +                   O"02400040024000400040", O"00410241024100410041",
C                                                                   !
     +                   O"00420242024200420042", O"02430043024300430043",
C                                "                                  #
     +                   O"00440244024400440044", O"02450045024500450045" /
C                                $                                  %
      DATA (CHPARTY(I),I=39,76) /
     +                   O"02460046024600460046", O"00470247024700470047",
C                                &                                  '
     +                   O"00500250025000500050", O"02510051025100510051",
C                                (                                  )
     +                   O"02520052025200520052", O"00530253025300530053",
C                                *                                  +
     +                   O"02540054025400540054", O"00550255025500550055",
C                                ,                                  -
     +                   O"00560256025600560056", O"02570057025700570057",
C                                .                                  /
     +                   O"00600260026000600060", O"02610061026100610061",
C                                0                                  1
     +                   O"02620062026200620062", O"00630263026300630063",
C                                2                                  3
     +                   O"02640064026400640064", O"00650265026500650065",
C                                4                                  5
     +                   O"00660266026600660066", O"02670067026700670067",
C                                6                                  7
     +                   O"02700070027000700070", O"00710271027100710071",
C                                8                                  9
     +                   O"00720272027200720072", O"02730073027300730073",
C                                :                                  ;
     +                   O"00740274027400740074", O"02750075027500750075",
C                                <                                  =
     +                   O"02760076027600760076", O"00770277027700770077",
C                                >                                  ?
     +                   O"03000100030001000100", O"01010301030101010101",
C                                @                                  A
     +                   O"01020302030201020102", O"03030103030301030103",
C                                B                                  C
     +                   O"01040304030401040104", O"03050105030501050105",
C                                D                                  E
     +                   O"03060106030601060106", O"01070307030701070107",
C                                F                                  G
     +                   O"01100310031001100110", O"03110111031101110111",
C                                H                                  I
     +                   O"03120112031201120112", O"01130313031301130113" /
C                                J                                  K
      DATA (CHPARTY(I),I=77,114) /
     +                   O"03140114031401140114", O"01150315031501150115",
C                                L                                  M
     +                   O"01160316031601160116", O"03170117031701170117",
C                                N                                  O
     +                   O"01200320032001200120", O"03210121032101210121",
C                                P                                  Q
     +                   O"03220122032201220122", O"01230323032301230123",
```

```
C                                R                                S
     +                   O"03240124032401240124", O"01250325032501250125",
C                                T                                U
     +                   O"01260326032601260126", O"03270127032701270127",
C                                V                                W
     +                   O"03300130033001300130", O"01310331033101310131",
C                                X                                Y
     +                   O"01320332033201320132", O"03330133033301330133",
C                                Z                                [
     +                   O"01340334033401340134", O"03350135033501350135",
C                                \                                ]
     +                   O"03360136033601360136", O"01370337033701370137",
C                                ↑                                UNDERSCORE
     +                   O"01400340034001400140", O"03410141034101410141",
C                                GRAVE ACCENT                     A
     +                   O"03420142034201420142", O"01430343034301430143",
C                                B                                C
     +                   O"03440144034401440144", O"01450345034501450145",
C                                D                                E
     +                   O"01460346034601460146", O"03470147034701470147",
C                                F                                G
     +                   O"03500150035001500150", O"01510351035101510151",
C                                H                                I
     +                   O"01520352035201520152", O"03530153035301530153",
C                                J                                K
     +                   O"01540354035401540154", O"03550155035501550155",
C                                L                                M
     +                   O"03560156035601560156", O"01570357035701570157",
C                                N                                O
     +                   O"03600160036001600160", O"01610361036101610161" /
      DATA (CHPARTY(I),I=115,128) /
C                                P                                Q
     +                   O"01620362036201620162", O"03630163036301630163",
C                                R                                S
     +                   O"01640364036401640164", O"03650165036501650165",
C                                T                                U
     +                   O"03660166036601660166", O"01670367036701670167",
C                                V                                W
     +                   O"01700370037001700170", O"03710171037101710171",
C                                X                                Y
     +                   O"03720172037201720172", O"01730373037301730173",
C                                Z                                LEFT BRACE
     +                   O"03740174037401740174", O"01750375037501750175",
C                                BAR                              RIGHT BRACE
     +                   O"01760376037601760176", O"03770177037701770177" /
C                                TILDE                            DEL
C
C     IS THE FD VALID?
C
      IF (FD .LT. 1 .OR. FD .GT. MAXFILE) THEN
         CALL DISPLA(' PUTC - INVALID FILE DESCRIPTOR',FD)
         CALL ABORT
      ENDIF
      IF (FMODE(FD) .EQ. 0) THEN
         CALL DISPLA(' PUTC - FILE DESCRIPTOR NOT OPEN',FD)
         CALL ABORT
      ENDIF
C
C     IS IT OK TO WRITE ON THIS STREAM?
C
```

```
      IF ((FMODE(FD).AND.WR) .NE. WR) THEN
         CALL DISPLA(' PUTC - WRITE ON READ-ONLY FILE ',FD)
         CALL ABORT
      ENDIF
C
C     IS IT A VALID CHARACTER?
C
C     IF ((CH.AND.MASK(48)) .NE. 0) THEN
C        CALL DISPLA(' PUTC - INVALID ASCII BYTE ',CH)
C        CALL ABORT
C     ENDIF
C
C     ADD ANOTHER CHARACTER TO THE OUTPUT BUFFER
C
      CH = TCH
10    IF (CTDEV(FD)) THEN
         IF (CH .EQ. NEL) CH = CR
         IF (CH .GE. 0 .AND. CH .LT. 128)
     +      CH = SHIFT(CHPARTY(CH+1),PARITY*12) .AND. O"7777"
      ENDIF
      IF (FWSHFT(FD) .EQ. 0) THEN
         IF (FNWDS(FD) .EQ. MAXWD) THEN
            CALL FFLUSH(FD)
         ENDIF
         FWSHFT(FD) = 48
         FNWDS(FD) = FNWDS(FD) + 1
         FCHBUF(FNWDS(FD),FD) = 0
      ELSE
         FWSHFT(FD) = FWSHFT(FD) - 12
      ENDIF
      FCHBUF(FNWDS(FD),FD) = FCHBUF(FNWDS(FD),FD) .OR.
     +   SHIFT(CH,FWSHFT(FD))
      IF (TCH .EQ. NEL .AND. (CH.AND.O"0177") .EQ. CR) THEN
         CH = LF
         GO TO 10
      ENDIF
      IF (TCH .EQ. NEL) CALL FFLUSH(FD)
      RETURN
      END
      SUBROUTINE PUTSTR(FD,STR)

CCC   PUTSTR - OUTPUT A STRING TO AN OUTPUT STREAM.
C
C     PUTSTR WILL ADD CHARACTERS FROM THE NULL TERMINATED CHARACTER
C     BUFFER STR TO THE SPECIFIED OUTPUT STREAM.
C
      IMPLICIT INTEGER (A-Z)
      INTEGER STR(200)
C-----FILE: C167,FILECOM
C
C     FILE I/O COMMON BLOCK DEFINITIONS
C
C     THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C     AND BEFORE ANY DATA DECLARATIONS.
C
      PARAMETER (MAXFILE=5, MAXWD=512)
      PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
      PARAMETER (ERROR=-2)
      PARAMETER (TAB=9, LF=10, CR=13, NEL=O"205")
      PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
```

```
      PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
      COMMON /FILEIOC/ FNAME(MAXFILE)
      CHARACTER*10 FNAME
      COMMON /FILEIO/ FMODE(MAXFILE),
     +   FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +   FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +   BINMODE, PARITY, DUPLEX
      INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
      LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
      DATA FMODE / MAXFILE*CLOSED /
      DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
      DATA RAWMODE, BINMODE / 2*.FALSE. /
      DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM
C
C     IS THE FD VALID?
C
      IF (FD .LT. 1 .OR. FD .GT. MAXFILE) THEN
         CALL DISPLA(' PUTC - INVALID FD ',FD)
         CALL ABORT
      ENDIF
      IF (FMODE(FD) .EQ. 0) THEN
         CALL DISPLA(' PUTC - FD NOT OPEN.',FD)
         RETURN
      ENDIF
C
C     IS IT OK TO WRITE ON THIS STREAM?
C
      IF ((FMODE(FD).AND.WR) .NE. WR) THEN
         CALL DISPLA(' PUTC - WRITE ON READ-ONLY FILE ',FD)
         CALL ABORT
      ENDIF
C
C     PUT CHARS IN THE OUTPUT BUFFER
C
      I = 1
10    IF (STR(I) .NE. 0) THEN
C
C        IS IT A VALID CHARACTER?
C
C        IF ((STR(I).AND.MASK(48)) .NE. 0) THEN
C           CALL DISPLA(' PUTSTR - INVALID ASCII BYTE ',STR(I))
C           CALL ABORT
C        ENDIF
         CALL PUTC(STR(I),FD)
         I = I + 1
         GO TO 10
      ENDIF
      RETURN
      END
      SUBROUTINE PUTINT(FD,INT,MINWID)

CCC   PUTINT - OUTPUT AN INTEGER.
C
      IMPLICIT INTEGER (A-Z)
      INTEGER STRING(21)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
```

```
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF

      WIDTH = 0
      IF (INT .LT. 0) THEN
         CALL PUTC(ASC('-'),FD)
         WIDTH = 1
      ENDIF
      VAL = IABS(INT)
      ASCII0 = ASC('0')
      NCH = 0
10    NCH = NCH + 1
      STRING(NCH) = MOD(VAL,10) + ASCII0
      VAL = VAL / 10
      IF (VAL .NE. 0 .AND. NCH .LT. 20) GO TO 10
      WIDTH = WIDTH + NCH
C
C     NOW OUTPUT THE DIGITS
C
20    CALL PUTC(STRING(NCH),FD)
      NCH = NCH - 1
      IF (NCH .GT. 0) GO TO 20
30    IF (WIDTH .LT. MINWID) THEN
         CALL PUTC(BLANK,FD)
         WIDTH = WIDTH + 1
         GO TO 30
      ENDIF
```

```
      RETURN
      END
      SUBROUTINE PUTDAY(FD,MM,DD,YY)

CCC   OUTPUT DAY OF WEEK.
C
      IMPLICIT INTEGER (A-Z)
      IZLR(IYR,M,IDY)=MOD((13*(M+10-(M+10)/13*12)-1)/5+IDY+77
     1 +5*(IYR+(M-14)/12-(IYR+(M-14)/12)/100*100)/4
     2 +(IYR+(M-14)/12)/400-(IYR+(M-14)/12)/100*2,7)+1
      WKDAY = IZLR(YY,MM,DD)
      IF (WKDAY .EQ. 1) THEN
         CALL FPRINTF(FD,'↑SUNDAY')
      ELSE IF (WKDAY .EQ. 2) THEN
         CALL FPRINTF(FD,'↑MONDAY')
      ELSE IF (WKDAY .EQ. 3) THEN
         CALL FPRINTF(FD,'↑TUESDAY')
      ELSE IF (WKDAY .EQ. 4) THEN
         CALL FPRINTF(FD,'↑WEDNESDAY')
      ELSE IF (WKDAY .EQ. 5) THEN
         CALL FPRINTF(FD,'↑THURSDAY')
      ELSE IF (WKDAY .EQ. 6) THEN
         CALL FPRINTF(FD,'↑FRIDAY')
      ELSE
         CALL FPRINTF(FD,'↑SATURDAY')
      ENDIF
      RETURN
      END
      SUBROUTINE PUTMNTH(FD,MM)

CCC   PUTMNTH - OUTPUT THE MONTH NAME.
C
      IMPLICIT INTEGER (A-Z)
      IF (MM .EQ. 1) THEN
         CALL FPRINTF(FD,'↑JANUARY',0)
      ELSE IF (MM .EQ. 2) THEN
         CALL FPRINTF(FD,'↑FEBRUARY',0)
      ELSE IF (MM .EQ. 3) THEN
         CALL FPRINTF(FD,'↑MARCH',0)
      ELSE IF (MM .EQ. 4) THEN
         CALL FPRINTF(FD,'↑APRIL',0)
      ELSE IF (MM .EQ. 5) THEN
         CALL FPRINTF(FD,'↑MAY',0)
      ELSE IF (MM .EQ. 6) THEN
         CALL FPRINTF(FD,'↑JUNE',0)
      ELSE IF (MM .EQ. 7) THEN
         CALL FPRINTF(FD,'↑JULY',0)
      ELSE IF (MM .EQ. 8) THEN
         CALL FPRINTF(FD,'↑AUGUST',0)
      ELSE IF (MM .EQ. 9) THEN
         CALL FPRINTF(FD,'↑SEPTEMBER',0)
      ELSE IF (MM .EQ. 10) THEN
         CALL FPRINTF(FD,'↑OCTOBER',0)
      ELSE IF (MM .EQ. 11) THEN
         CALL FPRINTF(FD,'↑NOVEMBER',0)
      ELSE IF (MM .EQ. 12) THEN
         CALL FPRINTF(FD,'↑DECEMBER',0)
      ELSE
         CALL FPRINTF(FD,'PUTMNTH - NO SUCH MONTH AS %D\N',MM)
      ENDIF
```

```
      RETURN
      END
      SUBROUTINE FPRINTF(FD,FMT,I1,I2,I3,I4)

CCC   FPRINTF - POOR ATTEMPT AT FORMATTED ASCII OUTPUT.
C
C     CONVERSION IS SIMILAR TO FPRINTF USED IN C.  SUPPORTED
C     CONVERSIONS ARE %D (INTEGER), %C (ASCII CHARACTER), %S (ASCII
C     STRING BUFFER).  A \N WILL MAP TO A NEWLINE, A \T WILL
C     WILL MAP TO A TAB, A \O WILL TERMINATE THE FORMAT SCANNING.
C     A \ FOLLOWED BY ANY OTHER CHARACTER WILL CAUSE THAT CHARACTER
C     TO BE OUTPUT.  THE DEFAULT OUTPUT CASE WILL BE LOWERCASE.
C     A ↑ FOLLOWED BY A LETTER WILL CAUSE THAT CHARACTER TO BE OUTPUT
C     AS UPPERCASE.  A %D CONVERSION MAY NOW SPECIFY A MINIMUM FIELD
C     WIDTH AS %<N>D (I.E. %10D) IN WHICH THE NUMBER WILL BE BLANK
C     PADDED TO THE RIGHT TO USE UP <N> CHARACTERS.
C
      IMPLICIT INTEGER (A-Z)
      CHARACTER*(*) FMT
C-----FILE: C167,FILECOM
C
C     FILE I/O COMMON BLOCK DEFINITIONS
C
C     THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C     AND BEFORE ANY DATA DECLARATIONS.
C
      PARAMETER (MAXFILE=5, MAXWD=512)
      PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
      PARAMETER (ERROR=-2)
      PARAMETER (TAB=9, LF=10, CR=13, NEL=O"205")
      PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
      PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
      COMMON /FILEIOC/ FNAME(MAXFILE)
      CHARACTER*10 FNAME
      COMMON /FILEIO/ FMODE(MAXFILE),
     +   FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +   FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +   BINMODE, PARITY, DUPLEX
      INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
      LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
      DATA FMODE / MAXFILE*CLOSED /
      DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
      DATA RAWMODE, BINMODE / 2*.FALSE. /
      DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM
C
C     IS THE FD VALID?
C
      IF (FD .LT. 1 .OR. FD .GT. MAXFILE) THEN
         CALL DISPLA(' FPRINTF - INVALID FD ',FD)
         CALL ABORT
      ENDIF
      IF (FMODE(FD) .EQ. 0) THEN
         CALL DISPLA(' FPRINTF - FD NOT OPEN.',FD)
         RETURN
      ENDIF
C
C     IS IT OK TO WRITE ON THIS STREAM?
C
      IF ((FMODE(FD).AND.WR) .NE. WR) THEN
```

```
         CALL DISPLA(' FPRINTF - WRITE ON READ-ONLY FILE ',FD)
         CALL ABORT
      ENDIF
C
C     NOW CALL THE REAL FPRINTF WORKHORSE
C
      CALL DOPRNT(FD,0,1,FMT,I1,I2,I3,I4)
      RETURN
      END
      SUBROUTINE SPRINTF(STR,FMT,I1,I2,I3,I4)

CCC   SPRINTF - POOR ATTEMPT AT DOING INTERNAL FORMATTED I/O.
C
C     SPRINTF IS THE SAME AS FPRINTF EXCEPT THAT IT WRITES TO
C     AND ASCII STRING BUFFER INSTEAD.
C
      IMPLICIT INTEGER (A-Z)
      CHARACTER*(*) FMT
C
C     CALL THE REAL SPRINTF WORKHORSE
C
      CALL DOPRNT(0,STR,2,FMT,I1,I2,I3,I4)
      RETURN
      END
      SUBROUTINE DOPRNT(FD,STRNG,PTYP,FMT,I1,I2,I3,I4)

CCC   DOPRNT - WORKHORSE FOR FORMATTED ASCII I/O.
C
C     CONVERSION IS SIMILAR TO FPRINTF USED IN C.  SUPPORTED
C     CONVERSIONS ARE %D (INTEGER), %C (ASCII CHARACTER), %S (ASCII
C     STRING BUFFER).  A \N WILL MAP TO A NEWLINE, A \T WILL
C     WILL MAP TO A TAB, A \O WILL TERMINATE THE FORMAT SCANNING.
C     A \ FOLLOWED BY ANY OTHER CHARACTER WILL CAUSE THAT CHARACTER
C     TO BE OUTPUT.  THE DEFAULT OUTPUT CASE WILL BE LOWERCASE.
C     A ↑ FOLLOWED BY A LETTER WILL CAUSE THAT CHARACTER TO BE OUTPUT
C     AS UPPERCASE.  A %D CONVERSION MAY NOW SPECIFY A MINIMUM FIELD
C     WIDTH AS %<N>D (I.E. %10D) IN WHICH THE NUMBER WILL BE BLANK
C     PADDED TO THE RIGHT TO USE UP <N> CHARACTERS.
C
      IMPLICIT INTEGER (A-Z)
      CHARACTER*(*) FMT
      INTEGER STRNG(200)
      INTEGER STR(21)
      CHARACTER*1 CH
C-----FILE: C167,FILECOM
C
C     FILE I/O COMMON BLOCK DEFINITIONS
C
C     THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C     AND BEFORE ANY DATA DECLARATIONS.
C
      PARAMETER (MAXFILE=5, MAXWD=512)
      PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
      PARAMETER (ERROR=-2)
      PARAMETER (TAB=9, LF=10, CR=13, NEL=O"205")
      PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
      PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
      COMMON /FILEIOC/ FNAME(MAXFILE)
      CHARACTER*10 FNAME
      COMMON /FILEIO/ FMODE(MAXFILE),
```

```
     +   FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +   FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +   BINMODE, PARITY, DUPLEX
      INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
      LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
      DATA FMODE / MAXFILE*CLOSED /
      DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
      DATA RAWMODE, BINMODE / 2*.FALSE. /
      DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM
C
C     CHECK FOR FILE OR STRING WRITE
C
      IF (PTYP .NE. 1 .AND. PTYP .NE. 2) THEN
         CALL DISPLA(' DOPRNT - INVALID WRITE FUNCTION',PTYP)
         CALL ABORT
      ENDIF
C
C     OUTPUT THE FORMATTED STRING
C
      IPTR = 1
      OPTR = 1
      FPTR = 1
      FMTLEN = LEN(FMT)
10    IF (FPTR .LE. FMTLEN) THEN
         CH = FMT(FPTR:FPTR)
         IF (CH .NE. '\' .AND. CH .NE. '%' .AND. CH .NE. '↑') THEN
            IF (PTYP .EQ. 1) THEN
               CALL PUTC(ASC(CH),FD)
            ELSE
               STRNG(OPTR) = ASC(CH)
               OPTR = OPTR + 1
            ENDIF
C
C        IS IT A QUOTE OR SPECIAL SEQUENCE CHARACTER?
C
         ELSE IF (CH .EQ. '\') THEN
            FPTR = FPTR+1
            CH = FMT(FPTR:FPTR)
            IF (CH .EQ. 'N' .AND. PTYP .EQ. 1) THEN
               CALL PUTC(NEL,FD)
            ELSE IF (CH .EQ. 'T' .AND. PTYP .EQ. 1) THEN
               CALL PUTC(TAB,FD)
            ELSE IF (CH .EQ. 'O') THEN
               IF (PTYP .EQ. 2) STRNG(OPTR) = 0
               RETURN
            ELSE IF (CH .EQ. 'N') THEN
               STRNG(OPTR) = NEL
               OPTR = OPTR + 1
            ELSE IF (CH .EQ. 'T') THEN
               STRNG(OPTR) = TAB
               OPTR = OPTR + 1
            ELSE
               IF (PTYP .EQ. 1) THEN
                  CALL PUTC(ASC(CH),FD)
               ELSE
                  STRNG(OPTR) = ASC(CH)
                  OPTR = OPTR + 1
               ENDIF
            ENDIF
```

```
C
C     IS IT AN UPPERCASE MAPPING?
C
         ELSE IF (CH .EQ. '↑') THEN
            FPTR = FPTR + 1
            CH = FMT(FPTR:FPTR)
            IF (CH .GE. 'A' .AND. CH .LE. 'Z') THEN
               ACH = ASC(CH)-32
            ELSE
               ACH = ASC(CH)
            ENDIF
            IF (PTYP .EQ. 1) THEN
               CALL PUTC(ACH,FD)
            ELSE
               STRNG(OPTR) = ACH
               OPTR = OPTR + 1
            ENDIF
C
C        MUST BE A CONVERSION (%)
C
         ELSE
            INTWDTH = 1
            FPTR = FPTR + 1
            CH = FMT(FPTR:FPTR)
C
C           IS IT AN INTEGER VALUE FORMAT SPEC?
C
20          IF (CH .EQ. 'D') THEN
               IF (IPTR .EQ. 1) THEN
                  ACH = I1
               ELSE IF (IPTR .EQ. 2) THEN
                  ACH = I2
               ELSE IF (IPTR .EQ. 3) THEN
                  ACH = I3
               ELSE
                  ACH = I4
               ENDIF
               IF (PTYP .EQ. 1) THEN
                  CALL PUTINT(FD,ACH,INTWDTH)
               ELSE
                  TLEN = ITOS(ACH,STRNG(OPTR),INTWDTH)
                  OPTR = OPTR + TLEN
               ENDIF
               IPTR = IPTR + 1
C
C           IS IT A CHARACTER VALUE OUTPUT SPEC?
C
            ELSE IF (CH .EQ. 'C') THEN
               IF (IPTR .EQ. 1) THEN
                  ACH = I1
               ELSE IF (IPTR .EQ. 2) THEN
                  ACH = I2
               ELSE IF (IPTR .EQ. 3) THEN
                  ACH = I3
               ELSE
                  ACH = I4
               ENDIF
               IF (PTYP .EQ. 1) THEN
                  CALL PUTC(ACH,FD)
               ELSE
```

```
            STRNG(OPTR) = ACH
            OPTR = OPTR + 1
         ENDIF
         IPTR = IPTR + 1
C
C        IS IT A STRING VALUE OUTPUT SPEC?
C
      ELSE IF (CH .EQ. 'S') THEN
         IF (IPTR .EQ. 1) THEN
            IF (PTYP .EQ. 1) THEN
               CALL PUTSTR(FD,I1)
            ELSE
               CALL STRCPY(I1,STRNG(OPTR))
               OPTR = OPTR + SLEN(I1)
            ENDIF
         ELSE IF (IPTR .EQ. 2) THEN
            IF (PTYP .EQ. 1) THEN
               CALL PUTSTR(FD,I2)
            ELSE
               CALL STRCPY(I2,STRNG(OPTR))
               OPTR = OPTR + SLEN(I2)
            ENDIF
         ELSE IF (IPTR .EQ. 3) THEN
            IF (PTYP .EQ. 1) THEN
               CALL PUTSTR(FD,I3)
            ELSE
               CALL STRCPY(I3,STRNG(OPTR))
               OPTR = OPTR + SLEN(I3)
            ENDIF
         ELSE
            IF (PTYP .EQ. 1) THEN
               CALL PUTSTR(FD,I4)
            ELSE
               CALL STRCPY(I4,STRNG(OPTR))
               OPTR = OPTR + SLEN(I4)
            ENDIF
         ENDIF
         IPTR = IPTR + 1
C
C        IS IT A FIELD WIDTH SPECIFIER?
C
      ELSE IF (CH .GE. '0' .AND. CH .LE. '9') THEN
         SPTR = 0
30       SPTR = SPTR + 1
         STR(SPTR) = ASC(CH)
         FPTR = FPTR + 1
         CH = FMT(FPTR:FPTR)
         IF (CH .GE. '0' .AND. CH .LE. '9') GO TO 30
         STR(SPTR+1) = 0
         INTWDTH = CTOI(STR)
         GO TO 20
C
C        UNKNOWN CONVERSION SO OUTPUT THE % AND CONVERSION CHAR
C
      ELSE
         IF (PTYP .EQ. 1) THEN
            CALL PUTC(ASC('%'),FD)
            CALL PUTC(ASC(CH),FD)
         ELSE
            STRNG(OPTR) = ASC('%')
```

```
                STRNG(OPTR+1) = ASC(CH)
                OPTR = OPTR + 2
             ENDIF
          ENDIF
          ENDIF
          FPTR = FPTR + 1
          GO TO 10
       ENDIF
       IF (PTYP .EQ. 2) STRNG(OPTR) = 0
       RETURN
       END
       SUBROUTINE STTY(MODE,VALUE)

CCC    STTY - SET A TERMINAL MODE.
C
       IMPLICIT INTEGER (A-Z)
       CHARACTER*(*) MODE
       INTEGER VALUE
C-----FILE: C167,KERMSIT
C
C      KERMIT SITE DEFINITIONS:
C
       PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
C-----FILE: C167,FILECOM
C
C      FILE I/O COMMON BLOCK DEFINITIONS
C
C      THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C      AND BEFORE ANY DATA DECLARATIONS.
C
       PARAMETER (MAXFILE=5, MAXWD=512)
       PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
       PARAMETER (ERROR=-2)
       PARAMETER (TAB=9, LF=10, CR=13, NEL=O"205")
       PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
       PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
       COMMON /FILEIOC/ FNAME(MAXFILE)
       CHARACTER*10 FNAME
       COMMON /FILEIO/ FMODE(MAXFILE),
     +    FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +    FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +    BINMODE, PARITY, DUPLEX
       INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
       LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
       DATA FMODE / MAXFILE*CLOSED /
       DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
       DATA RAWMODE, BINMODE / 2*.FALSE. /
       DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM
C
C      IS IT SETTING DUPLEX?
C
       IF (MODE .EQ. 'DUPLEX') THEN
          IF (VALUE .EQ. FULLDUP) THEN
C$           IF (UT2D .EQ. 1)
             CALL BELLC(L"FULL",0,0)
C$           ENDIF
             DUPLEX = FULLDUP
          ELSE IF (VALUE .EQ. HALFDUP) THEN
```

```
C$          IF (UT2D .EQ. 1)
            CALL BELLC(L"HALF",0,0)
C$          ENDIF
            DUPLEX = HALFDUP
         ELSE
            CALL DISPLA(' STTY - INVALID DUPLEX ',VALUE)
            CALL ABORT
         ENDIF
C
C     IS IT SETTING PARITY?
C
      ELSE IF (MODE .EQ. 'PARITY') THEN
         IF (VALUE .EQ. NOPAR .OR. VALUE .EQ. EVEPAR .OR.
     +       VALUE .EQ. ODDPAR .OR. VALUE .EQ. MRKPAR .OR.
     +       VALUE .EQ. SPCPAR) THEN
            PARITY = VALUE
         ELSE
            CALL DISPLA(' STTY - INVALID PARITY ',VALUE)
            CALL ABORT
         ENDIF
C
C     IS IT SETTING BINARY (NO TRANSLATION) I/O?
C
      ELSE IF (MODE .EQ. 'BINARY') THEN
         DO 100 I = 1,MAXFILE
            IF (FMODE(I) .NE. CLOSED) THEN
               IF (CTDEV(I)) THEN
                  IF (VALUE .EQ. 0) THEN
C$                   IF (UT2D .EQ. 1)
                     CALL IOP('EVE',FNAME(I))
C$                   ENDIF
                     BINMODE = .FALSE.
                  ELSE
C$                   IF (UT2D .EQ. 1)
                     CALL IOP('ODD',FNAME(I))
C$                   ENDIF
                     BINMODE = .TRUE.
                  ENDIF
               ENDIF
            ENDIF
100      CONTINUE
C
C     IS IT SETTING TRANSPARENT (RAW) I/O?
C
      ELSE IF (MODE .EQ. 'RAW') THEN
         IF (VALUE .EQ. 0) THEN
C$          IF (UT2D .EQ. 1)
            CALL BELLC(L"TTM",0,0)
C$          ENDIF
            RAWMODE = .FALSE.
         ELSE
C$          IF (UT2D .EQ. 1)
            CALL BELLC(L"ITM",O"2015",IRET)
C$          ENDIF
            RAWMODE = .TRUE.
            IF (IRET .NE. 0) THEN
               CALL REMARK(' STTY - CANNOT INITIATE TRANSPARENT MODE.')
               RAWMODE = .FALSE.
            ENDIF
         ENDIF
```

```
      ELSE
         CALL DISPLA(' STTY - INVALID MODE ',BOOL(MODE))
         CALL ABORT
      ENDIF
      RETURN
      END
      INTEGER FUNCTION GTTY(MODE)

CCC   GTTY - GET A TTY MODE.
C
      IMPLICIT INTEGER (A-Z)
      CHARACTER*(*) MODE
C-----FILE: C167,FILECOM
C
C     FILE I/O COMMON BLOCK DEFINITIONS
C
C     THIS FILE SHOULD BE INCLUDED AFTER ANY PROCEDURE DECLARATIONS
C     AND BEFORE ANY DATA DECLARATIONS.
C
      PARAMETER (MAXFILE=5, MAXWD=512)
      PARAMETER (EOF=-1, CLOSED=0, RD=1, WR=2, CREATE=3)
      PARAMETER (ERROR=-2)
      PARAMETER (TAB=9, LF=10, CR=13, NEL=O"205")
      PARAMETER (NOPAR=0, EVEPAR=1, ODDPAR=2, MRKPAR=3, SPCPAR=4)
      PARAMETER (UNKNOWN=0, FULLDUP=1, HALFDUP=2)
      COMMON /FILEIOC/ FNAME(MAXFILE)
      CHARACTER*10 FNAME
      COMMON /FILEIO/ FMODE(MAXFILE),
     +   FCHBUF(MAXWD,MAXFILE), FWPTR(MAXFILE), FNWDS(MAXFILE),
     +   FEOF(MAXFILE), FWSHFT(MAXFILE), CTDEV(MAXFILE), RAWMODE,
     +   BINMODE, PARITY, DUPLEX
      INTEGER FMODE, FCHBUF, FWPTR, FNWDS, FWSHFT, PARITY, DUPLEX
      LOGICAL FEOF, CTDEV, RAWMODE, BINMODE
      DATA FMODE / MAXFILE*CLOSED /
      DATA FWPTR,FNWDS / MAXFILE*0, MAXFILE*0 /
      DATA RAWMODE, BINMODE / 2*.FALSE. /
      DATA PARITY, DUPLEX / NOPAR, UNKNOWN /
C-----END OF FILE: FILECOM
C
C     IS IT DUPLEX THEY'RE LOOKING FOR?
C
      IF (MODE .EQ. 'DUPLEX') THEN
         GTTY = DUPLEX
C
C     IS IT PARITY THEY'RE LOOKING FOR?
C
      ELSE IF (MODE .EQ. 'PARITY') THEN
         GTTY = PARITY
      ELSE
         CALL DISPLA(' GTTY - INVALID MODE ',BOOL(MODE))
         CALL ABORT
      ENDIF
      RETURN
      END
      SUBROUTINE AS2DPC(ASTR,DSTR)

CCC   AS2DPC - TRANSLATE AN ASCII STRING BUFFER TO DPC CHAR STRING.
C
C     ASCII STRING IS TERMINATED BY A ZERO BYTE.
C
```

```
C     ***** THE NOS CONVERSION TABLE HAS NOT BEEN SET UP *****
C
C-----FILE: C167,KERMSIT
C
C     KERMIT SITE DEFINITIONS:
C
      PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
      INTEGER ASTR(100), CLEN
      CHARACTER DSTR*(*)
C
      CHARACTER*1 DPCTBL(128)
C$    IF (UT2D .EQ. 1)
      DATA DPCTBL/' ',31*' ',' ','!','"','#','$','%','&','''',
     +            '(',')','*','+',',','-','.','/','0',
     +            '1','2','3','4','5','6','7','8','9',
     +            ':',';','<','=','>','?','@',
     +            'A','B','C','D','E','F','G','H','I',
     +            'J','K','L','M','N','O','P','Q','R',
     +            'S','T','U','V','W','X','Y','Z',
     +            '[','\',']','↑',' ',' ',
     +            'A','B','C','D','E','F','G','H','I',
     +            'J','K','L','M','N','O','P','Q','R',
     +            'S','T','U','V','W','X','Y','Z',
     +            5*' '/
C$    ENDIF
C
      I = 1
      CLEN = LEN(DSTR)
      DSTR = ' '
10    IF (ASTR(I) .NE. 0 .AND. I .LE. CLEN) THEN
         IF (ASTR(I) .GT. 127) THEN
            DSTR(I:I) = DPCTBL(1)
         ELSE
            DSTR(I:I) = DPCTBL(ASTR(I) + 1)
         ENDIF
         I = I + 1
         GO TO 10
      ENDIF
C
      RETURN
      END
      INTEGER FUNCTION ASC(DPCH)

CCC   ASC - CONVERT A DPC CHARACTER TO ASCII.
C
C     ***** THE NOS CONVERSION TABLE IS NOT CORRECT *****
C
C-----FILE: C167,KERMSIT
C
C     KERMIT SITE DEFINITIONS:
C
      PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
      CHARACTER*1 DPCH
C
      INTEGER ASCTBL(64)
C$    IF (UT2D .EQ. 1)
      DATA ASCTBL/0,97,98,99,100,101,102,103,104,105,106,107,108,109,
C                   A  B  C   D   E   F   G   H   I   J   K   L   M
```

```
     +            110,111,112,113,114,115,116,117,118,119,120,121,122,
C                   N   O   P   Q   R   S   T   U   V   W   X   Y   Z
     +            48,49,50,51,52,53,54,55,56,57,
C                  0  1  2  3  4  5  6  7  8  9
     +            43,45,42,47,40,41,36,61,32,44,46,34,91,93,58,
C                  +  -  *  /  (  )  $  =     ,  .  "  [  ]  :
     +            39,35,33,38,94,37,60,62,64,63,92,59/
C                  '  #  !  &  ↑  %  <  >  @  ?  \  ;
C$    ELSE
      DATA ASCTBL/58,97,98,99,100,101,102,103,104,105,106,107,108,109,
C                 :  A  B  C   D   E   F   G   H   I   J   K   L   M
     +            110,111,112,113,114,115,116,117,118,119,120,121,122,
C                   N   O   P   Q   R   S   T   U   V   W   X   Y   Z
     +            48,49,50,51,52,53,54,55,56,57,
C                  0  1  2  3  4  5  6  7  8  9
     +            43,45,42,47,40,41,36,61,32,44,46,34,91,93,37,
C                  +  -  *  /  (  )  $  =     ,  .  "  [  ]  %
     +            39,35,33,38,94,37,60,62,64,63,92,59/
CWRONG>>>         '  #  !  &  ↑  %  <  >  @  ?  \  ;
C$    ENDIF
C
      CALL COLSEQ('DISPLAY')
      ASC = ASCTBL(ICHAR(DPCH) + 1)
C
      RETURN
      END
      SUBROUTINE DPC2AS(DSTR,ASTR,N)
C
C  TRANSLATE STRING OF DISPLAY CODE CHARACTERS TO UPPERCASE ASCII.
C  STRING IS N CHARACTERS (WORDS) LONG.
C
C     ***** THE NOS TRANSLATION TABLE HAS NOT BEEN DONE YET *****
C
C-----FILE: C167,KERMSIT
C
C     KERMIT SITE DEFINITIONS:
C
      PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
      CHARACTER*(*) DSTR
      INTEGER ASTR(200),N
C
      INTEGER ASCTBL(64)
C$    IF (UT2D .EQ. 1)
      DATA ASCTBL/0,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,
C                  A  B  C  D  E  F  G  H  I  J  K  L  M  N  O  P  Q
     +             82,83,84,85,86,87,88,89,90,48,49,50,51,52,53,54,55,
C                  R  S  T  U  V  W  X  Y  Z  0  1  2  3  4  5  6  7
     +             56,57,43,45,42,47,40,41,36,61,32,44,46,34,91,93,58,
C                  8  9  +  -  *  /  (  )  $  =     ,  .  "  [  ]  :
     +             39,35,33,38,94,37,60,62,64,63,92,59/
C                  '  #  !  &  ↑  %  <  >  @  ?  \  ;
C$    ENDIF
C
      CALL COLSEQ('DISPLAY')
      DO 1 I=1,N
         ASTR(I) = ASCTBL((ICHAR(DSTR(I:I))) + 1)
    1    CONTINUE
C
C     SET ASCII END-OF-STRING-BUFFER
```

```
C
      ASTR(N+1) = 0
C
      RETURN
      END
      INTEGER FUNCTION CTOI(ASTR)

CCC   CTOI - CONVERT CHARACTER BUFFER TO INTEGER.
C
C     CTOI CONVERTS THE NUMBER USING BASE 10 AS A DEFAULT.
C     A SUFFIX OF H WILL CONVERT USING BASE 16 AND A SUFFIX
C     OF O WILL CONVERT USING BASE 8.  DEFAULT SUFFIX IS
C     D.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
      PARAMETER (DIG0=48, DIG7=55, DIG9=57, BIGA=65, BIGB=66, BIGD=68)
      PARAMETER (BIGF=70, BIGH=72, BIGO=79, LETA=97, LETB=98, LETD=100)
      PARAMETER (LETF=102, LETH=104, LETO=111)
      INTEGER ASTR(200)

      BASE = 0
```

```
      PTR = 0
C
C     FIND LAST VALID DIGIT
C
10    PTR = PTR + 1
      IF (ASTR(PTR) .NE. 0) GO TO 10
      PTR = PTR - 1
      IF (ASTR(PTR) .EQ. LETO .OR. ASTR(PTR) .EQ. BIGO .OR.
     +    ASTR(PTR) .EQ. LETB .OR. ASTR(PTR) .EQ. BIGB .OR.
     +    ASTR(PTR) .EQ. LETH .OR. ASTR(PTR) .EQ. BIGH) THEN
         EOD = PTR - 1
      ELSE
         EOD = PTR
         PTR = PTR + 1
      ENDIF
C
C     TRY TO FIGURE OUT THE BASE
C
      IF (ASTR(PTR) .EQ. 0) THEN
         BASE = 10
      ELSE IF (ASTR(PTR) .EQ. LETO .OR. ASTR(PTR) .EQ. BIGO .OR.
     +         ASTR(PTR) .EQ. LETB .OR. ASTR(PTR) .EQ. BIGB) THEN
         BASE = 8
      ELSE IF (ASTR(PTR) .EQ. LETH .OR. ASTR(PTR) .EQ. BIGH) THEN
         BASE = 16
      ENDIF
C
C     IF DIDN'T FIND A BASE
C
      IF (BASE .EQ. 0) THEN
         CALL FPRINTF(STDOUT,'CTOI - INVALID BASE %C\N',ASTR(PTR),0,0,0)
         CTOI = 0
         RETURN
      ENDIF
C
C     ADD UP THE DIGITS
C
      TOTAL = 0
      ISNEG = 1
      DO 100 I = 1,EOD
         CH = ASTR(I)
         IF (CH .EQ. MINUS) THEN
            ISNEG = -1
            GO TO 100
         ENDIF
         IF (BASE .EQ. 10) THEN
            IF (CH .LT. DIG0 .OR. CH .GT. DIG9) THEN
               CALL FPRINTF(STDOUT,'CTOI - INVALID DECIMAL DIGIT %C\N',
     +                      CH,0,0,0)
               CTOI = 0
               RETURN
            ELSE
               CH = CH - DIG0
            ENDIF
         ELSE IF (BASE .EQ. 8) THEN
            IF (CH .LT. DIG0 .OR. CH .GT. DIG7) THEN
               CALL FPRINTF(STDOUT,'CTOI - INVALID OCTAL DIGIT %C\N',
     +                      CH,0,0,0)
               CTOI = 0
               RETURN
```

```
            ELSE
               CH = CH - DIG0
            ENDIF
         ELSE IF (BASE .EQ. 16) THEN
            IF (CH .GE. DIG0 .AND. CH .LE. DIG9) THEN
               CH = CH - DIG0
            ELSE IF (CH .GE. LETA .AND. CH .LE. LETF) THEN
               CH = 10 + CH - LETA
            ELSE IF (CH .GE. BIGA .AND. CH .LE. BIGF) THEN
               CH = 10 + CH - BIGA
            ELSE
               CALL FPRINTF(STDOUT,'CTOI - INVALID HEX DIGIT %C\N',
     +                      CH,0,0,0)
               CTOI = 0
               RETURN
            ENDIF
         ENDIF
         TOTAL = TOTAL*BASE + CH
100   CONTINUE
      CTOI = TOTAL * ISNEG
      RETURN
      END
      INTEGER FUNCTION ITOS(INT,STR,MINWID)

CCC   ITOS - CONVERT AN INTEGER TO STRING FORMAT.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
```

```
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
      INTEGER STR(200)

      WIDTH = 0
      IF (INT .LT. 0) THEN
         WIDTH = 1
         STR(WIDTH) = ASC('-')
      ENDIF
      VAL = IABS(INT)
      ASCII0 = ASC('0')
10    WIDTH = WIDTH + 1
      STR(WIDTH) = MOD(VAL,10) + ASCII0
      VAL = VAL / 10
      IF (VAL .NE. 0) GO TO 10
      STR(WIDTH+1) = 0
C
C     NOW REVERSE THE DIGITS
C
      IPTR = 1
      ENDPTR = WIDTH
      IF (STR(IPTR) .EQ. ASC('-')) IPTR = IPTR + 1
20    IF (IPTR .LT. ENDPTR) THEN
         TCH = STR(IPTR)
         STR(IPTR) = STR(ENDPTR)
         STR(ENDPTR) = TCH
         IPTR = IPTR + 1
         ENDPTR = ENDPTR - 1
         GO TO 20
      ENDIF
      ITOS = WIDTH
      RETURN
      END
      SUBROUTINE GETEMSG(STRNG)

CCC   GETEMSG - GET AN ERROR MESSAGE STRING FOR THE CURRENT ERROR.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
```

```
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,PROTCOM
C
C     KERMIT PROTOCOL COMMON BLOCK
C
      COMMON /PROTO/ PACKET(MAXPACK), RECPACK(MAXPACK),FILESTR(MAXPACK),
     +   PSIZE, PACKNUM, NUMTRY, MAXRTRY, MAXRINI, STATE, IFD, OFD, FFD,
     +   DELAYFP, SAVEDPX,
C     STORAGE FOR STATISTICS
     +   ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
      INTEGER PACKET, RECPACK, FILESTR, PSIZE, PACKNUM, NUMTRY, STATE
      INTEGER MAXRTRY, MAXRINI, IFD, OFD, FFD, DELAYFP, SAVEDPX
      INTEGER ABORTYP, STARTIM, ENDTIM, SCHCNT, RCHCNT, SCHOVRH, RCHOVRH
C-----END OF FILE: PROTCOM
      INTEGER DIREC(8,2)
      INTEGER PACKNAM(9,5)
      DATA DIREC / 115, 101, 110, 100, 4*0,
C                    S    E    N    D
     +             114, 101,  99, 101, 105, 118, 101, 0 /
C                    R    E    C    E    I    V    E
      DATA PACKNAM / 73, 110, 105, 116, 5*0,
C                     I    N    I    T
     +               70, 105, 108, 101, 110,  97, 109, 101, 0,
C                     F    I    L    E    N    A    M    E
     +               68,  97, 116,  97, 5*0,
C                     D    A    T    A
     +               69,  79,  70, 6*0,
C                     E    O    F
     +               66, 114, 101,  97, 107, 4*0 /
C                     B    R    E    A    K
         PTYP = ABORTYP.AND.O"77"
         IF (PTYP .EQ. 4) PTYP = 3
         IF (PTYP .EQ. O"10") PTYP = 4
         IF (PTYP .EQ. O"20") PTYP = 5
         DTYP = SHIFT(ABORTYP.AND.O"300",-6)
         IF ((ABORTYP.AND.TOOMANY) .NE. 0) THEN
            CALL SPRINTF(STRNG,'↑CANNOT %S %S PACKET',DIREC(1,
     +         DTYP),PACKNAM(1,PTYP),0,0)
         ELSE IF ((ABORTYP.AND.INVALID) .NE. 0) THEN
            CALL SPRINTF(STRNG,
```

```
     +     '↑RECEIVED AN INVALID PACKET WHILE TRYING TO %S %S PACKET',
     +     DIREC(1,DTYP),PACKNAM(1,PTYP),0,0)
         ELSE IF ((ABORTYP.AND.SEQERR) .NE. 0) THEN
            CALL SPRINTF(STRNG,
     +      '↑PACKET SEQUENCE ERROR WHILE TRYING TO %S %S PACKET',
     +      DIREC(1,DTYP),PACKNAM(1,PTYP),0,0)
         ELSE IF ((ABORTYP.AND.LCLFILE) .NE. 0) THEN
            CALL SPRINTF(STRNG,
     +         '↑FILE IS ALREADY LOCAL',0,0,0,0)
         ENDIF
      RETURN
      END
      INTEGER FUNCTION GETFILE(FN)

CCC   GETFIL - OPEN A FILE FOR WRITING PACKET DATA TO.
C
C     GETFILE WILL TRY TO CREATE A FILE TO WRITE TO.  IF IT
C     ALREADY EXISTS, THEN IT WILL FAIL.
C
      IMPLICIT INTEGER (A-Z)
      CHARACTER*10 FILENAM
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
```

```
C-----FILE: C167,DBUGCOM
C
C     DEBUG COMMON BLOCK
C
      PARAMETER (DBGOFF=0, DBGSTAT=1, DBGPACK=2, DBGALL=3)
      COMMON /DEBUG/ DEBUG, DEBUGFD, DEBUGFN(8)
C-----END OF FILE: DBUGCOM
C
C     GET THE DPC VERSION OF THE FILENAME
C
      CALL AS2DPC(FN,FILENAM)
      CALL FILCHK(FILENAM)
      GETFILE = FOPEN(FILENAM,3)
      RETURN
      END
      SUBROUTINE GETNOW(MM,DD,YY,HR,MIN,SEC)

CCC   GET THE CURRENT DATE AND TIME.
C
      IMPLICIT INTEGER (A-Z)
C-----FILE: C167,KERMSIT
C
C     KERMIT SITE DEFINITIONS:
C
      PARAMETER (UT2D=1, NOS=0, NOSBE=0, SCOPE=0)
C-----END OF FILE: KERMSIT
      CHARACTER*10 DATE, TIME, STRING

      STRING = DATE()
      OFFSET = ICHAR('0')
C$    IF (UT2D .EQ. 1)
      DD = (ICHAR(STRING(2:2))-OFFSET)*10 + ICHAR(STRING(3:3))-OFFSET
      MM = -1
      IF (STRING(5:7) .EQ. 'JAN') THEN
         MM = 1
      ELSE IF (STRING(5:7) .EQ. 'FEB') THEN
         MM = 2
      ELSE IF (STRING(5:7) .EQ. 'MAR') THEN
         MM = 3
      ELSE IF (STRING(5:7) .EQ. 'APR') THEN
         MM = 4
      ELSE IF (STRING(5:7) .EQ. 'MAY') THEN
         MM = 5
      ELSE IF (STRING(5:7) .EQ. 'JUN') THEN
         MM = 6
      ELSE IF (STRING(5:7) .EQ. 'JUL') THEN
         MM = 7
      ELSE IF (STRING(5:7) .EQ. 'AUG') THEN
         MM = 8
      ELSE IF (STRING(5:7) .EQ. 'SEP') THEN
         MM = 9
      ELSE IF (STRING(5:7) .EQ. 'OCT') THEN
         MM = 10
      ELSE IF (STRING(5:7) .EQ. 'NOV') THEN
         MM = 11
      ELSE IF (STRING(5:7) .EQ. 'DEC') THEN
         MM = 12
      ENDIF
      YY = (ICHAR(STRING(9:9))-OFFSET)*10 + ICHAR(STRING(10:10))-OFFSET
C$    ENDIF
```

```
C$    IF (NOS .EQ. 1)
      YY = (ICHAR(STRING(2:2))-OFFSET)*10 + ICHAR(STRING(3:3))-OFFSET
      DD = (ICHAR(STRING(5:5))-OFFSET)*10 + ICHAR(STRING(6:6))-OFFSET
      MM = (ICHAR(STRING(8:8))-OFFSET)*10 + ICHAR(STRING(9:9))-OFFSET
C$    ENDIF
C$    IF (NOSBE .EQ. 1)
       DD = (ICHAR(STRING(5:5))-OFFSET)*10 + ICHAR(STRING(6:6))-OFFSET
       MM = (ICHAR(STRING(2:2))-OFFSET)*10 + ICHAR(STRING(3:3))-OFFSET
       YY = (ICHAR(STRING(8:8))-OFFSET)*10 + ICHAR(STRING(9:9))-OFFSET
C$    ENDIF
      YY = YY + 1900
      STRING = TIME()
      HR  = (ICHAR(STRING(2:2))-OFFSET)*10 + ICHAR(STRING(3:3))-OFFSET
      MIN = (ICHAR(STRING(5:5))-OFFSET)*10 + ICHAR(STRING(6:6))-OFFSET
      SEC = (ICHAR(STRING(8:8))-OFFSET)*10 + ICHAR(STRING(9:9))-OFFSET
      RETURN
      END
      SUBROUTINE FILCHK(FN)
      CHARACTER *(*) FN
C
C CHECK VALIDITY OF FILENAME, REMOVE SPECIAL CHARACTERS,
C  TRUNCATE TO 7 CHARACTERS, BLANK FILL TO 10
C
      INTEGER PTR,CH,LENGTH
C
      PTR = 0
      LENGTH = LEN(FN)
      DO 2 I=1,7
    1    PTR = PTR + 1
         IF (PTR .GT. LENGTH) GO TO 3
         CH = ICHAR(FN(PTR:PTR))
         IF (CH .LT. 1 .OR. CH .GT. 36) GO TO 1
         FN(I:I)=FN(PTR:PTR)
    2 CONTINUE
      I = 8
C
    3 IF (LENGTH .GT. 7) THEN
         DO 4 J=I,LENGTH
            FN(J:J) = ' '
    4    CONTINUE
      ENDIF
C
      RETURN
      END
      SUBROUTINE RDPARAM(PDATA)

CCC   RDPARAM - GET THE PACKET PARAMETERS FROM THE OTHER KERMIT.
C
      IMPLICIT INTEGER (A-Z)
      INTEGER PDATA(100)
      INTEGER PARAMS(11)
      EQUIVALENCE (PARAMS,SPKSIZ)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
```

```
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=O"205", NULL=O"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=O"1000", INVALID=O"2000", SEQERR=O"4000")
      PARAMETER (LCLFILE=O"10000")
      PARAMETER (SENDING=O"100", READING=O"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=O"10")
      PARAMETER (BRKERR=O"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
C
C     DEFINE CTL AND UNCHAR STATEMENT FUNCTIONS
C
      CTL(ASCCH) = ASCCH .XOR. O"100"
      UNCHAR(ASCCH) = ASCCH - BLANK
C
C     CYCLE THROUGH THE LIST OF PARAMETERS UNTIL THE END-OF-LIST
C     IS FOUND (A 0 BYTE).
C
      I = 1
10    IF (PDATA(I) .NE. 0) THEN
C
```

```
C        IS IT THE PAD CHARACTER?
C
         IF (I .EQ. 4) THEN
            PARAMS(I) = CTL(PDATA(I))
            IF (PARAMS(I) .EQ. 0) PARAMS(I) = NULL
C
C        IS IT THE QUOTE CHARACTER?
C
         ELSE IF (I .EQ. 6) THEN
            PARAMS(I) = PDATA(I)
         ELSE
            IF (UNCHAR(PDATA(I)) .NE. 0) THEN
               PARAMS(I) = UNCHAR(PDATA(I))
            ENDIF
         ENDIF
         I = I + 1
         GO TO 10
      ENDIF
      RETURN
      END
      SUBROUTINE REMOVE(FN)

CCC   REMOVE - REMOVE A FILE FROM THE LOCAL FILE LIST.
C
      IMPLICIT INTEGER (A-Z)
      CHARACTER*10 FNAME

      CALL AS2DPC(FN,FNAME)
      OPEN(UNIT=1,FILE=FNAME)
      CLOSE(UNIT=1,STATUS='DELETE')
      RETURN
      END
      SUBROUTINE STRCPY(S1,S2)

CCC   STRCPY - COPY ONE ASCII STRING TO ANOTHER
C
      IMPLICIT INTEGER (A-Z)
      INTEGER S1(200),S2(200)

      I1 = 1
10    S2(I1) = S1(I1)
      IF (S1(I1) .NE. 0) THEN
         I1 = I1 + 1
         GO TO 10
      ENDIF
      RETURN
      END
      INTEGER FUNCTION SLEN(STR)

CCC   SLEN - RETURN THE LENGTH OF A ZERO TERMINATED ASCII STRING BUFFER.
C
      INTEGER STR(200)

      I = 0
10    IF (STR(I+1) .NE. 0) THEN
         I = I + 1
         GO TO 10
      ENDIF
      SLEN = I
      RETURN
```

```
      END
      INTEGER FUNCTION SNDPAR(PDATA)

CCC   SNDPAR - SET UP PARAMETERS TO SEND TO OTHER KERMIT.
C
      IMPLICIT INTEGER (A-Z)
      INTEGER PDATA(100)
C-----FILE: C167,KERMDEF
C
C     KERMIT SYMBOL DEFINITIONS
C
C     FILE I/O DEFINITIONS
C
      PARAMETER (STDIN=1, STDOUT=2)
C
C     CHARACTER DEFINITIONS
C
      PARAMETER (SOH=1, TAB=9, LF=10, CR=13, BLANK=32, MINUS=45)
      PARAMETER (QMARK=63, DEL=127, NEL=0"205", NULL=0"4000", EOF=-1)
C
C     MISCELLANEOUS
C
      PARAMETER (OK=1, ERROR=-2, ON=1, OFF=0)
C
C     PROTOCOL DEFINITIONS
C
      PARAMETER (UNKNOWN=0, FULL=1, HALF=2)
      PARAMETER (NONE=0, EVEN=1, ODD=2, MARK=3, SPACE=4)
      PARAMETER (MAXINIT=20, MAXTRY=5)
      PARAMETER (MAXPACK=94, MYTIME=10, MYPAD=0, MYPADCH=0)
      PARAMETER (MYEOL=13, MYQUOTE=35, QUOT8CH=78, MYCKTYP=49)
      PARAMETER (PREFXCH=126)
C
C     PACKET TYPES
C
      PARAMETER (A=65, B=66, C=67, D=68, E=69, F=70, G=71, L=76, N=78)
      PARAMETER (R=82, S=83, Y=89, Z=90)
C
C     PACKET ERROR DEFINITIONS
C
      PARAMETER (TOOMANY=0"1000", INVALID=0"2000", SEQERR=0"4000")
      PARAMETER (LCLFILE=0"10000")
      PARAMETER (SENDING=0"100", READING=0"200")
      PARAMETER (INITERR=1, FILERR=2, DATAERR=4, EOFERR=0"10")
      PARAMETER (BRKERR=0"20")
C-----END OF FILE: KERMDEF
C-----FILE: C167,PACKCOM
C
C     KERMIT PACKET DESCRIPTION COMMON BLOCK
C
C     DO NOT ALLOCATE ANY STORAGE BETWEEN PACKSIZ AND SNDSYNC!
C
C     ALLOCATE STORAGE FOR WHAT I WANT
      COMMON /PACKET/ PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH,
     +                QUOTE8, CHKTYP, RPREFIX, RESERVE(2), SYNC,
C     ALLOCATE STORAGE FOR WHAT PARTNER WANTS
     +                SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL, SPQUOTE,
     +                S8QUOTE, SCHKTYP, SREPEAT, UNUSED(2), SNDSYNC
      INTEGER SYNC, SNDSYNC
      INTEGER PACKSIZ, TIMEOUT, NPAD, PADCH, EOLCH, QUOTECH, QUOTE8,
```

```
     +        CHKTYP, RPREFIX, SPKSIZ, STIMOUT, SPAD, SPADCH, SPEOL,
     +        SPQUOTE, S8QUOTE, SCHKTYP, SREPEAT
C-----END OF FILE: PACKCOM
C
C     DEFINE CTL AND TOCHAR STATEMENT FUNCTIONS
C
      CTL(ASCCH) = ASCCH .XOR. O"100"
      TOCHAR(ASCCH) = ASCCH + BLANK
C
C     SEND WHAT WE WANT
C
      PDATA(1) = TOCHAR(PACKSIZ)
      PDATA(2) = TOCHAR(TIMEOUT)
      PDATA(3) = TOCHAR(NPAD)
      PDATA(4) = CTL(PADCH)
      PDATA(5) = TOCHAR(EOLCH)
      PDATA(6) = QUOTECH
      PDATA(7) = 0
C
C     RETURN LENGTH OF HOW MANY THINGS WE WANT TO SET
C
      SNDPAR = 6
C
C     OTHER VALUES ARE SET BY DEFAULT TO NOT OPERATE
C
      RETURN
      END
      SUBROUTINE SLEEP(SECONDS)
CC
C     SLEEP - HOLD FOR <SECONDS> SECONDS.
C
      IMPLICIT INTEGER (A-Z)

      DO 100 I=1,SECONDS
         CALL DELAY(1000)
100   CONTINUE
      RETURN
      END
      SUBROUTINE DELAY(MSEC)
CC
C     DELAY - HOLD THINGS UP FOR <MSEC> MILISECS.
C
C     **** THIS IS PROBABLY SYSTEM DEPENDENT CODE *****
C          IF YOU MODIFY IT USE CONDITIONAL COMPILATION
C
      INTEGER RTCL,RTCL1

      DATA RTCL/0/
C
C     SNOOP THE SYSTEM REAL TIME CLOCK AT LOW CORE ADDRESS 36B.
C
      CALL SNP(O"36",RTCL,1)
10    CALL SNP(O"36",RTCL1,1)
C
C     CONVERT FROM JIFS TO MILLISECS
C
      IF((RTCL1-RTCL)/4.096 .GT. MSEC) GO TO 20
C
C     SLEEP FOR 100 MILLISECS
C
```

```
      CALL PP(L"RCL",0)
      GO TO 10
20    RTCL = RTCL1
      RETURN
      END
      SUBROUTINE SNP(ADDR,MEMA,N)
CC
C     SNOOP - SNOOP LOW CORE.
C
C     **** THIS IS PROBABLY SYSTEM DEPENDENT CODE *****
C          IF YOU MODIFY IT USE CONDITIONAL COMPILATION
C
      INTEGER ADDR,MEMA(1)
      INTEGER SNPA(2)
C
C     BUILD THE SNOOP REQUEST FOR SNOOPING LOWCORE
C
      SNPA(1) = 0
      SNPA(2) = SHIFT(N,48) .OR. SHIFT(ADDR,24) .OR. LOCF(MEMA)
      CALL PP(L"SNPP",1,LOCF(SNPA))
      RETURN
      END
```

SY.FDC

0069

BIN 7W

CU20B, TOPS-20 Monitor 5.1(5101)-2

Request completed 19-Oct-83 4:59:09AM
112 Pages of Output

AUG 2 2 1983

Jim Knutson @ utexas-11

writing Kermit for Cyber 170/750
in Cyber C, rewriting it later in FTN5.

run home grown OS

which they wrote themselves

512 - 471 - 3241 ext 359

Computation Center Rm 1
U. of Texas
Austin, Texas, 78712

SEP 2 1 1983

Steve Jay
CE, U of AZ, Tucson
602-621-5074

Cyber 175 NOS/BE (old system, holdout)

plain NOS is newer

Wants K for Cyber; referred him to Knutson

OCT 1 3 1983

Knutson says it's done -
Fortran (needs preprocessor)