

```

41984,32768,3968,
/KERMIT FOR NIC-80 SERIES
/FROM KERM**.A
/
/CONSTRUCTION -
/  *RUN CLEAR
/  *RUN LOADER
/  @KERM**.C:M
/  *LOA FPP72 102000
/  *LOA NICBUG 114632
/  *LOA BOOTS
/  *STO KERMIT 102000-117777;110000
/
/LAYOUT -
/  0-2777          I/O DISK-BUFFER (0-1777 FOR FLOPPY)
/  3000-5777       DISK DIRECTORY (4000-5777 FOR FLOPPY)
/  6000-7777       FPP72, DIRFUN OVERLAYS, DEMON MONITOR HEAD
/  100000-100777   INPUT "INTERRUPT" ROTATING BUFFER
/  101000-101777   OUTPUT "INTERRUPT" ROTATING BUFFER
/  102000-102777   FPP72 ON ENTRY, THEN OUTPUT PACKET CONSTRUCTION
/  103000-103777   FPP72 ON ENTRY, THEN INPUT PACKET DECONSTRUCTION
/  104000-105777   TEXT FOR COMMAND SUMMARY, NOTES, ETC
/  106000-107777   WORK ROUTINES FOR RECEIVE MODE
/  110700-111777   COMMAND LIST, CONTROLLING ROUTINES, SOFTWARE "INTERRUPTS"
/  112000-113777   ROUTINES FOR TRANSMIT, PACKET CONSTRUCTION, ETC
/  114000-115777   FILENAME PARSING, NICBUG, BOOTS
/  116000-117777   DISK ROUTINES, GENERAL UTILITIES, DISPLAY
/
/DEFINITIONS
RSINF=6564
RSIN=44563
RSOUTF=6574
RSOUT=4573
TACMQZ=634354
INCOR=4341
PAGSKP
*110000
/PAGE 110000 HOLDS THE COMMAND BLOCK, SOFTWARE "INTERRUPTS" AND
/MAIN CONTROLLING ROUTINES
/USES W-PREFIX POINTERS
/COMMAND BLOCK
/ACCEPTS COMMANDS AFTER A ":" PROMPT
      JMS @ WINTRO      /ANNOUNCE KERMIT?
COMBAK,ONEM IGBFLG      /IGNORE REMOTE SYSTEM
      ZERM IGAFLG      /CLEAR FLAGS
      ZERM NULAFG
      ZERM NULBFG
      ZERM LCFLAG
      ZERM CMDFLG
      ZERM IGCFLG
      ZERM IGTOFG
      ZERM CXFLG
      ZERM CZFLG
      JMS RSEMP
      ONEM IGBFLG      /IGNORE B-INPUT
      JMS CRLF
      MEMA (":
      JMS TYPE
      JMS CREAD        /READ SANS TIME-OUT
LCBAK, A-MZ ("B        /CALL NICBUG

```

```

ZERZ
JMP @ BUGOUT
A-MZ ("C           /CONNECT TO REMOTE SYSTEM
ZERZ
JMP DUMSRT
A-MZ ("D           /DIRECTORY LISTING
ZERZ
JMS @ WDIRLS
A-MZ ("E           /SET ECHO
ZERZ
JMS MAKEKO
A-MZ ("H           /HELP! PRINT COMMAND SUMMARY
ZERZ
JMS @ WHELP
A-MZ ("I           /TOGGLE INDICATORS & ARCHIVING
ZERZ
JMS @ WARCTO
A-MZ ("K           /KILL ECHO
ZERZ
JMS KILEKO
A-MZ ("L           /LOGIN ON REMOTE SYSTEM
ZERZ
JMS LOGIN
A-MZ ("M           /QUIT TO DEMON MONITOR
ZERZ
JMP @ MONOUT
A-MZ ("N           /PRINT NOTES ON OPERATION
ZERZ
JMS @ WNOTPR
A-MZ ("P           /SET KERMIT PARAMETERS
ZERZ
JMS SETPAR
A-MZ ("Q           /QUIT TO DEMON
ZERZ
JMP @ MONOUT
A-MZ ("R           /RECEIVE DATA
ZERZ
JMS GETDAT
A-MZ ("S           /SEND DATA
ZERZ
JMS PUTDAT
A-MZ ("T           /TOGGLE RS232 B-INPUT ON/OFF
ZERZ
JMS BTOG
A-MZ ("U           /CONFIGURE USERNAME, PASSWORD
ZERZ
JMS @ WCONFI
MMOZ LCFLAG
JMP LCASE
MMOZ CMDFLG
JMP BADCHA
COMFIN, JMS CRLF
JMP COMBAK
BADCHA, MEMA ("?    /EXPOSTULATION
JMS TYPE
JMP COMFIN
LCASE, A-MA (40     /PERHAPS LOWER-CASE?
ONEM LCFLAG
JMP LCBK
CMDFLG, 0

```

```

LCFLAG,0
/
/DUMB TERMINAL MODE
/ONLY EXIT IS VIA A CTRL/G
DUMSRT,JMS CRLF
      JMS QIKRET
DUMTRM,ZERM IGAFLG      /CLEAR FLAGS
      ZERM IGBFLG
      ZERM KBFLG
      ZERM NULAFG
      ZERM NULBFG
      ZERM CMDFLG
      ZERM RTFLG
      JMS OPENB          /SEND AN X-ON
      JMS RSDUTY         /DUTY CYCLE
      JMP #-1            /AND KEEP CYCLING TIL INTERRUPTED
/RS232 DUTY CYCLE
/LOOKS AT ALL FOUR RS232 FLAGS, AND THE TWO BUFFERS
/AND TAKES APPROPRIATE ACTION
/A-INPUT ARRIVALS GO TO OP-BUFFER IF KBFLG=0, TO IP-BUFFER IF =1
/B-INPUT ARRIVALS GO TO IP-BUFFER
/FLAGS IGAFLG, IGBFLG CAUSE A,B INPUTS TO BE IGNORED
/THE OP-BUFFER IS TRANSMITTED OUT VIA RS232-B
/THE IP-BUFFER IS PRINTED ON THE VDU
/CHARACTERS IN BUFFERS ARE ALWAYS PROCESSED AS SOON AS POSSIBLE
/BY BEING EITHER PRINTED OR TRANSMITTED
/THE PRINTING OR TRANSMISSION MAY BE INHIBITED BY SETTING NULAFG
/OR NULBFG, BUT THE BUFFERS WILL STILL RECEIVE CHARACTERS
/CONTROL FLAGS
KBFLG, 0      /SEND KB CHAR TO OP IF =0, TO IP IF =1
IGAFLG,0      /IGNORE KB, EXCEPT CTRL/G, IF =1
IGBFLG,0      /IGNORE B-INPUT, IF =1
NULAFG,0      /INHIBIT IP-PRINTING IF =1
NULBFG,0      /SUSPEND OP-PRINTING IF =1
IGCRFG,0      /INHIBIT CR FACILITY IF =1
IPKTFG,0      /1=PACKET ARRIVING
CXFLG, 0      /1=ABORT CURRENT FILE
CZFLG, 0      /1=ABORT WHOLE FILE SERIES
/THE RS232 DUTY CYCLE
RSDUTY,0
      RSINF          /ANYTHING ON B-INPUT?
      ZERZ           /NO
      JMS RSBINP      /YES, DEAL WITH IT
      TTYRF          /ANYTHING FROM KB?
      ZERZ           /NO
      JMS TTYINP      /YES, DEAL WITH IT
      RSOUTF         /IS B-OUTPUT READY?
      ZERZ           /NO
      JMS RSBOP       /YES, SEND CHAR FROM OP-BUFFER
      TTYPF          /IS A-OUTPUT READY?
      ZERZ           /NO
      JMS TTYOP       /YES, PRINT CHAR FROM IP-BUFFER
      JMP @ RSDUTY
/INPUT ROUTINE FOR TTY KEYBOARD
/DO NOT CALL UNLESS TTYRF IS SKIPPING
/JUMP TO COMMAND BLOCK ON CTRL/G, NICBUG ON CTRL/B
/OTHERWISE, PUT CHAR IN OP-BUFFER IF KBFLG=0
/OR IN IP-BUFFER IF KBFLG=1
/BUT IF IGAFLG=1, JUST READ CHAR AND IGNORE IT
/HOLD LAST CHAR IN TTYCHA

```

```

/IF IN RECEIVE/TRANSMIT MODE, WATCH OUT FOR CR AND CTRL/Z
TTYINP,0
    RDTTY
    A-MZ ("G-100      /JUMP TO COMMD BLOCK ON BELL
    ZERZ
    JMP COMBAK
    A-MZ ("B-100      /NICBUG ON ^B
    ZERZ
    JMP @ BUGOUT
    MEMZ RTFLG        /R OT T MODE?
    ZERZ
    JMP NOTRT         /NO
    A-MZ ("X-100      /YES, ABORT CURRENT FILE
    ZERZ
    ONEM CXFLG
    A-MZ ("Z-100      /ABORT FILE SERIES
    ZERZ
    ONEM CZFLG
    A-MZ (215         /RESEND LAST PACKET ON CR
    JMP NOTRT
    MEMZ IGCRTG       /UNLESS WE ARE IGNORING THEM?
    JMP NOTRT         /WE ARE
    JMP RSEND
NOTRT, MEMZ IGAFLG    /IGNORE CHAR IF IGAFLG=1
    JMP @ TTYINP
    ACCM TTYCHA       /PRESERVE CHAR
    MEMZ KBFLG        /IS KBFLG SET?
    JMP KBPRN         /YES, SO CHAR TO IP-BUFFER
    JMS ACIBUF        /NO, SO CHAR TO OP-BUFFER
    JMP @ TTYINP
KBPRN, JMS ACIBUF
    JMP @ TTYINP
TTYCHA,0              /CURRENT TTY CHARACTER
/INPUT ROUTINE FOR RS232 CHANNEL-B
/DO NOT CALL UNLESS RSINF IS SKIPPING
/IF A CTRL/S, SET NULBFG=1; IF A CTRL/Q, SET NULBFG=0
/OTHERWISE, PUT CHAR INTO IP-BUFFER
/BUT IF IGBFLG=1, READ CHARACTER AND IGNORE IT
/IF IPKTFG IS SET, PASS CHAR TO INPUT PACKET BUFFER INSTEAD
/HOLD LAST CHAR IN RSBCHA
RSBINP,0
    RSIN
    A+MA (200         /ADD BIT-7 FOR NIC
    A-MZ ("S-100      /SET NULBFG ON CTRL/S
    ZERZ
    JMP WATSET
    A-MZ ("Q-100      /CLEAR NULBFG ON CTRL/Q
    ZERZ
    JMP WATCLR
    MEMZ IGBFLG       /IGNORE CHAR IF IGBFLG=1
    JMP @ RSBINP
    ACCM RSBCHA
    MEMZ IPKTFG       /INCOMING PACKET?
    JMP PKTING        /YES
    JMS ACIBUF        /NO, ADD TO IP-BUFFER
    JMP @ RSBINP
PKTING, JMS @ WAIPBU  /ADD TO IP-PACKET BUFFER
    JMP @ RSBINP
WATSET, ONEM NULBFG
    JMP @ RSBINP

```

```

WATCLR,ZERM NULBFG
      JMP @ RSBINP
RSBCHA,0          /CURRENT RS232-B CHARACTER
/RS232-B OUTPUT ROUTINE
/TRANSMIT CHARACTER FROM OP-BUFFER
/DO NOT CALL UNLESS RSOUTF IS SKIPPING
/IF OCHACT=0, EXIT WITHOUT DOING ANYTHING
/IF NULBFG=1, EXIT WITHOUT DOING ANYTHING
RSBOP, 0
      MEMZ OCHACT
      ZERZ
      JMP @ RSBOP
      MEMZ NULBFG
      JMP @ RSBOP
      JMS OBUFAC
      A-MA (200
      A+MM CHKSUM
      RSOUT
      JMP @ RSBOP
CHKSUM,0
CHKMSK,1777777
/RS232-A OUTPUT ROUTINE FOR TTY
/PRINT FROM IP-BUFFER
/DO NOT CALL UNLESS TTYPF IS SKIPPING
/IF ICHACT=0, EXIT WITHOUT DOING ANYTHING
/IF NULAFG=1, REMOVE CHARACTER FROM BUFFER BUT DON'T PRINT IT
/ONLY PRINT APPROVED CTRL/CHARACTERS
TTYOP, 0
      MEMZ ICHACT
      ZERZ
      JMP @ TTYOP
      JMS IBUFAC
      ANDA (377
      ACCM TTYHOL
      MEMZ NULAFG
      JMP @ TTYOP
      A-MA (240
      EXCT AC19
      JMP CTRLUK
      A+MA (240
TITBAK,PRTTY
      JMP @ TTYOP
CTRLUK,A+MA (240
      A-MZ (210
      ZERZ
      JMP TITBAK
      A-MZ (211
      ZERZ
      JMP TITBAK
      A-MZ (212
      ZERZ
      JMP TITBAK
      A-MZ (215
      ZERZ
      JMP TITBAK
      JMP @ TTYOP
TTYHOL,0
/TRANSFER CHARACTER FROM ACC TO OP-BUFFER
/UNLESS BUFFER IS ALREADY FULL
/BUMP POINTER, COUNTER

```

```

ACOBUF,0
    ACCM OPHOLD      /TEMPORARY STORE
    MEMA OCHACT      /IS BUFFER FULL?
    A-MA (777
    SKIP AC19
    JMP ACBOUT      /YES, JUMP OUT
    MEMA OPHOLD      /NO, ADD CHAR
    ACCM @ OPUTPT
    MPOM OCHACT      /BUMP COUNTER
    MPOMA OPUTPT     /AND POINTER
    A-MA TOPOP
    EXCT AC19        /HIT THE TOP?
    JMP ACBOUT      /NO, SO EXIT
    MEMA BOTOP       /YES, RESET POINTER
    ACCM OPUTPT
ACBOUT,MEMA OPHOLD   /RECOVER COPY OF CHAR
    JMP @ ACOBUF
/TRANSFER CHARACTER (IF ANY) FROM OP-BUFFER TO ACC
OBUFAC,0
    MEMZ OCHACT      /ANY CHARS?
    ZERZ
    JMP @ OBUFAC     /NO, JUMP OUT
    MEMA @ OPIKPT     /COLLECT CHAR
    ACCM OPHOLD       /HOLD IT
    MMOM OCHACT       /DECREMENT COUNTER
    MPOMA OPIKPT      /BUMP POINTER
    A-MA TOPOP
    SKIP AC19        /HIT THE TOP?
    JMP ROPBUF       /YES
    MEMA OPHOLD       /NO, SO RECOVER CHAR
    JMP @ OBUFAC     /AND EXIT
ROPBUF,MEMA BOTOP   /RESET POINTER
    ACCM OPIKPT
    MEMA OPHOLD       /RECOVER CHAR
    JMP @ OBUFAC     /AND EXIT
OPHOLD,0
OPUTPT,101000
OPIKPT,101000
OCHACT,0
BOTOP, 101000
TOPOP, 101777
/TRANSFER CHARACTER FROM ACC TO IP-BUFFER
/UNLESS BUFFER IS ALREADY FULL
/BUMP POINTER, COUNTER
ACIBUF,0
    A-MZ ("Q-100     /DO NOT PRINT CTRL/Q
    ZERZ
    JMP @ ACIBUF
    A-MZ ("A-100     /OR ^A
    ZERZ
    JMP @ ACIBUF
    ACCM IPHOLD      /TEMPORARY STORE
    MEMA ICHACT      /IS BUFFER FULL?
    A-MA (777
    SKIP AC19
    JMP @ ACIBUF     /YES, JUMP OUT
    MEMA IPHOLD      /NO, ADD THE CHAR
    ACCM @ IPUTPT
    MPOM ICHACT      /BUP THE COUNTER
    MPOMA IPUTPT     /AND POINTER

```

```

    A-MA TOPIP
    EXCT AC19          /HIT THE TOP?
    JMP @ ACIBUF       /NO, SO EXIT
    MEMA BOTIP        /RESET POINTER
    ACCM IPUTPT
    JMP @ ACIBUF
/TRANSFER CHARACTER (IF ANY) FROM IP-BUFFER TO ACC
/BUMP POINTER, DECREMENT COUNTER
IBUFAC,0
    MEMZ ICHACT        /ANY CHARS?
    ZERZ
    JMP @ IBUFAC       /NO, JUMP OUT
    MEMA @ IPIKPT      /COLLECT CHAR
    ACCM IPHOLD        /HOLD IT
    MMOM ICHACT        /DECREMENT COUNTER
    MPOMA IPIKPT       /BUMP POINTER
    A-MA TOPIP
    SKIP AC19          /HIT THE TOP?
    JMP RIPBUF         /YES
    MEMA IPHOLD        /NO, SO RECOVER CHAR

    JMP @ IBUFAC       /AND EXIT
RIPBUF, MEMA BOTIP     /RESET POINTER
    ACCM IPIKPT
    MEMA IPHOLD        /RECOVER CHAR
    JMP @ IBUFAC       /AND EXIT
IPHOLD,0
IPUTPT,100000
IPIKPT,100000
ICHACT,0
BOTIP, 100000
TOPIP, 100777
/WATCH OUT FOR CHARACTER IN ACC ON ENTRY
/EMPTIES IP-BUFFER BEFORE EXITING
WATCHA,0
    ACCM WACHOL
    JMS RSDUTY
    MEMA RSBCHA
    A-MZ WACHOL
    JMP #-3
    JMS EMPTIP
    ZERM RSBCHA
    JMP @ WATCHA
WACHOL,0
/LOOK OUT FOR EITHER OF TWO SPECIFIED CHARACTERS
/IN ADDRESSES FOLLOWING CALL
/EXITS WITH CHAR IN ACC, AFTER EMPTYING IP-BUFFER
WATCH2,0
    MEMA @ WATCH2
    ACCM CHAR1
    MPOM WATCH2
    MEMA @ WATCH2
    ACCM CHAR2
    MPOM WATCH2
NUWACH, JMS RSDUTY
    MEMA RSBCHA
    A-MZ CHAR1
    ZERZ
    JMP WOUT1
    A-MZ CHAR2

```

```

        ZERZ
        JMP WOUT2
        JMP NUWACH
WOUT1,  JMS EMPTIP
        ZERM RSBCHA
        MEMA CHAR1
        JMP @ WATCH2
WOUT2,  JMS EMPTIP
        ZERM RSBCHA
        MEMA CHAR2
        JMP @ WATCH2
CHAR1,  0
CHAR2,  0
/EMPTY BOTH TEXT BUFFERS
RSEMPT,0
        JMS EMPTIP
        JMS EMPTOP
        JMP @ RSEMPT
/EMPTY THE IP-BUFFER
EMPTIP,0
        JMS RSDUTY
        MEMZ ICHACT
        JMP #-2
        JMP @ EMPTIP
/EMPTY THE OP-BUFFER
EMPTOP,0
        JMS RSDUTY
        MEMZ OCHACT
        JMP #-2
        JMP @ EMPTOP
/RECEIVE AND TRANSMIT ROUTINES
/CURRENT MODE IS SHOWN BY RTFLG
/IF RTFLG=1, IN TRANSMIT MODE
/
        0   NOT IN CONTACT WITH REMOTE SYSTEM
/
        -1  IN RECEIVE MODE
RTFLG,  0
/
/RECEIVE FILES IN KERMIT PROTOCOL
/ASSUMES REMOTE MACHINE IS ALREADY SENDING INITIATION PACKETS
GETDAT,0
        ONEM CMDFLG
        JMS CRLF
        JMS @ WRTCLR      /CLEAR I/O BUFFERS
GDLOOP, JMS @ WSAVE       /SAVE MSA
        JMS @ WGETPA      /FIND A SPACE ON THE DISK
        JMS @ WDISSO      /RESTORE MSA
        JMS @ WPRGSE      /SET TYPROG COUNTER
        JMS @ WSARRO      /INITIALISE ARROW
        MEMZ RTFLG        /ALREADY TALKING?
        JMP GDNTRY        /YES
        MONM RTFLG        /SET RTFLG FOR RECEIVE
        JMS @ WRECIN      /NO, GET INITIATION PACKET
        JMS @ WRTCAL      /COMPUTE TIME-OUT PERIOD
        JMS @ WACKPK      /SEND AN ACK
        JMS @ WNAMGE      /WATCH FOR FILEHEADER PACKET
GDNTRY, JMS @ WFILSA      /SAY WHAT'S HAPPENING
        MEMZ @ WDIAFL     /IF DIAGNOSTICS ARE ON,
        JMS @ WPRGSE      /RESET TYPROG COUNTER
        JMS @ WRECDA      /RECEIVE AND STORE DATA ON DISK
        JMS @ WSAVE       /SAVE MSA

```



```

JMS @ WDIREN      /MAKE DIR ENTRY
JMS @ WDISO      /RESTORE MSA
GXZBAK, JMS @ WACKPK /SEND AN ACK
JMS @ WMORFI     /MORE FILES TO COME?
JMP GDLOOP      /YES, BACK FOR NEXT
JMS @ WACKPK     /NO, SO ACK THE BYE
GDEND, ZERM RTFLG /CLEAR RTFLG FOR NO R/T
JMP @ GETDAT
/TRANSMIT FILES IN KERMIT PROTOCOL
PUTDAT, 0
ONEM CMDFLG
JMS @ WRTCLR     /CLEAR I/O BUFFERS
NEWNAM, JMS CRLF
JMS UNPCK       /GET FILENAME FROM -KB
ENF
ONEM IGCRFG     /INHIBIT CR RESEND FACILITY
JMS @ WGETLI
ZERM IGCRFG     /ENABLE CR RESEND AGAIN
JMS @ WGETDA     /DIGEST REPLY
ZERM @ WNFILE    /ZERO FILE NUMBER
JMS @ WDIRLO     /LOAD DIR AT 2000 OR 3000 (D OR F)
NEWPUT, JMS @ WSARRO /INITIALISE ARROW
JMS @ WLOOKU     /LOOK UP DIR DATA, SAY WHAT'S HAPPENING
ZERZ
JMP SETLOD      /GOT A FILENAME MATCH
MEMZ RTFLG
JMP MOQERY      /SEE IF ANY MORE TO BE SENT
JMP NEWNAM      /TRY AGAIN
SETLOD, JMS @ WLODSE /SET THINGS UP FOR LOADING FILE
MEMZ @ WDIAFL   /IF DIAGNOSTICS ARE ON,
JMS @ WPRGSE    /RESET TYPROG COUNTER
MEMZ RTFLG     /ALREADY IN CONTACT?
JMP NEXFH      /YES
JMS @ WINISN    /NO, INITIATE CONTACT
ONEM RTFLG     /SET TRANSMISSION FLAG
JMS @ WRTCAL    /CALCULATE TIMEOUT REPEATS
NEXFH, JMS @ WFILHE /SEND FILENAME, GET ACK
CONPUT, JMS @ WNEXRE /PREPARE TO MOVE RECORD
JMS @ WLOAD     /LOAD <=1 RECORD
JMS @ WPKTOU    /SEND IT OFF AS PACKETS
MEMZ @ WLASFL   /LAST RECORD?
JMP FINPUT     /YES, FINISH OFF
JMP CONPUT     /NO, LOAD NEXT ONE
FINPUT, JMS @ WENDPK /SEND FILE-FINISHING Z-PKT
PXBK, MEMZ @ WJOKER /ANY JOKERS?
JMP NEWPUT     /YES, SEEK NEXT MATCH
MOQERY, JMS CRLF /NO
JMS CRLF
JMS UNPCK      /ANY MORE FILES TO SEND?
MORQ
JMS READ
A-MZ ("Y
ZERZ
JMP NEWNAM     /YES, ASK FOR NEXT FILENAME
JMS @ WCLOSI   /NO, SEND B-PACKET TO TERMINATE
ZERM RTFLG     /TRANSMISSION OVER
JMP @ PUTDAT
ENF, TEXT %ENTER NAME OF FILE(S) FOR TRANSMISSION%
MORQ, TEXT %ANY MORE FILES TO SEND (Y OR N)? %
/TRY TO ESCAPE FROM STUCK RECEIVE/TRANSMIT LOOP

```

```

/RESEND LAST PACKET, RETURN TO WAIT LOOP IN GETPKT
RSEND, JMS @ WPUTPK      /RESEND THE PACKET
      JMP @ WGPLP1      /LOOK FOR RESPONSE
/ABORT CURRENT TRANSMISSION, SEE IF MORE NEEDED
TABORT, JMS @ WDISCR     /SEND Z-PACKET WITH A D
      JMP MOQERY        /SEEK NEXT FILENAME
/SET KERMIT PARAMETERS
SETPAR, 0
      ONEM CMDFLG
      JMS CRLF
      JMS CRLF
      JMS UNPCK
      NOTYET
      JMS CRLF
      JMP @ SETPAR
NOTYET, TEXT %NOT YET AUTOMATED - USE NICBUG (:B) FOR NOW%
/READ CHARACTER FROM KEYBOARD
/PRINT IT IF NULAFG=0, BUT NOT IN NULAFG=1
/EMERGES WITH CHAR IN ACC
READ, 0
      ZERM IGAFLG      /TAKE NOTICE OF KB
      JMS EMPTIP      /COMPLETE IP-PRINTING
      ONEM KBFLG      /ROUTE TO PRINTER
      ZERM TTYCHA
      JMS TSET        /INITIALISE TIME-OUT COUNT
      JMS RSDUTY      /AWAIT CHAR
      JMS TCOUNT     /TIMED OUT YET?
      JMP TIMOUT      /YES
      MEMAZ TTYCHA
      ZERZ
      JMP #-5
      ACCM REDHOL     /HOLD IT
      JMS EMPTIP      /WAIT FOR IT TO BE PRINTED
      MEMA REDHOL     /RECOVER IT
      JMP @ READ
TIMOUT, JMS CRLF      /TIMED OUT
      JMS UNPCK
      TOUT
      JMS CRLF
      JMP COMBAK
REDHOL, 0
TOUT, TEXT %TIMED OUT%
/SECRET READ, WITHOUT PRINTING
SREAD, 0
      ONEM NULAFG
      JMS READ
      ZERM NULAFG
      JMP @ SREAD
/VERSION OF READ FOR COMMAND BLOCK, WITHOUT TIME-OUT FACILITY
CREAD, 0
      ONEM IGTOFG
      JMS READ
      ZERM IGTOFG
      JMP @ CREAD
/TIME-OUT COUNTER FOR INCLUSION IN WAIT-LOOPS SUCH AS READ
/TAKES 1ST RA WHEN TIMED OUT
/NORMALLY 2ND RA
/BUT CAN BE INHIBITED BY SETTING IGTOFG=1
TCOUNT, 0
      MEMZ IGTOFG      /INHIBITION?

```

```

        JMP TCOUT          /YES, JUMP STRAIGHT OUT
        MMOMZ TCNT         /COUNTED OUT?
        JMP TCOUT         /NO, NOT YET
        MEMA @ WRSANU      /YES, RESET COUNTER
        ACCM TCNT
        MMOMZ TNUM        /WAS IT LAST SESSION?
        JMP TCOUT         /NO
        JMP @ TCOUNT      /YES, 1ST RA
TCOUT,  MPOM TCOUNT      /TAKE 2ND RA
        JMP @ TCOUNT
TCNT,   0
TNUM,   0
IGTOFG,0                  /0-DO, 1=DON'T, DO TIME-OUT
/SET TIME-OUT COUNTERS
TSET,   0
        MEMA @ WRSANU
        ACCM TCNT
        MEMA (7
        ACCM TNUM
        JMP @ TSET
/TYPE CHARACTER IN ACC
/PROVIDED THAT NULAFG=0
TYPE,   0
        ACCM TYPHOL
        JMS ACIBUF
        JMS EMPTIP
        MEMA TYPHOL
        JMP @ TYPE
TYPHOL,0
/TRANSMIT CHARACTER IN ACC
/PROVIDED THAT NULBFG=0
RSTYPE,0
        ACCM BTYPHO
        JMS ACOBUF
        JMS EMPTOP
        MEMA BTYPHO
        JMP @ RSTYPE
BTYPHO,0
/TYPE CRLF, PROVIDED THAT NULAFG=0
CRLF,   0
        MEMA (215
        JMS TYPE
        MEMA (212
        JMS TYPE
        JMP @ CRLF
/TRANSMIT CRLF, PROVIDED THAT NULBFG=0
RSCRLF,0
        MEMA (215
        JMS RSTYPE
        MEMA (212
        JMS RSTYPE
        JMP @ RSCRLF
/TRANSMIT RETURN ONLY, NEXT TIME RSDUTY IS CALLED
QIKRET,0
        MEMA (215
        JMS ACOBUF
        JMP @ QIKRET
/SEND A RETURN, AND PAUSE
SLORET,0
        MEMA (215

```

```

        JMS RSTYPE
        MEMA SLONUM
        ACCM SLOCNT
        JMS RSDUTY
        MMOMZ SLOCNT
        JMP #-2
        JMP @ SLORET
SLONUM,10000
SLOCNT,0
/LOG IN ON SYM
LOGIN,0
        ONEM CMDFLG
        MEMA @ WUSRNA
        EXCT ZAC
        JMP UNFIGD
        ZERM IGBFLG
        ZERM IGAFLG
        ZERM KBFLG
        JMS RSEMPT
        JMS PADGET
        JMS SYMGET
        JMS SYMENT
        JMP DUMTRM
UNFIGD,JMS CRLF
        JMS UNPCK
        UF
        JMS CRLF
        JMP @ LOGIN
UF,      TEXT %ENTER "U" TO CONFIGURE%
/MAKE CONTACT WITH THE PAD
PADGET,0
        JMS SLORET
        JMS SLORET
        JMS SLORET
        JMS SLORET
        JMS QIKRET
        MEMA (">
        JMS WATCHA
        JMS RSEMPT
        JMP @ PADGET
/MAKE CONTACT WITH THE SYM
SYMGET,0
        JMS RSPCK
        CALSYM
        JMS QIKRET
        MEMA (":
        JMS WATCHA
        JMP @ SYMGET
CALSYM,TEXT %CALL LANCS.CENT1%
/ENTER USERNAME, ACCEPT PASSWORD, WAIT FOR ">" CHARACTER
SYMENT,0
        JMS @ WASCTR
        USRNAM
        MEMA (":
        JMS WATCHA
        JMS @ WASCTR
        PASWRD
        MEMA (">
        JMS WATCHA
        JMP @ SYMENT

```

```

/SWITCH OFF SYMMETRY ECHO
KILEKO,0
    ONEM CMDFLG
    JMS OPENB
    JMS QIKRET
    MEMA (">
    JMS WATCHA
    JMS RSPCK
    ECHOFF
    JMS QIKRET
    MEMA (">
    JMS WATCHA
    JMP @ KILEKO
ECHOFF,TEXT %STTY -ECHO%
/SWITCH ECHO ON AGAIN
MAKEKO,0
    ONEM CMDFLG
    JMS OPENB
    JMS QIKRET
    MEMA (">
    JMS WATCHA
    JMS RSPCK
    ECHON
    JMS QIKRET
    MEMA (">
    JMS WATCHA
    JMP @ MAKEKO
ECHON, TEXT %STTY ECHO%
/LOG-OFF, EXIT TO MONITOR
LOGOUT,0
    ONEM CMDFLG
    ZERM IGBFLG
    ZERM NULBFG
    MEMA ("Q-100
    JMS RSTYPE
    JMS QIKRET
    MEMA (">
    JMS WATCHA
    JMS RSPCK
    LO
    JMS QIKRET
    MEMA ("*
    JMS WATCHA
    JMS EMPTIP
    JMP @ MONOUT
LO, TEXT %LOGOUT%
/TOGGLE THE B-INPUT ON/OFF
BTOG, 0
    ONEM CMDFLG
    JMS CRLF
    JMS UNPCK
    RBI
    MEMZ IGBFLG
    JMP BTOFF
    JMS UNPCK
    RBON
    JMS OPENB
    JMP BTGOUT
BTOFF, JMS UNPCK
RBOFF

```

```

        JMS SHUTB
BTGOUT, JMS CRLF
        JMP @ BTOG
RBI,    TEXT %RS232 B-INPUT %
RBON,   TEXT %OPEN%
RBOFF,  TEXT %SHUT%
/SHUT DOWN THE B-INPUT
SHUTB,  0
        JMS RSEMPT
        MEMA ("S-100
        JMS RSTYPE
        ONEM IGBFLG
        JMP @ SHUTB
/OPEN UP THE B-INPUT AGAIN
OPENB,  0
        JMS RSEMPT
        ZERM IGBFLG
        MEMA ("Q-100
        JMS RSTYPE
        JMP @ OPENB
/UNPACK 6-BIT TEXT AND TYPE
UNPCK,  0
        MEMA @ UNPCK
        ACCM UNP100
        MPOM UNPCK
UNP200, MEMA @ UNP100
        JMS UNP300
UNPBAK, MPOM UNP100
        JMP UNP200
UNP300, 0
        ACCM UNP400
        RASH 14
        JMS UNP500
        RASH 6
        JMS UNP500
        JMS UNP500
        JMP @ UNP300
UNP500, 0
        ANDA (77
UPEXIT, A-MZ (77
        ZERZ
        JMP UNPOUT
        A+MA (240
        JMS ACIBUF
        MEMA UNP400
        JMP @ UNP500
UNPOUT, JMS EMPTIP
        JMP @ UNPCK
UNP100, 0
UNP400, 0
/VERSION OF UNPCK FOR FILENAME PRINTING
/JUMPS OUT ON FINDING A SPACE, INSTEAD OF A 77
FPCK,   0
        MEMA @ FPCK
        ACCM FPCADD
        MPOM FPCK
        MEMA INSSP
        ACCM UPEXIT
        JMS UNPCK
FPCADD, 0

```

```

        MEMA INS77
        ACCM UPEXIT
        JMP @ FPCK
INSSP, A-MZ (0
INS77, A-MZ (77
/UNPACK 6-BIT TEXT DOWN RS232-B OUTPUT
/CONVERTING TO LOWER-CASE
RSPCK, 0
        MEMA @ RSPCK
        ACCM RSP100
        MPOM RSPCK
RSP200, MEMA @ RSP100
        JMS RSP300
        MPOM RSP100
        JMP RSP200
RSP300, 0
        ACCM RSP400
        RASH 14
        JMS RSP500
        RASH 6
        JMS RSP500
        JMS RSP500
        JMP @ RSP300
RSP500, 0
        ANDA (77
        A-MZ (77
        ZERZ
        JMP RSPOUT
        A-MA (40
        EXCT AC19 ZAC
        A-MA (40
        A+MA (340
        JMS ACOBUF
        MEMA RSP400
        JMP @ RSP500
RSPOUT, JMS EMPTOP
        JMP @ RSPCK
RSP100, 0
RSP400, 0
/DISK POINTERS
WDISSO, 0
MONOUT, 7600
/PAGE 106000 POINTERS
WMORFI, MORFIN
WRECIN, RECINI
WNAMGE, NAMGET
WFILSA, FILSAY
WRECDA, RECDAT
/PAGE 112000 POINTERS
WDIAFL, DIAFLG
WRSANU, RSANUM
WARCTO, ARCTOG
WRTCLR, RTCLR
WINISN, INISND
WACKPK, ACKPKT
WSIDCO, SIDCON
WFILHE, FILHED
WPKTOU, PKTOUT
WENDPK, ENDPKT
WCLOSI, CLOSIT

```

```

WPUTPK,PUTPKT
WGPLP1,GPLUP1
WDISCR,DISCRD
WAIPBU,AIPBUF
WPRGSE,PRGSET
WRTCAL,RTCALC
/PAGE 114000 POINTERS
WGETLI,GETLIN
WGETDA,GETDAB
WASFLA,ASFLAG
WNFILE,NFILES
WLOOKU,LOOKUP
WJOKER,JOKER
BUGOUT,114700
/PAGE 116000 POINTERS
WSARRO,SARROW
WUSRNA,USRNAM
WHELP,HELP
WNOTPR,NOTPRN
WDIRLO,DIRLOD
WLODSE,LODSET
WNEXRE,NEXREC
WLOAD,LOAD
WLASFL,LASFLG
WINTRO,INTROD
WCONFI,CONFIG
WASCTR,ASCTRN
WSAVE,SAVE
WDIREN,DIRENT
WGETPA,GETPAD
WDIRLS,DIRLST
WDISKI,DISKID
PAGSKP
*104000
/PAGE 104000 HOLDS TEXT FOR COMMAND SUMMARY ETC
/KERMIT COMMAND SUMMARY
COMSUM,TEXT %      ***KERMIT COMMAND SUMMARY***%
TEXT %B - CALL NICBUG%
TEXT %C - CONNECT (DUMB TERMINAL MODE)%
TEXT % ^G - RETURN TO COMMAND BLOCK%
TEXT %D - DIRECTORY LISTING%
TEXT %E - SET EXTERNAL ECHO%
TEXT %H - HELP! PRINT COMMAND SUMMARY%
TEXT %I - TOGGLE INDICATOR-PRINTING & PACKET-ARCHIVING ON/OFF%
TEXT %K - KILL EXTERNAL ECHO%
TEXT %L - LOGIN ON REMOTE SYSTEM%
TEXT %M - QUIT, RETURN TO DEMON MONITOR%
TEXT %N - PRINT NOTES ON OPERATION%
TEXT %P - SET KERMIT PARAMETERS%
TEXT %Q - QUIT, RETURN TO DEMON MONITOR%
TEXT %R - RECEIVE DATA, FILE ON DISK%
TEXT % ^G - UNCONDITIONAL RETURN TO COMMAND BLOCK%
TEXT % ^X - ABORT TRANSFER OF CURRENT FILE%
TEXT % ^Z - ABORT TRANSFER OF WHOLE FILE SERIES%
TEXT % CR - RETRANSMIT LAST PACKET, AND CONTINUE%
TEXT %S - SEND FILE(S) TO REMOTE SYSTEM%
TEXT % (WITH ^G,^X,^Z,CR, SAME AS FOR R-COMMAND)%
TEXT %T - TOGGLE RS232 B-INPUT ON/OFF%
TEXT %U - CONFIGURE USERNAME AND PASSWORD%
740000

```



```

NOTES, TEXT %                               ***NOTES***%
TEXT %1. OPERATION IS MORE RELIABLE AFTER A MINOR MODIFICATION TO THE RS-232%
TEXT %PCB IN THE NIC-80 (SEE INSTRUCTIONS).%
TEXT % %
TEXT %2. TO AVOID BUFFER-OVERFLOW AND LOSS OF CHARACTERS WHEN A LOT OF DATA%
TEXT %ARRIVES FAST IN TERMINAL-EMULATOR MODE (:C), THE RS-232 A-CHANNEL SHOULD%
TEXT %BE RUN AT A BAUD RATE EQUAL TO (OR FASTER THAN) THE B-CHANNEL RATE.%
TEXT % %
TEXT %3. INDICATOR-PRINTING AND PACKET-ARCHIVING (:I) IS TO ASSIST DE-BUGGING:%
TEXT %THE PREVIOUS 3 OR 4 PACKETS ARE PRESERVED AT OCTAL 200 INTERVALS IN%
TEXT %102000-102777 (OUTPUT PACKETS) AND 103000-103777 (INPUT), FOR INSPECTION%
TEXT %WITH NICBUG (:B), AND THE INDICATOR SYMBOLS ARE PRINTED DURING OPERATION.%
TEXT % %
TEXT %4. WITH INDICATOR-PRINTING (:I) SELECTED, THE (MOSTLY LOWER-CASE)%
TEXT %SYMBOLS FOR OUTGOING PACKETS ARE -%
TEXT %      S - PACKET SENT%
TEXT %      S - (UPPER-CASE) SAME PACKET RE-SENT%
TEXT %INDICATOR SYMBOLS FOR INCOMING PACKETS ARE -%
TEXT %      R - GOOD (UNCORRUPTED) PACKET RECEIVED%
TEXT %      C - CHECKSUM ERROR%
TEXT %      L - LENGTH ERROR%
TEXT %      N - NAK RECEIVED (NOT PRINTED IF SEQ.NO. ALSO WRONG)%
TEXT %      N - (UPPER-CASE) NAK TO *NEXT* PACKET RECEIVED%
TEXT %      T - TIME-OUT, AWAITING PACKET%
TEXT %      T - (UPPER-CASE) TIME-OUT, DURING RECEPTION OF PACKET%
TEXT %      W - WRONG SEQUENCE NUMBER%
TEXT % %
TEXT %5. IT IS ASSUMED THAT THE NIC-80 IS CONTROLLED FROM A VDU TERMINAL,%
TEXT %NOT AN ORIGINAL ASR33 TELETYPE (WHICH CANNOT PRINT THE LOWER-CASE%
TEXT %INDICATOR SYMBOLS).%
TEXT % %
740000
PAGSKP
*106000
/PAGE 106000 HOLDS THE MAIN WORK ROUTINES FOR RECEIVING DATA
/USES V-PREFIX POINTERS
/
/RECEIVE PACKETS AND STORE AS DATA FILES ON DISK
/ASSUMES THAT A FILE-HEADER HAS ALREADY ARRIVED AND THAT A SPACE
/HAS BEEN FOUND ON THE DISK
RECDAT,0
    ZERM ENDFLG      /CLEAR FLAGS
    ZERM NEFLG
    ZERM LCRFLG
    ZERM DOLFLG
    ZERM SBFLG
    JMS CLRDB        /CLEAR DISK BUFFER
    ZERM DBPNT        /SET POINTER
    MEMA @ VRECSI     /SET RECORD WORD-COUNTER
    ACCM DBCNT
    ACCM @ VRITWR     /AND STORE SIZE
    ZERM @ VWRDST     /ZERO WORDS STORED
    MEMZ @ VASFLA     /IS IT ASCII OR NUMERICAL?
    JMP RDASCI        /ASCII
    JMS @ VACKPK       /NUMERICAL, SEND AN ACK
    JMS DATGET        /AND GET FIRST DATA PACKET
    JMS INRPIP        /INITIALISE PIP POINTER, COUNTER
    JMS FDGET         /SCOOP & STORE FILE DIR DATA
RDNLUP, JMS BINMOV    /MOVE NUMBER TO BINBUF (&RELOAD)
    JMS BINMAK        /CONVERT TO INTEGER (&STORE)

```

```

        JMS BMPDB           /CHECK POSITION, STORE IF READY
        JMP RDNLUP          /BACK FOR NEXT
        JMP @ RECDAT        /ENDFLG=1, LAST BUFFER STORED
RDASCI, JMS @ VACKPK        /SEND AN ACK
        JMS DATGET          /GET FIRST DATA PACKET
        JMS INRPIP          /INITIALISE PIP POINTER, COUNTER
        JMS FDGET           /SCOOP & STORE FILE DIR DATA
RDALUP, JMS PNTMOV          /MOVE QUINT TO PENTUM (&RELOAD)
        JMS PAKDWN          /TRANSFER AS PACASC
        JMS BMPDB           /CHECK POSITION, STORE IF READY
        JMP RDALUP          /BACK FOR MORE
        JMP @ RECDAT        /ENDFLG=1, LAST BUFFER STORED
/POINTERS, COUNTER, FOR PACKET IN OUTPUT BUFFER
ROPAKS=102000
ROMARK,ROPAKS              /FOR ^A MARK
ROLEN, ROPAKS+1            /LENGTH
ROSEQ, ROPAKS+2            /SEQUENCE NO.
ROTYPE,ROPAKS+3            /PACKET TYPE
RODATS,ROPAKS+4            /START OF DATA
ROPKPT,0                   /POINTER
ROPKCT,0                   /COUNTER
/POINTERS, COUNTER, FOR PACKET IN INPUT BUFFER
RIPAKS=103000
RIMARK,RIPAKS              /FOR ^A MARK
RILEN, RIPAKS+1            /LENGTH
RISEQ, RIPAKS+2            /SEQUENCE NO.
RITYPE,RIPAKS+3            /PACKET TYPE
RIDATS,RIPAKS+4            /START OF DATA
RIPKPT,0                   /POINTER
RIPKCT,0                   /COUNTER
/GENERAL POINTERS, VARIABLES, FOR RECEIVE MODE
DBPNT, 0                   /DISK-BUFFER POINTER
DBCNT, 0                   /DISK-BUFFER COUNTER
ENDFLG,0                   /=1 WHEN Z-PACKET ARRIVES
NEFLG, 0                   /=1 WHEN ASCII REACHES (RECSIZ-130)
LCRFLG,0                   /=1 WHEN LAST CR ADDED
DOLFLG,0                   /=1 WHEN DOLLAR ADDED
SBFLG, 0                   /=1 WHEN ASCII BUFFER STORABLE
RTRYCT,0                   /TRY COUNTER FOR RECEIVE
/INITIALISE POINTER, COUNTER, FOR READING DATA
INRPIP,0
        MEMA RIDATS
        ACCM RIPKPT
        MEMA @ RILEN
        A-MA (243
        ACCM RIPKCT
        JMP @ INRPIP
/CLEAR THE DISK BUFFER
CLRDB, 0
        ZERM CLDBPT
        MEMA @ VRECSI
        ACCM CLDBCT
CLDLUP,ZERM @ CLDBPT
        MPOM CLDBPT
        MMOMZ CLDBCT
        JMP CLDLUP
        JMP @ CLRDB
CLDBPT,0
CLDBCT,0
/TEST PROGRESS

```

```

/BUMP DISK BUFFER & STORE IF FULL, NORMALLY 1ST RA
/IF ENDFLG=1, DO FINAL STORE, TAKE 2ND RA
/WITH ASCII (ASFLAG=1), SET NEFLG WHEN 130 SHORT OFBUFFER TOP
/AND STORE THE BUFFER WHEN SBFLG SETS
/NOTE THAT, FOR ASCII, THE BUMPING IS DONE BY PAKDWN
BMPDB, 0
    MEMZ @ VASFLA    /ASCII OR NUMERICAL?
    JMP BMPASC       /ASCII
    MPOM DBPNT
    MPOM @ VARWDS
    MMOM DBCNT
    MEMZ ENDFLG      /NUMERICAL, ALL DONE?
    JMP LASBIN       /YES
    MEMA DBCNT       /NO, ANY SPACE LEFT?
    SKIP ZAC
    JMP @ BMPDB      /YES, TAKE 1ST RA
STOBUF, JMS @ VWRITE /NO, SO STORE BUFFER
    JMS CLRDB       /CLEAR THE AREA
    ZERM DBPNT      /RESET POINTER
    MEMA @ VRECSI   /AND COUNTER
    ACCM DBCNT
    JMP @ BMPDB      /TAKE 1ST RA
LASBIN, MEMA DBPNT  /STORE WHAT WE HAVE
    AMOM @ VRITWR
BDBOUT, JMS @ VWRITE /WRITE BUFFER TO DISK
    MPOM BMPDB      /TAKE 2ND RA
    JMP @ BMPDB
BMPASC, MMOZ SBFLG  /READY TO STORE ASCII BUFFER?
    JMP NETST       /NO, CHECK POSITION
    MEMZ ENDFLG     /YES, IS IT LAST ONE?
    JMP BDBOUT      /YES, FINISH OFF
    ZERM SBFLG      /NO, CLEAR FLAGS
    ZERM NEFLG
    ZERM LCRFLG
    JMP STOBUF      /AND GO AND STORE BUFFER
NETST, MEMA DBCNT   /NEAR END OF BUFFER?
    A-MA (130
    EXCT AC19 ZAC
    ONEM NEFLG      /YES, SET NEAR-END FLAG
    JMP @ BMPDB     /TAKE 1ST RA
/COLLECT AND STORE THE DIRECTORY DATA FOR THE FILE
/ASSUMED TO BE THREE DECIMAL NUMBERS AT FRONT OF FIRST PACKET
/SIZE, LOAD ADDRESS, STARTING ADDRESS
FDGET, 0
    MEMA @ RIDATS    /IS 1ST CHAR A DIGIT?
    JMS DIGCHK
    ZERZ
    JMP @ FDGET      /NO, JUMP OUT
    MEMA VSIZE       /SET POINTER
    ACCM DBPNT
    MEMA (3          /SET COUNTER
    ACCM FDGCNT
FDGLUP, JMS BINMOV  /DECIMAL-OCTAL CONVERSION LOOP
    JMS BINMAK
    MPOM DBPNT
    MMOMZ FDGCNT
    JMP FDGLUP
    ZERM DBPNT      /RESET POINTERS
    MEMA (5
    A+MM RIPKPT

```

```

M-AM RIPKCT
MEMA @ VSIZE      /INITIALISE ARROW
ACCM @ VWRDTO
ZERM @ VARWDS
JMP @ FDGET
FDGCNT,0
/MOVE ONE DECIMAL INTEGER FROM PIP-BUFFER TO BIN BUFFER
/IGNORE LEADING ZEROES, DIGITS AFTER A DECIMAL POINT, AND NON-NUMERIC
/CHARACTERS WHICH ARE ASSUMED TO BE SEPARATORS
BINMOV,0
MEMA BINST          /CLEAR BINBUF BUFFER
ACCM BINPT
MEMA (7
ACCM BINCT
BMOV1, ZERM @ BINPT
MPOM BINPT
MMOMZ BINCT
JMP BMOV1
MEMA BINST          /RESET BIN POINTER, COUNTER
ACCM BINPT
MEMA (7
ACCM BINCT
BMOV2, MEMA @ RIPKPT /FIND FIRST DIGIT, OR MINUS
JMS DIGCHK          /EXAMINE CHAR
JMP BMOV3            /GOT ONE
JMS PIPBMP           /BUMP PACKET (OR GET NEXT)
JMP BMOV2            /RETRY
BMOV3, ACCM @ BINPT  /DEPOSIT CHAR (DIGIT OR -)
JMS PIPBMP           /BUMP PACKET (OR GET NEXT)
MPOM BINPT           /BUMP BIN POINTER
MMOMA BINCT          /SAFETY CHECK
EXCT AC19
JMP BMOTP            /TOO MANY DIGITS
MEMA @ RIPKPT        /LOOK AT NEXT CHAR
JMS DIGCHK           /IS IT A DIGIT?
JMP BMOV3            /YES, GRAB IT
JMP @ BINMOV         /NO, JOB FINISHED
BMOTP, MEMA BINST     /ENSURE BINT IS MAXIMUM INTEGER
ACCM BINPT
MEMA @ BINPT
A-MZ (" -           /OF APPROPRIATE SIGN
ZERZ
MPOM BINPT
MEMA ("9
ACCM @ BINPT
OTPOUT, MEMA @ RIPKPT /MOVE ALONG TO A NON-DIGIT
JMS DIGCHK
ZERZ
JMP @ BINMOV         /FOUND ONE, EXIT
JMS PIPBMP
JMP OTPOUT
/CHECK CHAR IN ACC ON ENTRY
/IF A DIGIT OR MINUS, TAKE 1ST RA WITH CHAR IN ACC
/OTHERWISE, IF NON-NUMERIC, TAKE 2ND RA
DIGCHK,0
ACCM DCHOLD
A-MA (260
EXCT AC19
JMP MINCHK
MEMA DCHOLD

```

```

A-MA (272
EXCT AC19
JMP NUMCHA
JMP NONNUM
NUMCHA, MEMA DCHOLD
JMP @ DIGCHK
NONNUM, MPOM DIGCHK
MEMA DCHOLD
JMP @ DIGCHK
MINCHK, MEMA DCHOLD
A-MZ (255
JMP NONNUM
JMP NUMCHA
DCHOLD, 0
/RE-CREATE A BINARY INTEGER FROM ASCII DECIMAL NUMBER
/NUMBER TO BE IN BINBUF, STARTING AT THE BEGINNING
/FIRST ADDRESS MAY CONTAIN A MINUS SIGN OR A DIGIT
/IF FIRST ADDRESS CONTAINS A ZERO, THE NUMBER IS TREATED AS ZERO
/ROUTINE DEPOSITS RESULT AT DBPNT, AND EXITS WITH IT IN ACC
BINMAK, 0
MEMA DESRT          /INITIALISE THINGS
ACCM DEPNT
ZERM BINT
ZERM BINSIN
MEMA @ BINST        /IS NUMBER ZERO?
A-MA (260
EXCT ZAC
JMP NULFIN          /YES, SO EXIT
MEMA @ BINST        /NO, IS NUMBER NEGATIVE?
A-MA (255
EXCT ZAC
ONEM BINSIN         /YES, SET FLAG
MEMA BINEND         /FIND LOWEST DIGIT
ACCM BINPT
MEMA (7
ACCM BINCT
BMLUP1, MEMAZ @ BINPT /SEARCH LOOP
JMP GOTDIG
MMOM BINPT
MMOMZ BINCT
JMP BMLUP1
GOTDIG, JMS UNASCI   /REMOVE ASCII BIAS, TEST
JMP NULFIN          /ERROR, TREAT AS ZERO
ACCM BINT           /OK, UNITS
BMLUP2, MMOM BINPT
MEMA @ BINPT
JMS UNASCI          /REMOVE ASCII BIAS, TEST
JMP BINFIN          /NON-DIGIT, FINISH OFF
ACCM BINMUL         /OK, MULTIPLY BY DEC EQUIVT
MEMA @ DEPNT
TACMQ
MULT
BINMUL, 0
SKIP ZAC            /TEST FOR OVERFLOW
JMP MAXFIN          /IT HAS
TMQAC              /OK
A+MMA BINT         /ADD RESULT TO BINT
EXCT AC19          /HAS BIT-19 SET?
JMP MAXFIN          /YES
MPOM DEPNT         /BUMP DEC EQUIVT POINTER

```

JMP BMLUP2	/AND BACK FOR NEXT DIGIT
BINFIN, MEMA BINT	/FINISH OFF, ADJUST SIGN?
MAXOUT, MEMZ BINSIN	
ANGA	
MNOUT, ACCM BINT	/LEAVE RESULT IN BINT
ACCM @ DBPNT	/AND DBPNT
JMP @ BINMAK	
NULFIN, ZERAM BINT	/LEAVE ZERO IN BINT, ACC, DBPNT
ACCM @ DBPNT	
JMP @ BINMAK	
MAXFIN, MEMA MAXPOS	/OVERFLOW, SO SET MAX INTEGER
MMOZ BINSIN	/CHECKING SIGN
JMP MAXOUT	
MEMA MAXNEG	
JMP MNOUT	
BINST, BINBUF	/ST OF BUFFER FOR BINARY CONVERSION
BINEND, BINBUF+6	/POINTER TO END
BINBUF, BLOCK 7	
BINPT, 0	/POINTER
BINCT, 0	/COUNTER
BINT, 0	/OCTAL BESULT
BINSIN, 0	/1=NEGATIVE NUMBER
MAXPOS, 1777777	/MAX NO. FOR OVERFLOWS
MAXNEG, 2000000	
DESRT, DELST	/DECIMAL EQUIVALENTS LIST
DEPNT, 0	/DE POINTER
DELST, 12	/10
144	/100
1750	/1000
23420	/10000
303240	/100000
/MOVE 5 CHARS FROM PIP-BUFFER TO PENTUM BUFFER	
/IF <5 AVAILABLE, GET NEXT PACKET, RESET POINTER	
/PROCESS CONTROL-PREFIXES, BUT IGNORE ANY LFS	
/WHEN ENDFLG=1, ADD DOLLAR, SET DOLFLG, AND FILL REST OF PENTUM WITH FFS	
/IF DOLFLG ALREADY SET, JUST FF-FILL THE WHOLE PENTUM	
PNTMOV, 0	
JMS PNTSET	/CLEAR PENTUM BUFFER, SET POINTER AND COUNTER
MEMZ DOLFLG	/DOLLAR ALREADY SET?
JMP PNTFF	/YES, JUST FF-FILL
PNTLUP, MEMZ ENDFLG	/END OF DATA?
JMP PNTDOL	/YES, SET DOLLAR AND FFS
MEMA @ RIPKPT	/GET A CHAR
A-MZ @ VRQCTL	/IS IT THE CTRL PREFIX?
JMP NONCTL	/NO
JMS PIPBMP	/YES, GET NEXT CHAR
MEMA @ RIPKPT	
A-MZ @ VRQCTL	/ANOTHER PREFIX?
ZERZ	
JMP NONCTL	/YES, TREAT AS NORMAL
A-MZ ("J	/IS IT AN LF?
JMP CLCHA	/NO, TREAT AS CTRL/CHA
JMS PIPBMP	/YES, IGNORE IT
JMP PNTLUP	
CLCHA, A-MA (100	/REMOVE CTRL BIAS
A-MZ (215	/IS IT A CR?
JMP NONCTL	/NO
MEMZ NEFLG	/YES, IS NEFLG SET?
ZERZ	
JMP NONCTL	/NO, JUST CARRY ON

```

        ACCM @ PENTPT      /YES, DEPOSIT THE CR
        MPOM PENTPT        /BUMP POINTERS
        JMS PIPBMP
        ONEM LCRFLG        /SET THE LAST-CR FLAG
        JMP FFLOOP         /AND START FF-FILLING IF POSS
NONCTL, ACCM @ PENTPT      /DEPOSIT CHAR IN PENTUM
        MPOM PENTPT        /BUMP POINTERS
        JMS PIPBMP
        MMOMZ PENTCT       /GOT FIVE YET?
        JMP PNTLUP         /NO, BACK FOR NEXT
        JMP @ PNTMOV        /YES, ALL DONE, JUMP OUT
PNTDOL, MEMA (244         /DEPOSIT DOLLAR TO CLOSE
        ACCM @ PENTPT
        ONEM DOLFLG       /SET DOLLAR FLAG
        JMP FFLOOP
PNTFF, MEMA (214          /DEPOSIT FFS TO FINISH RECORD
        ACCM @ PENTPT
        MPOM PENTPT
        ONEM SBFLG        /SET STORE-BUFFER FLAG
FFLOOP, MMOMZ PENTCT      /GOT FIVE YET?
        JMP PNTFF         /NO, GET ANOTHER FF
        JMP @ PNTMOV        /YES, BUFFER FINISHED
/BUMP POINTER, DECREMENT COUNTER, FOR PIP-BUFFER
/ENTER WITH LAST CHAR IN ACC
/IF DATA PACKET FINISHED, LOAD NEXT ONE
/IF NEXT IS A Z-PACKET, DATGET WILL SET ENDFLG
PIPBMF, 0
        MPOM RIPKPT       /BUMP PIP POINTER
        MMOMZ RIPKCT      /END OF PACKET?
        JMP @ PIPBMP       /NO, EXIT
        JMS @ VACKPK       /YES, SEND AN ACK
        JMS DATGET         /AND GET NEXT ONE
        JMS INRPIP        /RESET POINTER, COUNTER
        JMP @ PIPBMP
/CLEAR PENTUM BUFFER AND SET PENTUM POINTERS
PNTSET, 0
        MEMA PENTST
        ACCM PENTPT
        MEMA (5
        ACCM PENTCT
PSLUP, ZERM @ PENTPT
        MPOM PENTPT
        MMOMZ PENTCT
        JMP PSLUP
        MEMA PENTST
        ACCM PENTPT
        MEMA (5
        ACCM PENTCT
        JMP @ PNTSET
/CREATE 2 WORDS OF PACASC FROM 5 ASCII CHARACTERS IN PENTUM BUFFER
/DEPOSIT THEM IN DISK BUFFER WITH DBPNT POINTER
PAKDWN, 0
        MEMA PENTST
        ACCM PENTPT
        MEMA @ PENTPT      /CHAR-1
        MPOM PENTPT
        RLSH 10
        ACCM @ DBPNT
        MEMA @ PENTPT      /CHAR-2
        MPOM PENTPT

```

```

LASH 4
A+MM @ DBPNT
MEMA @ PENTPT /CHAR-3
MPOM PENTPT
ACCM PAKHOL
RISH 4
ANDA (17
A+MM @ DBPNT
MPOM DBPNT /START ON SECOND WORD
MPOM @ VARWDS
MMOM DBCNT
MEMA PAKHOL
ANDA (17
RLSH 4
ACCM @ DBPNT
MEMA @ PENTPT /CHAR-4
MPOM PENTPT
LASH 10
A+MM @ DBPNT
MEMA @ PENTPT /CHAR-5
MPOM PENTPT
A+MM @ DBPNT
MPOM DBPNT
MPOM @ VARWDS
MMOM DBCNT
JMP @ PAKDWN
PAKHOL, 0
PENTUM, BLOCK 5 /5-CHAR BUFFER FOR PAKDWN
PENTST, PENTUM
PENTPT, 0
PENTCT, 0
/ROUTINE TO STRIP ASCII BIAS FROM CHAR IN ACC AND TEST IT
/IF NON-NUMERICAL, FIRST RA
/NORMALLY, 2ND RA WITH RESULT IN ACC
UNASCI, 0
A-MA (260 /STRIP BIAS
ACCM UNANUM /HOLD RESULT
EXCT AC19
JMP @ UNASCI /NON-NUMERIC
A-MA (12
SKIP AC19
JMP @ UNASCI /NON-NUMERIC
MPOM UNASCI /OK, GOOD DIGIT
MEMA UNANUM
JMP @ UNASCI
UNANUM, 0
/LOOK OUT FOR INITIATION S-PACKET
/STORE RELEVANT DATA
RECINI, 0
MEMA (240 /INITIALISE SEQ.NO.
ACCM @ VSEQNO
JMS @ VINPOT /GET A PACKET
A-MZ ("S /IS IT S-TYPE?
JMP BADRIP /NO
JMS @ VSIDCO /YES, COLLECT THE DATA
JMP @ RECINI
BADRIP, MEMA ("S /ABORT WITH AN S
JMS @ VABORT
/LOOK FOR AN F-TYPE PACKET WITH THE FILENAME
/CONVERT IT TO 6-BIT AND STORE

```



```

NAMGET,0
    JMS @ VINPUT      /GET A PACKET
    A-MZ ("F          /IS IT F-TYPE?
    JMP BADNG         /NO
    JMS MAKFN         /YES, CONVERT FILENAME
    JMP @ NAMGET
BADNG, MEMA ("F      /ABORT WITH AN F
    JMS @ VABORT
/SEE IF THERE ARE MORE FILES TO COME
/IF AN F-PACKET ARRIVES, SCOOP UP FILENAME AND TAKE 1ST RA
/IF A B-PACKET ARRIVES, TAKE 2ND RA
MORFIN,0
    JMS @ VINPUT      /GET A PACKET
    A-MZ ("B          /IS IT A BYE?
    ZERZ
    JMP BYESYM        /YES
    A-MZ ("F          /IS IT A FILEHEADER PACKET?
    ZERZ
    JMP NEXFIL        /YES
    MEMA ("B          /ABORT WITH A B
    JMS @ VABORT
BYESYM,MPOM MORFIN   /END OF SESSION, 2ND RA
    JMP @ MORFIN
NEXFIL,JMS MAKFN     /COLLECT FILENAME
    JMP @ MORFIN      /AND 1ST RA
/LOOK FOR A DATA PACKET
DATGET,0
    JMS @ VINPUT      /GET A PACKET
    A-MZ ("D          /IS IT D-TYPE?
    ZERZ
    JMP @ DATGET      /YES, EXIT
    A-MZ ("Z          /IS IT Z-TYPE?
    JMP BADDG         /NO
    ONEM ENDFLG       /YES, SET END-OF-FILE FLAG
    MEMA @ RILEN      /ANYTHING IN DATA FIELD?
    A-MZ (243
    ZERZ
    JMP @ DATGET      /NO, KEEP THE FILE
    MEMA @ RIDATS     /YES, IS IT A D-DISCARD?
    A-MZ ("D
        JMP @ DATGET  /NO, SO KEEP THE FILE
    JMP @ VGXZBA      /YES, A D, SO DISCARD IT
BADDG, MEMA ("D      /ABORT WITH A D
    JMS @ VABORT
/SEND ACK WITH INITIALISATION DATA
RINACK,0
    JMS @ VINISN      /SAME AS FOR S, BUT NOT SENT
    MEMA ("Y
    ACCM @ ROTYPE     /CHANGE TYPE TO Y
    JMS @ VCOMPX      /COMPLETE THE PACKET
    ZERM @ VRETFL
    JMS @ VPUTPK      /SEND IT
    JMS @ VBMPSE      /BUMP SEQ.NO.
    JMP @ RINACK
/PICK UP FILENAME FROM FILE-HEADER PACKET
/CONVERT TO 6-BIT, DEPOSIT IN FILNAM, FILNAM+1
/IGNORE CHARS AFTER 6TH
MAKFN, 0
    ZERM EXTNUM       /CLEAR EXTENSION
    ZERM @ VASFLA     /AND THE .A FLAG

```

```

MEMA RIDATS      /SET PACKET POINTER
ACCM RIPKPT
MEMA @ RILEN     /AND NAME CHAR COUNTER
A-MA (243
ACCM RIPKCT
EXCT AC19 ZAC
JMP @ VNULNA
JMS FWORD        /FORM FIRST WORD
ACCM @ VFILNA    /DEPOSIT IT
JMS FWORD        /FORM SECOND WORD
ACCM @ VFILN1    /DEPOSIT IT
JMS EXTLUK       /CHECK FOR LATE EXTENSION
MEMA @ VFILN1    /SET EXTENSION, IF ANY
LLSH 2
A+MA EXTNUM
RLSH 2
ACCM @ VFILN1
JMP @ MAKFN
/FORM FILENAME WORD FROM 3 CHARS STARTING AT RIPKPT
/RETURN ZERO IF NAME ALREADY COMPLETED
FWORD, 0
ZERM FWHOLD
MEMA RIPKCT
EXCT AC19 ZAC
JMP FWNUL
JMS PIKCHA
JMS CONV86
LASH 14
ACCM FWHOLD
JMS FWBMP
JMS PIKCHA
JMS CONV86
LASH 6
A+MM FWHOLD
JMS FWBMP
JMS PIKCHA
JMS CONV86
A+MM FWHOLD
JMS FWBMP
MEMA FWHOLD
JMP @ FWORD
FWNUL, ZERA
JMP @ FWORD
FWHOLD, 0
/PICK UP A FILENAME CHARACTER FROM THE FILEHEADER PACKET
/IF A "." IS FOUND, BUMP POINTER AND EXAMINE NEXT CHARACTER
/IF IT IS AN A, B, OR C, SET EXTNUM AND EXIT FWORD WITH RIPKCT=0
PIKCHA, 0
MEMA @ RIPKPT    /COLLECT CHAR
A-MZ (".        /IS IT A "."?
JMP @ PIKCHA     /NO, EXIT
JMS FWBMP        /YES, BUMP POINTER
JMS EXTCHK       /SET EXTENSION IF A, B, OR C
MEMZ EXTNUM      /WAS IT?
ZERZ
JMP @ PIKCHA     /NO, EXIT WITH CHAR
MEMA FWHOLD      /YES, EXIT FWORD WITH FWHOLD
JMP @ FWORD
/CHECK WHETHER RIPKPT POINTS TO A VALID EXTENSION
/IF SO, SET EXTNUM=1,2,3 FOR A,B,C

```

```

/IF NOT, EXIT WITH CHAR IN ACC
/TO BE CALLED AFTER A "." HAS BEEN FOUND
EXTCHK, 0
    MEMA @ RIPKPT      /COLLECT CHAR
    A-MZ ("A           /IS IT A VALID EXTENSION?
    ZERZ
    JMP AEXT
    A-MZ ("B
    ZERZ
    JMP BEXT
    A-MZ ("C
    ZERZ
    JMP CEXT
    A-MZ ("a
    ZERZ
    JMP AEXT
    A-MZ ("b
    ZERZ
    JMP BEXT
    A-MZ ("c
    JMP @ EXTCHK      /NO, EXIT WITH CHAR
CEXT,  MEMA (3        /YES, SET .C
    ACCM EXTNUM
    JMP EXCOUT
AEXT,  ONEM EXTNUM    /SET .A
    ONEM @ VASFLA
    JMP EXCOUT
BEXT,  MEMA (2
    ACCM EXTNUM
EXCOUT, ZERM RIPKPT   /DON'T LOOK ANY FURTHER
    JMP @ EXTCHK
EXTNUM, 0             /0=NULL, 1=.A, 2=.B, 3=.C
/SEE IF THERE IS AN A, B, OR C EXTENSION BEYOND THE 6-CHAR LIMIT
/IF SO, RECORD IT AND EXIT
EXTLUK, 0
    MEMA RIPKCT      /ANY CHARS LEFT?
    EXCT AC19 ZAC
    JMP @ EXTLUK     /NO, JUMP OUT
EXTLUP, MEMA @ RIPKPT /YES, COLLECT ONE
    A-MZ ("."        /IS IT A "."?
    ZERZ
    JMP CHACHK       /YES
    MPOM RIPKPT      /NO BUMP POINTER
    MMOMZ RIPKCT     /ALL DONE?
    JMP EXTLUP       /NO, BACK FOR NEXT
    JMP @ EXTLUK     /YES, EXIT
CHACHK, MPOM RIPKPT  /BUMP POINTER
    MMOMZ RIPKCT     /ANY MORE CHARS?
    ZERZ
    JMP @ EXTLUK     /NO, EXIT
    JMS EXTCHK       /YES, SEEK EXTENSION
    JMP EXTLUP       /ANS SEE IF THERE ARE ANY MORE
/BUMP PACKET POINTER, DECREMENT COUNTER
/JUMP OUT OF FWORD WHEN DONE
FWBMP, 0
    MPOM RIPKPT
    MMOMZ RIPKCT
    JMP @ FWBMP
    MEMA FWHOLD
    JMP @ FWORD

```

```

/CONVERT 8-BIT ASCII TO 6-BIT ASCII
/INPUT AND OUTPUT CHARS BOTH IN ACC
/CONVERT ANY LOWER-CASE TO UPPER-CASE
/CONVERT ANY NON-TEXTUAL, NON-NUMERIC, CHARS TO ZZZ...
CONV86,0
    ACCM HOLD8
    A-MA (260          /TEXT OR NUMERICAL?
    EXCT AC19
    JMP CONVZ          /NEITHER, SO CONVERT TO Z
    MEMA HOLD8          /COULD BE
    A-MA (271
    EXCT AC19 ZAC
    JMP NUMUC          /YES, NUMERICAL
    MEMA HOLD8          /IS IT TEXTUAL?
    A-MA (301
    EXCT AC19
    JMP CONVZ          /NO, ITS PUNCTUAL
    MEMA HOLD8
    A-MA (332
    EXCT AC19 ZAC
    JMP NUMUC          /YES, UPPER-CASE
    MEMA HOLD8          /IS IT LOWER-CASE?
    A-MA (341
    EXCT AC19
    JMP CONVZ          /NO, ITS PUNCTUAL
    MEMA HOLD8
    A-MA (372
    EXCT AC19 ZAC
    JMP LCTXT          /YES, IT'S LOWER-CASE
CONVZ, MEMA ("Z-240    /CONVERT TO Z
    JMP @ CONV86
NUMUC, MEMA HOLD8      /CONVERT TO NUMERICAL UR UPPER-CASE
    A-MA (240
    JMP @ CONV86
LCTXT, MEMA HOLD8      /CONVERT LC TO 6-BIT UC
    A-MA (300
    JMP @ CONV86
HOLD8, 0                /TO HOLD 8-BIT CHAR
/SAY WHAT IS HAPPENING, PRINT INCOMING FILENAME
FILSAY,0
    MEMZ @ VDIAFL
    JMS @ VCRLF
    JMS @ VUNPCK        /PRINT "RECEIVING"
    RECNAM
    JMS @ VFPC
    FILNAM
    MEMZ EXTNUM          /IS THERE AN EXTENSION?
    JMP PEXTN            /YES
FLSOUT, JMP @ FILSAY    /NO, EXIT
PEXTN, MEMA ("
    JMS @ VTYPE
    MEMA EXTNUM
    A+MA (300
    JMS @ VTYPE
    JMP FLSOUT
RECNAM,TEXT %RECEIVING %
/POINTERS TO PAGE 110000
VGXZBA,GXZBAK
VCRLF, CRLF
VUNPCK,UNPCK

```

```

VFPCK, FPCK
VTYPE, TYPE
VGDL00, GDLOOP
VGDEND, GDEND
VCOMBA, COMBAK
/PAGE 112000
VDIAFL, DIAFLG
VRQCTL, RQCTL
VRETFL, RETFLG
VGETPK, GETPKT
VINPUT, INPUT
VABORT, ABORT
VCHKCH, CHKCHK
VSIDCO, SIDCON
VNAKPK, NAKPKT
VINISN, INISND
VCOMPCK, COMPKT
VPUTPK, PUTPKT
VACKPK, ACKPKT
VBMPSE, BMPSEQ
VSEQNO, SEQNO
VDECSE, DECSEQ
/PAGE 114000
VSIZE, SIZE
VASFLA, ASFLAG
VJOKER, JOKER
VFILNA, FILNAM
VFILN1, FILNAM+1
/PAGE 116000
VARWDS, ARWDS
VWRDTO, WRDTOT
VRECSI, RECSIZ
VRITWR, RITWRD
VWRDST, WRDSTO
VWRITE, WRITE
VSAVE, SAVE
VNULNA, NULNAM
PAGSKP
*112000
/PAGE 112000 HOLDS THE MAIN WORK ROUTINES FOR DATA-TRANSMISSION
/USES X-PREFIX POINTERS
/
SEQNO, 0 /PACKET SEQUENCE NUMBER (ASCII CHAR)
/
/POINTERS, COUNTERS, CONSTANTS FOR PACKET CONSTRUCTION IN OUTPUT BUFFER
OPAKST=102000
OMARK, OPAKST /FOR ^A MARK
OLEN, OPAKST+1 /LENGTH
OSEQ, OPAKST+2 /SEQUENCE NUMBER
OTYPE, OPAKST+3 /PACKET TYPE
ODATST, OPAKST+4 /START OF DATA
OPAKPT, 0 /OUTPUT PACKET POINTER
OPAKCT, 0 /OUTPUT PACKET COUNTER
/POINTERS, COUNTERS, CONSTANTS FOR PACKET DECONSTRUCTION
/IN INPUT BUFFER
IPAKST=103000
IMARK, IPAKST /FOR ^A MARK
ILEN, IPAKST+1 /LENGTH
ISEQ, IPAKST+2 /SEQUENCE NUMBER
ITYPE, IPAKST+3 /PACKET TYPE

```

```

IDATST,IPAKST+4          /START OF DATA
IPAKPT,0                 /INPUT PACKET COUNTER
IPAKCT,0                 /INPUT PACKET COUNTER
/LOCAL PARAMETER VALUES, TO BE SENT
/DEFAULTS THAT MAY BE ADJUSTED AS NECESSARY, IN REAL (UNCODED) UNITS
LMAXL, 136               /MAX PKT LENGTH WE CAN RECEIVE (94)
LTIME, 5                 /PERIOD (S) FOR US TO BE TIMED OUT
LNPAD, 0                 /NO. OF PADS WE WANT
LPADC, 200               /PAD CHAR WE WANT (=NULL)
LEOL, 215                /PKT TERMINATOR WE WANT (CR)
LQCTL, "#                /CONTROL PREFIX WE SEND
/REMOTE PARAMETER VALUES
/TO BE SET AT INITIATION, BUT DEFAULTS FOR NOW
/IN REAL (UNCODED) UNITS
RMAXL, 136               /MAX PKT LENGTH WE CAN SEND
RTIME, 5                 /TIME-OUT (S) WE ARE TO USE
RNPAD, 0                 /NO. OF PADS TO SEND
RPADC, 200               /PAD CHAR TO SEND
REOL, 215                /PKT TERMINATOR WE ARE TO SEND
RQCTL, "#                /CONTROL-PREFIX TO EXPECT
/TRANSMIT INITIATION PACKET, GET ACK
INISND,0
    MEMA (240             /INITIALISE PKT SEQ NO.
    ACCM SEQNO
    JMS CLPOPB            /CLEAR POP-BUFFER
    MEMA ("S              /S-TYPE PACKET
    ACCM @ OTYPE
    MEMA ODATST           /SET DATA POINTER
    ACCM OPAKPT
    MEMA LMAXL            /SET OUR MAXL
    A+MA (200
    JMS POPPUT
    MEMA LTIME            /SET OUR TIME-OUT
    A+MA (200
    JMS POPPUT
    MEMA LNPAD            /SE OUR PAD NO.
    A+MA (200
    JMS POPPUT
    MEMA LPADC            /SET OUR PAD CHAR
    JMS POPPUT
    MEMA LEOL             /SET OUR TERMINATOR
    JMS POPPUT
    MEMA LQCTL            /SET OUR CONTROL-PREFIX
    A-MA (40
    JMS POPPUT
    MPOM OPAKPT           /SPACE FOR CHECKSUM
    MEMA LEOL             /TERMIATOR
    ACCM @ OPAKPT
    JMS COMPKT            /COMPLETE THE PACKET
    ZERM RETFLG
    JMS PUTPKT            /YES, SEND THE PACKET
    JMS ACKGET            /AND GET AN ACK
    JMP @ INISND
/ADD CONTENTS OF ACC+40 AT OPAKPT, AND INCREMENT
POPPUT,0
    A+MA (40
    ACCM @ OPAKPT
    MPOM OPAKPT
    JMP @ POPPUT
/DECONSTRUCT/INTERPRET SEND-INITIATION ACK

```

```

SIDCON,0
    MEMA IDATST      /COLLECT PARAMETERS
    ACCM IPAKPT
    JMS NEXSID      /MAX PKT LENGTH TO SEND
    ACCM RMAXL
    JMS NEXSID      /TIMEOUT WE ARE TO USE
    ACCM RTIME
    JMS NEXSID      /NO. OF PADS TO SEND
    ACCM RNPAD
    JMS NEXSID      /PAD CHAR TO SEND
    A+MA (240
    ACCM RPADC
    JMS NEXSID      /PKT TERMINATOR TO SEND
    A+MA (200
    ACCM REOL
    JMS NEXSID      /CONTROL-PREFIX TO EXPECT
    A+MA (240
    ACCM RQCTL
    JMP @ SIDCON
/SUBROUTINE TO COLLECT SI DATA AND BUMP POINTER
NEXSID,0
    MEMA @ IPAKPT
    A-MA (240
    MPOM IPAKPT
    JMP @ NEXSID
/SEND PACKET WITH FILENAME, GET ACK
FILHED,0
    JMS CLPOPB      /CLEAR POP BUFFER
    MEMA ("F        /F-TYPE PACKET
    ACCM @ OTYPE
    MEMA ODATST     /SET DATA POINTER
    ACCM OPAKPT
    MEMA @ XFILN1   /INSERT FIRST 3 CHARS OF NAME
    JMS NAMPAK
    MEMA @ XFILN2   /INSERT SECOND THREE CHARS
    JMS NAMPAK
    MEMA @ XFILN2   /ANY EXTENSION?
    LLSH 2
    ANDA (3
    EXCT ZAC
    JMP NONEXT      /NO
    ACCM XTENSN     /YES, PRESERVE IT
    MEMA ("        /SEND THE "."
    ACCM @ OPAKPT
    MPOM OPAKPT
    MEMA XTENSN     /SEND THE A, B, OR C
    A+MA (300
    ACCM @ OPAKPT
    MPOM OPAKPT
NONEXT,MPOM OPAKPT /SPACE FOR CHECKSUM
    MEMA REOL      /TERMINATOR
    ACCM @ OPAKPT
    JMS BMPSEQ     /BUMP SEQ.NO.
    JMS COMPKT     /COMPLETE THE PKT
    JMS PUTPKT     /SEND THE PACKET
    ZERM FDPFLG    /CLEAR FIRST-DATA-PKT FLAG
    JMS ACKGET     /GET AN ACK
    JMP @ FILHED
XTENSN,0
/UNPACK 6-BIT FILENAME WORD IN ACC INTO POP-BUFFER

```

```

NAMPAK, 0
    ACCM NHOLD
    RASH 14
    JMS LETPUT
    RASH 6
    JMS LETPUT
    JMS LETPUT
    JMP @ NAMPAK
NHOLD, 0
/CONVERT LOWEST 6 BITS OF ACC TO 8-BIT ASCII CHAR
/AND PLACE IT IN POP-BUFFER
LETPUT, 0
    ANDA (77
    EXCT ZAC          /ASSUME NULL IMPLIES END
    JMP @ NAMPAK
    A-MZ (77         /AND 77 DITTO
    ZERZ
    JMP @ NAMPAK
    A+MA (240        /OTHERWISE, CONVERY
    ACCM @ OPAKPT    /DEPOSIT
    MPOM OPAKPT      /BUMP POINTER
    MEMA NHOLD       /RECOVER WORD
    JMP @ LETPUT     /AND EXIT
/SEND OFF THE WORDS LOADED AT 0 AS DATA PACKETS
/IF ASFLAG=-1, UNPACK AS PAIRS OF PACASC
/IF ASFLAG=0, TREAT AS INTEGER DATA
PKTOUT, 0
    ZERM PKOPNT      /SET POINTER
    MEMA @ XREDWR    /SET COUNTER
    ACCM PKOCNT
    MEMA (10         /INITIALISE COMRET
    ACCM NUMCNT
DPLUP, JMS DATPKT    /SEND DATA PACKET
    MEMZ @ XCXFLG    /-KB INTERRUPTION?
    JMP CXOUT        /YES
    MEMZ @ XCZFLG
    JMP CZOUT        /YES
    JMS ACHAR        /EXTERNAL ABORTION?
    A-MZ ("X
    ZERZ
    JMP CXOUT        /YES
    A-MZ ("Z
    ZERZ
    JMP CZOUT        /YES
    MEMA PKOCNT      /ANY MORE WORDS?
    SKIP AC19 ZAC
    JMP DPLUP        /YES, SEND NEXT PACKET
    JMP @ PKTOUT     /NO, ALL SENT
CXOUT, JMS DISCRD    /DISCARD CURRENT FILE
    ZERM @ XCXFLG    /CLEAR FLAG
    JMS ACKGET       /GET ACK
    JMP @ XPXBAK     /SEE IF ANY MORE FILES TO GO
CZOUT, JMS DISCRD    /DISCARD CURRENT FILE
    ZERM @ XCZFLG    /CLEAR FLAG
    JMS ACKGET       /SEEK ACK
    JMP @ XMOQER     /ASK FOR NEW FILENAME
PKOPNT, 0           /DATA PICKUP POINTER
PKOCNT, 0           /DATA PICKUP COUNTER
/MAKE UP AND TRANSMIT A DATA PACKET FROM PKOPNT POINTER
/TREATING AS PACASC IF ASFLAG=-1, AND INTEGER DATA IF =0

```



```

/INCREMENT PKOPNT, DECREMENT PKOCNT AS NEEDED
/DO NOT EXCEED RMAXL IN TOTAL LENGTH, AND DO NOT OVERRUN PKOCNT=0
DATPKT,0
    JMS CLPOPB          /CLEAR POP BUFFER
    MEMA ("D
    ACCM @ OTYPE
    MEMA ODATST        /SET POP-BUFFER POINTER
    ACCM OPAKPT
    MEMA RMAXL         /SET PKT LENGTH
    A-MA (20           /TO ALLOW FOR COMPLETIONS
    ACCM OPAKCT
    MMOZ FDPFLG        /FIRST DATA PACKET?
    JMS FDATA          /YES, SEND FILE DIR DATA AT FRONT
    ONEM FDPFLG
DMLUP, JMS WRDPUT       /MOVE A WORD (OR 2 IF ASFLAG=-1)
    MEMA OPAKCT        /AT END OF PKT?
    EXCT AC19 ZAC
    JMP DMFIN          /YES, FINISH OFF
    MEMA PKOCNT        /NO; AT END OF DATA?
    EXCT AC19 ZAC
    JMP DMFIN          /YES, FINISH OFF
    JMP DMLUP          /NO, MOVE ANOTHER WORD (OR 2)
DMFIN, MPOM OPAKPT     /SPACE FOR CHECKSUM
    MEMA REOL          /ADD TERMINATOR
    ACCM @ OPAKPT
    JMS BMPSEQ         /BUMP SEQ.NO.
    JMS COMPKT         /COMPLETE THE PKT
    JMS PUTPKT         /SEND THE PACKET
    JMS ACKGET         /GET AN ACK
    JMP @ DATPKT
FDPFLG,0              /0=FIRST DATA PACKET
/IF ASFLAG=0, MOVE ONE INTEGER INTO POP-BUFFER AS A DECIMAL NUMBER
/FOLLOWED BY A COMMA, WITH A CRLF AFTER EVERY 8TH ENTRY
/BUT IF ASFLAG=-1, UNPACK TWO ADDRESSES AS FIVE ASCII CHARACTERS
WRDPUT,0
    MPOZ @ XASFLA     /ASCII OR INTEGER?
    JMP WPINTG         /INTEGER
    JMS DILATE         /SEND AS ASCII
    JMP @ WRDPUT
WPINTG,JMS DECOUT      /SEND AS NUMERICAL
    JMS COMRET
    JMP @ WRDPUT
/EXPAND NEXT PAIR OF WORDS IN DISK BUFFER
/PLACE THE 5 CHARS IN PACKET OP-BUFFER
DILATE,0
    MEMA @ PKOPNT
    LLSH 10
    ANDA (377
    JMS ACCPOP         /CHAR-1
    MEMA @ PKOPNT
    RLSH 4
    ANDA (377
    JMS ACCPOP         /CHAR-2
    MEMA @ PKOPNT
    ANDA (17
    LASH 4
    ACCM DILHOL
    MPOM PKOPNT        /LOOK AT SECOND WORD
    MMOM PKOCNT
    MPOM @ XARWDS

```

```

MEMA @ PKOPNT
LLSH 4
ANDA (17
A+MA DILHOL
JMS ACCPOP          /CHAR3
MEMA @ PKOPNT
RLSH 10
ANDA (377
JMS ACCPOP          /CHAR-4
MEMA @ PKOPNT
ANDA (377
JMS ACCPOP          /CHAR-5
MPOM PKOPNT
MMOM PKOCNT
MPOM @ XARWDS
JMP @ DILATE
DILHOL,0
/MOVE ACC INTO POP-BUFFER, BUMP POINTER, DECREMENT COUNTER
/ENCODING IT IF NECESSARY, BUT IGNORING ZEROES AND FORMFEEDS
ACCPop,0
    ACCM APHOLD
    EXCT ZAC          /IGNORE ZERO
    JMP @ ACCPOP
    A-MZ (214         /FF IMPLIES END OF RECORD
    ZERZ
    JMP ENDREC
    A-MA (240         /ENCODING NEEDED?
    EXCT AC19
    JMP CODCHA        /YES
    MEMA APHOLD        /NO, JUST TRANSFER
    A-MZ LQCTL        /UNLESS IT IS THE CTRL/PREFIX?
    ZERZ
    JMP PREPRE        /YES IT IS
    ACCM @ OPAKPT
    APOUT, MPOM OPAKPT
    MMOM OPAKCT
    JMP @ ACCPOP
    PREPRE, ACCM @ OPAKPT /PREFIX THE PREFIX
    MPOM OPAKPT
    MMOM OPAKCT
    ACCM @ OPAKPT
    JMP APOUT
    CODCHA, MEMA LQCTL /SEND CONTROL-PREFIX
    ACCM @ OPAKPT
    MPOM OPAKPT
    MMOM OPAKCT
    MEMA APHOLD        /SEND UN-CTRL-ED CHAR
    A+MA (100
    ACCM @ OPAKPT
    A-MZ ("M          /WAS IT A CR?
    JMP APOUT          /NO
    MPOM OPAKPT        /YES, SEND AN LF+100
    MMOM OPAKCT
    MEMA LQCTL        /WITH PREFIX
    ACCM @ OPAKPT
    MPOM OPAKPT
    MMOM OPAKCT
    MEMA ("J
    ACCM @ OPAKPT
    MMOM OPAKCT

```

```

        JMP APOUT
ENDREC,ZERM PKOCNT      /END OF RECORD
        JMP @ ACCPOP
APHOLD,0
/TRANSFER CHAR TO PACKET IP-BUFFER
/BUT DO NOT OVERRUN LIMITS
/IF ANOTHER ^A ARRIVES, ASSUME THAT IT STARTS A NEW PACKET
/AND RESET BUFFER POINTER AND COUNTER
AIPBUF,0
        ACCM APBHOL
        MMOMA IPAKCT
        EXCT AC19 ZAC
        JMP @ AIPBUF
        MEMA APBHOL
        A-MA ("A-100
        EXCT ZAC
        JMS SETIPK
        MEMA APBHOL
        ACCM @ IPAKPT
        MPOM IPAKPT
        JMP @ AIPBUF
APBHOL,0
/SEND A Z-PACKET TO CLOSE THE FILE
ENDPKT,0
        JMS CLPOPB      /CLEAR POP-BUFFER
        MEMA ("Z        /Z-TYPE PACKET
        ACCM @ OTYPE
        MEMA ODATST
        APOM OPAKPT      /SPACE FOR CHECKSUM
        MEMA REOL       /TERMINATOR
        ACCM @ OPAKPT
        JMS BMPSEQ      /BUMP SEQ.NO.
        JMS COMPKT      /COMPLETE THE PACKET
        JMS PUTPKT      /SEND THE PACKET
        JMS ACKGET      /GET AN ACK
        JMP @ ENDPKT
/SEND Z-PACKET WITH D IN THE DATA FIELD, TO ABORT CURRENT TRANSMISSION
DISCRD,0
        JMS CLPOPB      /CLEAR POP-BUFFER
        MEMA ("Z        /Z-TYPE PACKET
        ACCM @ OTYPE
        MEMA ("D        /D FOR DISCARD
        ACCM @ ODATST
        MEMA ODATST
        A+MA (2         /SPACE FOR CHECKSUM
        ACCM OPAKPT
        MEMA REOL       /TERMINATOR
        ACCM @ OPAKPT
        JMS BMPSEQ      /BUMP SEQ.NO.
        JMS COMPKT      /COMPLETE THE PACKET
        JMS PUTPKT      /SEND THE PACKET
        JMP @ DISCRD
/SEND B-PACKET TO CLOSE TRANSMISSION
CLOSIT,0
        JMS PRGSET      /INITIALISE TYPROG
        JMS CLPOPB      /CLEAR POP-BUFFER
        MEMA ("B        /B-TYPE PACKET
        ACCM @ OTYPE
        MEMA ODATST
        APOM OPAKPT      /SPACE FOR CHECKSUM

```

```

MEMA REOL          /TERMINATOR
ACCM @ OPAKPT
JMS BMPSEQ         /BUMP SEQ.NO.
JMS COMPKT         /COMPLETE THE PACKET
JMS PUTPKT         /SEND THE PACKET
JMS ACKGET         /GET AN ACK
JMP @ CLOSIT
/COMPLETE THE PACKET IN THE POP-BUFFER
COMPKT,0
JMS MRKSEQ         /SET MARK AND SEQ. NO.
JMS LENCAL         /CALCULATE AND SET LENGTH
JMS CHKMAK         /CALCULATE AND SET CHECKSUM
ZERM RETFLG
JMP @ COMPKT
/SET MARK AND SEQUENCE NUMBER FOR OUTGOING PACKET
MRKSEQ,0
MEMA ("A-100      /MARK
ACCM @ OMARK
MEMA SEQNO         /SEQ. NO.
ACCM @ OSEQ
JMP @ MRKSEQ
/COMPUTE LENGTH OF PACKET AT OMARK, DEPOSIT RESULT AT OLEN
LENCAL,0
MEMA OMARK
ACCM LENCNT
ZERM LENCNT
LENLUP, MEMA @ LENCNT
MPOM LENCNT
MPOM LENCNT
A-MZ REOL
JMP LENLUP
MEMA LENCNT
A+MA (240-3
ACCM @ OLEN
JMP @ LENCAL
LENPNT,0
LENCNT,0
/PREPARE CHECKSUM FOR OUTGOING PACKET
/MUST BE *LAST* ACTION PRIOR TO TRANSMISSION
CHKMAK,0
MEMA OLEN          /SET POINTER, COUNTER
ACCM OPAKPT
MEMA @ OLEN
A-MA (240
ACCM OPAKCT
ZERA
OCHLUP, A+MA @ OPAKPT /SUMMATION LOOP
A-MA (200          /SUBTRACT THE NIC BIAS
MPOM OPAKPT
MMOMZ OPAKCT
JMP OCHLUP
ACCM ROCSUM
JMS CMAK           /CREATE THE CHECKSUM
ACCM OCSUM
ACCM @ OPAKPT      /DEPOSIT IT
JMP @ CHKMAK
ROCSUM,0           /RAW OP CHECKSUM
OCSUM,0            /KERMIT-ED VERSION
/LOOK FOR AN ACK
ACKGET,0

```

```

        JMS INPUT           /GET A PACKET
        A-MZ ("Y           /IS IT AN ACK?
        JMP BADAG          /NO
        JMP @ ACKGET       /YES, EXIT
BADAG,  MEMA ("Y           /ABORT WITH A Y
        JMS ABORT
/COLLECT DATA CHAR, IF ANY, FROM ACK
/EMERGE WITH RESULT IN ACC, OR ZAC IF NULL DATA FIELD
ACHAR, 0
        MEMA @ ILEN        /IS THERE ANY DATA?
        A-MZ (243
        JMP GOTCH         /YES
        ZERA              /NO, SET ZAC
        JMP @ ACHAR
GOTCH,  MEMA @ IDATST      /COLLECT DATA CHAR
        JMP @ ACHAR
/SEND ACK FOR CURRENT SEQUENCE NUMBER
ACKPKT,0
        JMS CLPOPB        /CLEAR POP-BUFFER
        MEMA ("Y          /Y-TYPE PACKET
        ACCM @ OTYPE
        MEMZ @ XCXFLG     /X-ABORTION?
        JMP CXABRT        /YES
        MEMZ @ XCZFLG     /Z-ABORTION?
        JMP CZABRT        /YES
        MEMA ODATST
        APOM OPAKPT
        MEMA REOL         /TERMINATOR
        ACCM @ OPAKPT
        JMS COMPKT        /COMPLETE THE PACKET
        ZERM RETFLG
        JMS PUTPKT        /SEND IT
        JMS BMPSEQ        /BUMP SEQ.NO.
        JMP @ ACKPKT
CXABRT, MEMA ("X          /ACK WITH AN X
        JMP DOXZ
CZABRT, MEMA ("Z          /ACK WITH A Z
DOXZ,   ACCM @ ODATST
        MEMA ODATST
        A+MA (2           /SPACE FOR CHECKSUM
        ACCM OPAKPT
        MEMA REOL         /TERMINATOR
        ACCM @ OPAKPT
        JMS COMPKT        /COMPLETE THE PACKET
        ZERM RETFLG
        JMS PUTPKT        /SEND IT
        JMS BMPSEQ        /BUMP SEQ.NO.
        ZERM @ XCXFLG     /CLEAR FLAGS
        ZERM @ XCZFLG
        JMP @ ACKPKT
/SEND NAK FOR CURRENT SEQUENCE NUMBER
NAKPKT,0
        JMS CLPOPB        /CLEAR POP-BUFFER
        MEMA ("N          /N-TYPE PACKET
        ACCM @ OTYPE
        MEMA ODATST
        ACCM OPAKPT
        MEMA REOL         /TERMINATOR
        ACCM @ OPAKPT
        JMS COMPKT        /COMPLETE THE PACKET

```

```

    ZERM RETFLG
    JMS PUTPKT      /SEND IT
    JMP @ NAKPKT
/SEND A PACKET
/IT MUST BE PRE-PREPARED AT OMARK ET SEQ
PUTPKT,0
    JMS SETOPK      /SET POINTER, COUNTER
    JMS PADPUT      /SEND PAD CHARS?
PPLUP, MEMA @ OPAKPT /TRANSFER CHAR TO OP-BUFFER
    JMS @ XACOBUS
    MPOM OPAKPT     /BUMP POINTER
    A-MZ REOL      /GOT TERMINATOR?
    ZERZ
    JMP PPFIN       /YES, ALL GONE
    MMOMZ OPAKCT    /NO, BUT CHECK COUNTER
    JMP PPLUP       /AND BACK FOR NEXT
PPFIN, JMS @ XRS DUT /EMPTY THE OP BUFFER
    JMS @ XARROW    /DISPLAYING PROGRESS ARROW
    MEMZ @ XOCHAC
    JMP PPFIN
    MEMA ("s        /PRINT "s" FOR PACKET SENT
    MEMZ RETFLG
    MEMA ("S        /PRINT "S" FOR RETRANSMITTED PACKET
    JMS TYPROG
    ONEM RETFLG
    JMP @ PUTPKT
RETFLG,0           /1=RETRANSMISSION
/SEND PAD CHARACTERS, AS AGREED
PADPUT,0
    MEMA RNPAD      /ANY TO SEND?
    EXCT AC19 ZAC
    JMP @ PADPUT     /NO, JUMP OUT
    ACCM PADCNT      /YES, SET COUNTER
PDPLUP, MEMA RPADC  /TRANSFER PADS TO OP-BUFFER
    JMS @ XACOBUS
    MMOMZ PADCNT
    JMP PDPLUP
    JMP @ PADPUT
PADCNT,0
/UNIVERSAL PACKET INPUT ROUTINE
/IF ALL IS WELL, EXITS WITH TYPE IN ACC
/BUT ABORTS AFTER 10 TIMEOUTS AND RETURNS TO COMMAND BLOCK
/TYPES AN "r" IF ALL IS OK
/OTHER DIAGNOSTIC INDICATORS ARE TYPED BY SUBROUTINES
INPUT, 0
    MEMA (12        /SET RETRY COUNTER
    ACCM TINCNT
REINP, JMS GETPKT   /CATCH A PACKET
    EXCT ZAC        /TIMED OUT?
    JMP BADINP      /YES
    JMS CHKLEN      /CHECK LENGTH
    JMP BADINP      /WRONG
    JMS CHKCHK      /CHECK CHECKSUM
    JMP BADINP      /WRONG
    JMS SEQCHK      /CHECK SEQ.NO.
    JMP SCASE       /WRONG, BUT IS IT SPECIAL CASE?
    MEMA @ ITYPE    /CHECK TYPE
    A-MZ ("N        /IS IT A NAK?
    JMP GUDINP      /NO, A GOOD PACKET
    MEMA ("n        /YES, TYPE n

```

```

JMS TYPROG
JMP BADINP
SCASE, JMS SEQNAK      /CHECK THE SPECIAL CASE
JMP BADINP            /IT ISN'T
MEMA ("Y              /IT IS, SO TREAT AS AN ACK
JMP @ INPUT
BADINP,MMOMZ TINCNT    /STILL RETRYING?
JMP RESEND            /YES
MEMA ("T              /NO, FED UP, ABORT SESSION
JMS ABORT
RESEND,JMS PUTPKT     /RESEND LAST PACKET
JMP REINP
GUDINP,MEMA ("r        /PRINT r TO INDICATE GOOD INPUT
JMS TYPROG
MEMA @ ITYPE          /EXIT WITH TYPE IN ACC
JMP @ INPUT
TINCNT,0
/GET A PACKET FROM RS232-B CHANNEL, TRANSFER TO PACKET IP-BUFFER
/WATCH OUT FOR ^A, THEN COLLECT RESULT AT 103000
/IF NOTHING, TIME-OUT AFTER RTIME SECONDS WITH ACC=0
/NORMALLY, EXIT WITH ACC=1
GETPKT,0
JMS CLPIPB            /CLEAR INPUT BUFFER
MEMA RSANUM           /SET TIME-OUT COUNT
ACCM GPCNT
JMS SETIPK            /INITIALISE I-PKT VARIABLES
ZERM @ XIGBFL         /PREPARE TO WATCH B-INPUT
ZERM @ XRSBCH         /CLEAR CHAR BUFF
GPLUP1,JMS @ XRSDUT    /WATCH LOOP
MEMA @ XRSBCH
A-MZ ("A-100          /GOT A MARK YET?
ZERZ
JMP GOTPKT            /YES, COLLECT THE PACKET
MMOMZ GPCNT           /TIME-OUT YET?
JMP GPLUP1            /NO
ONEM @ XIGBFL         /YES, STOP WATCHING B-INPUT
MEMA ("t              /INDICATE TIMEOUT WITH A t
JMS TYPROG
ZERA                  /AND EXIT WITH ZAC
JMP @ GETPKT
GOTPKT,ONEM @ XIPKTF   /SET INCOMING PACKET FLAG
MEMA ("A-100
JMS AIPBUF
GPLUP2,JMS @ XRSDUT    /PACKET BUILDING LOOP
MEMA @ XRSBCH
A-MZ LEOL            /WATCH FOR TERMINATOR
ZERZ
JMP GPKFIN            /GOT IT
JMS GPDLAY            /TARRY A MOMENT
MMOMZ GPCNT
JMP GPLUP2            /NOT TIMED-OUT YET
ZERM @ XIPKTF         /TIMED OUT IN MID-PKT, CLEAR PKT FLG
ONEM @ XIGBFL         /STOP WATCHING B-INPUT
MEMA ("T              /INDICATE TIMEOUT WITH A T
JMS TYPROG
ZERA                  /AND EXIT WITH ZAC
JMP @ GETPKT
GPKFIN,ZERM @ XIPKTF   /PACKET IS COMPLETE
ONEAM @ XIGBFL        /STOP WATCHING B-INPUT, EXIT ACO
JMP @ GETPKT

```

```

RSANUM,200000          /DEFAULT TIME-OUT COUNT
GPCNT, 0
/VARIABLE DELAY
GPDLAY,0
    MEMA GPDNUM
    AMOAZ
    JMP #-1
    JMP @ GPDLAY
GPDNUM,1
/CHECK SEQUENCE NUMBER OF INPUT PACKET
/FIRST RA IF WRONG, SECOND RA IF OK
SEQCHK,0
    MEMA SEQNO
    A-MZ @ ISEQ
    JMP BADSEQ          /WRONG
    MPOM SEQCHK          /OK, 2ND RA
    JMP @ SEQCHK
BADSEQ,MEMA ("w          /PRINT "w" FOR WRONG SEQ.NO.
    JMS TYPROG
    JMP @ SEQCHK          /AND 1ST RA
/CHECK FOR SPECIAL CASE OF NAK TO *NEXT* PACKET
/FIRST RA IF NOT, SECOND RA IF IT IS THE SPECIAL CASE
SEQNAK,0
    MEMA @ ITYPE          /IS IT A NAK?
    A-MZ ("N
    JMP @ SEQNAK          /NO
    MEMA @ ISEQ          /IS (@ISEQ - SEQNO)=1 ?
    A-MA SEQNO
    AMOZ
    JMP ORDNAK          /NO, ORDINARY NAK
    MEMA ("N          /YES, THIS IS IT, PRINT "N"
    JMS TYPROG
    MPOM SEQNAK          /TAKE 2ND RA
    JMP @ SEQNAK
ORDNAK,MEMA ("n          /PRINT "n" FOR NAK
    JMS TYPROG
    JMP @ SEQNAK          /AND TAKE 1ST RA
/CHECK THAT INPUT PACKET LENGTH IS CORRECT
/FIRST RA IF WRONG, SECOND RA IF OK
CHKLEN,0
    JMS SETIPK
CKLOOP,MPOM IPAKPT          /CALCULATE LENGTH
    MEMA @ IPAKPT
    A-MZ REOL          /GOT TERMINATOR?
    JMP CKLOOP          /NO
    MEMA IPAKPT          /YES, DO THE COMPARISON
    A-MA ISEQ
    A+MA (240
    A-MZ @ ILEN
    JMP BADLEN          /DISAGREES
    MPOM CHKLEN          /OK, TAKE 2ND RA
    JMP @ CHKLEN
BADLEN,MEMA ("l          /PRINT "l" FOR WRONG LENGTH
    JMS TYPROG
    JMP @ CHKLEN          /TAKE 1ST RA
/UNIVERSAL ABORTION ROUTINE
/CODED AS A SUBROUTINE TO FACILITATE DEBUGGING
/IF A WRONG INPUT PACKET WAS FOUND, ENTER WITH CORRECT TYPE IN ACC
/IF TEN FAILURES, ENTER WITH T IN ACC
ABORT, 0

```



```

ZERM @ XRTFLG      /ABORT THE SESSION
ONEM @ XIGBFL      /STOP WATCHING B-INPUT
ACCM HABORT
JMS @ XCRLF
JMS @ XCRLF
JMS @ XUNPCK
ABSING
JMS @ XCRLF
MEMA HABORT
A-MZ ("T           /NO-VALID-PACKET?
ZERZ
JMP NVP            /YES
JMS @ XUNPCK       /MUST BE WRONG TYPE OF PACKET
RECP
MEMA @ ITYPE       /SAY WHAT WE GOT
JMS @ XTYPE
JMS @ XUNPCK
DASHP1
MEMA HABORT        /SAY WHAT WE WANTED
JMS @ XTYPE
JMS @ XUNPCK
DASHP2
JMP ABOUT          /AND EXIT
NVP, JMS @ XUNPCK   /ANNOUNCE 10 TIMEOUTS
TENTS
ABOUT, JMS @ XCRLF
JMP @ XCOMBA       /BACK TO COMMAND BLOCK
HABORT,0
ABSING,TEXT %ABORTING SESSION -%
RECP, TEXT %RECEIVED %
DASHP1,TEXT %-PACKET WHEN EXPECTING %
DASHP2,TEXT %-PACKET%
TENTS, TEXT %TEN TIME-OUTS WITH NO VALID INPUT PACKET RECEIVED%
/BUMP PACKET SEQUENCE NUMBER
BMPSEQ,0
MPOMA SEQNO
A-MZ (340
JMP @ BMPSEQ
MEMA (240
ACCM SEQNO
JMP @ BMPSEQ
/DECREMENT PACKET SEQUENCE NUMBER
DECSEQ,0
MMOMA SEQNO
A-MZ (237
JMP @ DECSEQ
MEMA (337
ACCM SEQNO
JMP @ BMPSEQ
/CLEAR PACKET IP-BUFFER
CLPIPB,0
MEMA IMARK
JMS CLPBUF
JMP @ CLPIPB
/CLEAR PACKET OP-BUFFER
CLPOPB,0
MEMA OMARK
JMS CLPBUF
JMP @ CLPOPB
/CLEAR FIRST 200 ADDRESSES OF PACKET IP/OP BUFFER

```

```

/START TO BE IN ACC ON ENTRY
/BUT IF DIAFLG=1, PUSH OLD PKTS UP IN THE BED AND RETAIN THEM
CLPBUF,0
    MMOZ DIAFLG
    JMP ORDCLP
    JMS DIACLR
    JMP @ CLPBUF
ORDCLP,ACCM CLRBPT
    MEMA (200
    ACCM CLRBCT
CLBLUP,ZERM @ CLRBPT
    MPOM CLRBPT
    MMOMZ CLRBCT
    JMP CLBLUP
    JMP @ CLPBUF
CLRBPT,0
CLRBCT,0
/DIAGNOSTIC BUFFER CLEARANCE
/MOVES EVERYTHING UP IN THE BED BY 200
/FIRST BLOCK ADD TO BE IN ACC ON ENTRY
/BUT IF FIRST ADDRESS =0, JUST JUMPS OUT - BECAUSE CLEARANCE UN-NEEDED
DIACLR,0
    ACCM BOTDC
    MEMZ @ BOTDC
    ZERZ
    JMP @ DIACLR
    MEMA BOTDC
    A+MA (777
    ACCM DCPT2
    A-MA (200
    ACCM DCPT1
    MEMA (600
    ACCM DCCNT
DCLOOP,MEMA @ DCPT1
    ZERM @ DCPT1
    ACCM @ DCPT2
    MMOM DCPT1
    MMOM DCPT2
    MMOMZ DCCNT
    JMP DCLOOP
    JMP @ DIACLR
BOTDC, 0
DCPT1, 0
DCPT2, 0
DCCNT, 0
DIAFLG,0
/TOGGLE PACKET-ARCHIVING ON/OFF
ARCTOG,0
    ONEM @ XCMDFL
    JMS @ XCRLF
    JMS @ XUNPCK
    INDARC
    MEMZ DIAFLG
    JMP ARCOFF
    ONEM DIAFLG
    JMS @ XUNPCK
    AON
    JMP ARCOUT
ARCOFF,ZERM DIAFLG
    JMS @ XUNPCK

```

```

        AOFF
ARCOUT,JMS @ XCRLF
        JMP @ ARCTOG
INDARC,TEXT %INDICATORS & ARCHIVING %
AOFF, TEXT %OFF%
AON, TEXT %ON%
/CLEAR BOTH PACKET BUFFERS FULLY
RTCLR, 0
        MEMA OMARK
        ACCM RTCPNT
        MPOA (1777
        ACCM RTCCNT
RTCLUP,ZERM @ RTCPNT
        MPOM RTCPNT
        MMOMZ RTCCNT
        JMP RTCLUP
        JMP @ RTCLR
RTCPNT,0
RTCCNT,0
/CHECK CHECKSUM FOR PACKET IN INPUT BUFFER
/IF ERROR, 1ST RA
/NORMALLY, IF OK, 2ND RA
CHKCHK,0
        MEMA ILEN          /SET POINTER, COUNTER
        ACCM IPAKPT
        MEMA @ ILEN
        A-MA (240
        EXCT AC19 ZAC
        JMP BADCHK
        ACCM IPAKCT
        ZERA
ICHLUP,A+MA @ IPAKPT      /SUMMATION LOOP
        A-MA (200          /SUBTRACT NIC BIAS
        MPOM IPAKPT
        MMOMZ IPAKCT
        JMP ICHLUP
        ACCM RICSUM
        JMS CMAK          /CREATE CHECKSUM
        ACCM ICSUM
        A-MA @ IPAKPT     /COMPARE RESULT IN PKT
        SKIP ZAC
        JMP BADCHK        /ERROR
        MPOM CHKCHK       /OK, IT MATCHES
        JMP @ CHKCHK
BADCHK,MEMA ("c          /TYPE "c" TO INDICATE ERROR
        JMS TYPROG
        JMP @ CHKCHK      /AND FIRST RA
RICSUM,0                /RAW IP CHECKSUM
ICSUM, 0                /KERMIT-ED VERSION
/CREATE 8-BIT KERMIT CHECKSUM
/RAW CHECKSUM TO BE IN ACC ON ENTRY, EXIT WITH RESULT IN ACC
CMAK, 0
        ACCM CMHOLD
        ANDA (300
        RASH 6
        A+MA CMHOLD
        ANDA (77
        A+MA (240
        JMP @ CMAK
CMHOLD,0

```

```

CMWORK,0
/SET POINTERS, COUNTER, ETC, FOR INCOMING PACKET
SETIPK,0
    MEMA IMARK        /SET POINTER
    ACCM IPAKPT
    MEMA LMAXL        /SET COUNTER, FOR SAFETY
    ACCM IPAKCT
    JMP @ SETIPK
/SET POINTERS, COUNTER, ETC, FOR OUTGOING PACKET
SETOPK,0
    MEMA OMARK        /SET POINTER
    ACCM OPAKPT
    MEMA RMAXL        /SET COUNTER, FOR SAFETY
    ACCM OPAKCT
    JMP @ SETOPK
/TYPE A CHARACTER TO INDICATE PROGRESS
/ACTIVE IN ARCHIVAL/DIAGNOSTIC MODE, WHEN DIAFLG=1
/CHARACTER TO BE IN ACC ON ENTRY
/COUNTER SHOULD BE SET PRIOR TO RECEIVE/TRANSMIT
TYPROG,0
    MMOZ DIAFLG        /UNLESS DIAFLG IS SET...
    JMP @ TYPROG        /...JUMP OUT
    JMS @ XACIBU        /RELY ON FUTURE RSDUTIES TO PRINT
    MMOMZ TYPRCT        /END OF LINE ON TERMINAL?
    JMP @ TYPROG        /NO
    JMS PRGSET          /YES, RESET COUNTER
    JMP @ TYPROG
TYPRCT,0
/INITIALISE TYPROG COUNTER
PRGSET,0
    MEMA (215          /POP A CRLF INTO THE PRINTER BUFFER
    JMS @ XACIBU
    MEMA (212
    JMS @ XACIBU
    MEMA TYPRNO        /SET THE COUNTER
    ACCM TYPRCT
    JMP @ PRGSET
TYPRNO,106
/COMPUTE NUMBER FOR TIME-OUT COUNTER
/LEAVE RESULT IN RSANUM FOR GETPKT TO USE
/IF ACC OVERFLOWS, JUST SET 2**19-1
RTCALC,0
    MEMA RTIME
    TACMQ
    MULT
    31463              /DECIMAL 13107
    SKIP ZAC
    JMP RSAOTP
    TMQAC
    EXCT AC19
    JMP RSAOTP
RTCOUT,ACCM RSANUM
    JMP @ RTCALC
RSAOTP,MEMA MAXRSA
    JMP RTCOUT
MAXRSA,1777777
/ENSURE HEADER IS NOT TRANSFERRED TO SYMMETRY
/IF NO HEADER, 2ND RA
/IF HEADER, BUMP POINTER & DECREMENT COUNTER BY 1K
/IF COUNTER THEN =0 (IE NIC-298), TAKE 1ST RA

```

```

/OTHERWISE, 2ND RA
TSTHED, 0
    MEMA @ IDTPNT
    A-MZ IDTRM
    JMP TSTOK
    MPOA (1777
    A+MM TRANPT
    M-AMZ TRANCT
    JMP TSTOK
    JMP @ TSTHED
TSTOK, MPOM TSTHED
    JMP @ TSTHED
IDTPNT, 2
IDTRM, 770320
TRANPT, 0
TRANCT, 0
/SEND FILE DIRECTORY DATA AT FRONT OF FIRST DATA PACKET
/SIZE, LOAD ADDRESS, STARTING ADDRESS
FDATA, 0
    MEMA XSIZE          /SET PICK-UP POINTER
    ACCM PKOPNT
    MEMA (3              /COUNTER
    ACCM FDCNT
    ACCM NUMCNT
FDLOOP, JMS DECOUT      /SEND LOOP
    JMS COMRET
    MMOMZ FDCNT
    JMP FDLOOP
    ZERM PKOPNT          /RESET POINTER, COUNTER
    MEMA @ XREDWR
    ACCM PKOCNT
    JMP @ FDATA
FDCNT, 0
/TYPE A COMMA, WITH A CR EVERY 8TH LINE
COMRET, 0
    MEMA ("
    JMS ACCPOP
    MMOMZ NUMCNT
    JMP @ COMRET
    MEMA (215
    JMS ACCPOP
    MEMA (10
    ACCM NUMCNT
    JMP @ COMRET
NUMCNT, 0
/MOVE INTEGER INTO POP-BUFFER AS A SIGNED DECIMAL NUMBER
/WITH SPECIAL TREATMENT FOR 2000000, WHICH SETS BIGFLG
DECOUT, 0
    ZERM BIGFLG
    MEMA @ PKOPNT
    MPOM PKOPNT
    MMOM PKOCNT
    MPOM @ XARWDS
    ACCM DECNUM
    EXCT ZAC
    JMP NULDEC
    ZERM SGNFLG
    SKIP AC19
    JMP POSDEC
    MNGAM DECNUM

```

```

        EXCT AC19
        ONEM BIGFLG
        MONM SGNFLG
POSDEC, MEMA (5
        ACCM DIGITS
        MEMA EQUJST
        ACCM EQUIPNT
        ZERM SPACFG
DECLUP, MEMA @ EQUIPNT
        ACCM DECDIV
        MEMA DECNUM
        LASH 1
        TACMQ
        ZERA
        MEMZ BIGFLG
        ONEA
        ZERM BIGFLG
        DIVD
DECDIV, 0
        RASH 1
        ACCM DECNUM
        TMQAC
        SKIP ZAC
        JMP NONORT
        MPOZ SPACFG
        JMP NOTSIG
NONORT, MPOZ SPACFG
        JMS SGNPRN
        MONM SPACFG
        A+MA (260
        JMS ACCPOP
NOTSIG, MPOM EQUIPNT
        MMOMZ DIGITS
        JMP DECLUP
        MPOZ SPACFG
        JMS SGNPRN
        MEMA DECNUM
        A+MA (260
        JMS ACCPOP
        JMP @ DECOUT
NULDEC, MEMA (260
        JMS ACCPOP
        JMP @ DECOUT
DECNUM, 0
SGNFLG, 0
SPACFG, 0
BIGFLG, 0
DIGITS, 0
EQUIPNT, 0
EQUJST, EQUJLS
EQUJLS, 303240
        23420
        1750
        144
        12
BIGNUM, 1000000
/PRINT SIGN, IF NEGATIVE
SGNPRN, 0
        MPOZ SGNFLG
        JMP @ SGNPRN

```

```

/NUMBER TO BE SENT
/SIGN FLAG
/LEADING ZERO SUPPRESS FLAG
/SET BY A 2000000
/COUNTER FOR DIGITS
/POINTER FOR DECIMAL EQUIVALENTS
/START OF DECIMAL EQUIVALENTS
/100000
/10000
/1000
/100
/10
/HALF OF MOD(2000000)

```

```

        ACCM SGNTEM
        MEMA (" -
        JMS ACCPOP
        MEMA SGNTEM
        JMP @ SGNPRN
SGNTEM, 0
/PAGE 110000 POINTERS
XCXFLG, CXFLG
XCZFLG, CZFLG
XPXBAK, PXBAK
XMOQER, MOQERY
XCMDFL, CMDFLG
XRTFLG, RTFLG
XIGBFL, IGBFLG
XTYPE, TYPE
XCRLF, CRLF
XUNPCK, UNPCK
XCOMBA, COMBAK
XOCHAC, OCHACT
XRS DUT, RSDUTY
XACOB U, ACOBU F
XACIB U, ACIBU F
XRSBCH, RSBCHA
XIPKTF, IPKTFG
/PAGE 114000 POINTERS
XSIZE, SIZE
XFILN1, FILNAM
XFILN2, FILNAM+1
XASFLA, ASFLAG
/PAGE 116000 POINTERS
XREDWR, REDWRD
XARROW, ARROW
XARWDS, ARWDS
XWRDST, WRDSTO
PAGSKP
*114000
/PAGE 114000 HOLDS THE FILENAME-PARSING ROUTINES
/THEN, SPACE FOR NICBUG AND BOOTS
/USES Y-PREFIX POINTERS
/ROUTINES FOR FILENAME SPECIFICATION
/
/GET A LINE FROM THE KEYBOARD, USING READ.  LINE EDITING
/IS AVAILABLE USING <RUBOUT>, <BS>, <CTRL-O>, & <LF>.  LINE IS TERM-
/INATED BY A <CR> OR THE 79'TH CHARACTER.  CHARACTER COUNT WILL
/BE IN LCC, AND LP WILL POINT TO START OF THE BUFFER.
GETLIN, 0
        JMS @ YCRLF
GTLIN0, JMS @ YCRLF
        MEMA (" @
        JMS @ YTYPE
        MEMA LINBUF
        ACCM LP /INITIALIZE LINE PNTR
        ZERM LCC /AND CHAR CNTR
        MEMA @ YRECSI /AND DIR COUNTER
        ACCM DIRSIZ
GTLIN1, JMS @ YREAD /READ AND ECHO CHAR
        A-MZ (215 /CR?
        JMP GTLIN2 /NO
/PROCESS CR
DOCR, MEMA (215 / (THIS IS FOR LINE OVFLW)

```

```

        JMS      RUBCHK
        JMS      @ YCRLF
        MEMA     LINBUF
        ACCM     LP          /RESTORE LP
        MPOM     LCC        /COUNT CR AS A CHAR
        JMP      @GETLIN /DONE
GTLIN2, A-MZ    (212      /LF?
        JMP      GTLIN3    /NO
/PROCESS LF
KBECHL, JMS     RUBCHK
        MEMA     LINBUF
        ACCM     LP
        MEMA     LCC
        ACCMZ    RUBFLG
        ZERZ
        JMP      GTLIN1    /NOTHING TO ECHO
        MEMA     (" ^
        JMS      @ YTYPE   /FIRST ECHO " ^"
        JMS      @ YCRLF   /  THEN @ YCRLF
KBECLP, MEMA    @LP        /  THEN THE LINE
        MPOM     LP
        JMS      @ YTYPE
        MMOMZ    RUBFLG
        JMP      KBECLP
        JMP      GTLIN1    /LINE ECHOED
GTLIN3, A-MZ    (210      /BS?
        ZERZ
        MEMA     (377      /YES, SUBSTITUTE RUBOUT
        A-MZ     (377      /RUBOUT?
        JMP      GTLIN4    /NO
/PROCESS RUBOUT
BKSPC, MMOMA   LCC
        EXCT     AC19
        JMP      GTLIN0    /NOTHING TO DELETE
        MMOM     LP
        MEMZ     RUBFLG    /RUBOUTS UNDERWAY?
        JMP      BKSP1     /YES
        MONM     RUBFLG    /NO, TURN FLAG ON
        MEMA     (" [
        JMS      @ YTYPE   /ECHO " [ "
BKSP1, MEMA    @LP
        JMS      KBECHO    /ECHO DELETED CHAR
        JMP      GTLIN1
GTLIN4, A-MZ    ("O-100   /CTRL-O?
        JMP      GTLIN5    /NO
/PROCESS CTRL-O
ERSLN, JMS     RUBCHK
        JMP      GTLIN0
GTLIN5, A-MZ    (200      /NULL?
        ZERZ
        JMP      GTLIN1    /IGNORE NULLS
        JMS      RUBCHK
        MPOMA    LCC        /INCREMENT COUNT
        A-MZ     (117      /COUNT=79?
        JMP      GTLIN1    /NO
        JMP      DOCR       /YES, FORCE END OF LINE

RUBCHK, 0
        ACCM     @LP
RUBCK1, MPOM    LP

```



```

MCPZ    RUBFLG  /RUBOUTS UNDERWAY?
JMP      @RUBCHK /NO
MEMA     ("]     /YES, FINISH BRACKETS
JMS      @ YTYPE
ZERM     RUBFLG
MMOM     LP
MEMA     @LP
JMP      RUBCK1

```

```

KBECHO, 0
A-MZ     (211    /HT?
JMP      #+3
MEMA     (240    /YES, ECHO 2 SPACES
JMS      @ YTYPE
JMS      @ YTYPE
JMP      @KBECHO

```

```

LINBUF, ZZZEND
RUBFLG, 0
LP,     0
LCC,    0

```

```

/GETDAB: PARSES A LINE OF THE FORM (/) (NAMGRP) (.X) (-D2) (:Z)
/ WHERE NAMGRP.X MAY CONTAIN *'S AND/OR ?'S
GETDAB, 0

```

```

ZERM JOKER
ZERM NFILES
MEMA ASFADD
ACCM    TEMP
MEMA    (12
ZERM    @TEMP
MPOM    TEMP
AMOAZ
JMP      #-3      /CLEAR THE DAB & MASKS
JMS      GETNBC
A-MZ     (240      /(IF SPACE, ACTUALLY CR)
ZERZ
JMP      #+3
A-MZ     ("/      /IGNORE LEADING '/'
MMOM     LP
JMS      GETNAM
MEMA     NAMWRD
ACCM     NAME1
MEMA     NAMASK
ACCM     NAMSK1
JMS      GETNAM
MEMA     NAMWRD
ACCM     NAME2
MEMA     NAMASK
LASH     2
A+MA     (3
RLSH     2
ACCM     NAMSK2
GTDAB1, MEMA    @LP
MPOM     LP
A-MZ     (".
ZERZ
JMP      GTDABE
A-MZ     (" -
ZERZ

```

```

JMP DABERR
  A-MZ      (":
  JMP       @GETDAB
  MEMA      @LP
  ACCM      OPTION
  JMP       @GETDAB

GTDABE, JMS   GETNBC   /GET EXTENSION
  A-MZ      ("P
  ZERZ
  JMP       GTDABE   /IGNORE 'P'
  A-MZ      ("*
  ZERZ
  JMP       GTDAB2   /'*' MEANS ANY EXTENSION
  A-MZ      ("?
  ZERZ
  JMP       GTDAB2   /  SO DOES '?'
  A-MA      ("@
  EXCT      ZAC AC19
  JMP       DABERR   /ILLEGAL CHAR
  A-MA      (4
  SKIP      AC19
  JMP       DABERR   /ILLEGAL EXTENSION
  A+MA      (4
  RLSH      2
  A+MM      NAME2
  JMP       GTDAB1   /EXTENSION IN BITS 18-19
GTDAB2, MEMA  NAMSK2
  LASH      2
  RISH      2
  ACCM      NAMSK2   /ALLOW ANY EXTENSION
ONEM JOKER
  JMP       GTDAB1

DABERR, JMS @ YCRLF           /ERROR IN DAB
  JMS @ YUNPCK
  ERR8
  JMS @ YEXPLA
  JMP @ YNEWNA

JOKER, 0                               /SET =1 BY ANY "?" OR "*" CHARACTERS

GETNAM, 0                               /FORM NEXT FILENAME WORD
  ZERM      NAMASK
  ZERM      NAMWRD
  MEMA      (3
  ACCM      LTRCNT
GTNAM1, JMS   GETNBC
  A-MZ      ("*                               /SORT OUT THE JOKERS
  ZERZ
  ONEM JOKER
  A-MZ      ("?
  ZERZ
  ONEM JOKER

  A-MZ      ("*
  ZERZ
  JMP       @GETNAM / * MEANS ANY 3 CHARS
  A-MZ      ("?
  ZERZ

```

```

        JMP      GTNAM2  /? MEANS ANY 1 CHAR
        A-MA      (240
        ACCM      TEMP
        MEMA      LTRCNT
        ACCM      TEMP2
        MEMA      TEMP
        LLSH      6
        MMOMZ     TEMP2
        JMP      #-2      /ROTATE TOO FAR
        RLSH      6        /POSITION CORRECTLY
        A+MM      NAMWRD   /BUILD NAME
        MEMA      LTRCNT
        ACCM      TEMP2
        MEMA      (77
        LLSH      6
        MMOMZ     TEMP2
        JMP      #-2
        RLSH      6
        A+MM      NAMASK   /BUILD NAME MASK
GTNAM2, MMOMZ     LTRCNT
        JMP      GTNAM1
        JMP      @GETNAM /EXIT

GETNBC, 0        /GET NEXT NON-BLANK CHAR
GTNBC1, MEMA     @LP
        MPOM      LP
        A-MZ      (240
        ZERZ
        JMP      GTNBC1  /IGNORE SPACE
        A-MZ      (211
        ZERZ
        JMP      GTNBC1  /IGNORE TAB
        A-MZ      (215
        ZERZ
        JMP      GTNBC2
        A-MA      (240
        EXCT      AC19
        JMP      DABERR   /CONTROL CHAR
        A-MA      ("[-240
        SKIP      AC19
        JMP      DABERR   /LOWER CASE OR SPECIAL
        A+MA      ("[
        A-MZ      (" -
        ZERZ
        JMP      GTNBC2  /'-' STARTS DEVICE CODE
        A-MZ      (" .
        ZERZ
        JMP      GTNBC2  /'.' STARTS EXTENSION
        A-MZ      (" :    /':' PRECEDES OPTION
        JMP      @GETNBC /NONE OF THE ABOVE
GTNBC2, MEMA     (240    /SUBSTITUTE A SPACE
        MMOM      LP      /BACK UP TO TERMINATOR
        JMP      @GETNBC

```

```

LTRCNT, 0
NAMWRD, 0
NAMASK, 0
ASFLAG, 0
NAME1, 0
NAME2, 0

```

```

FRSTRK, 0
SIZE, 0
MLA, 0
PSA, 0
NAMSK1, 0
NAMSK2, 0
OPTION, 0
NFILES, 0 /NUMBER OF FILES FOUND
TEMP, 0 /WORK ADDRESSES
TEMP2, 0
REMDER, 0
ASFADD, ASFLAG
/LOOKUP: SEARCHES THE DISK DIRECTORY STARTING AT THE NEXT
/POSITION (DIRPNT+5).
/1ST RA IF NO FILE FOUND, 2ND RA IF FILENAME FINDS A MATCH
LOOKUP, 0
LKP2, MEMA (5
      A+MM DIRPNT
      M-AMA DIRSIZ
      EXCT AC19
      JMP LKP5
      MCPZ @DIRPNT /END OF DIRECTORY?
      JMP LKP6 /NO
LKP5, MEMZ NFILES /ANY FILES FOUND?
      JMP @ LOOKUP /YES, SO EXIT
      JMS @ YUNPCK /NO, TELL HIM SO
      ERR1
      JMP @ LOOKUP /RESTART
/SEARCH THE DIRECTORY
LKP6, MEMAZ @DIRPNT /LOOK AT WORD 1
      JMP LKP7 /NOT DELETED
      JMP LKP2 /NO, SKIP IT
LKP7, MEMA DIRPNT /TRY TO MATCH THE NAME
      APOM DRPNT2
      MEMA NAME1
      ANDA NAMSK1
      ACCM TEMP
      MEMA @DIRPNT
      ANDA NAMSK1
      A-MZ TEMP
      JMP LKP2 /DIDN'T MATCH
      MEMA NAME2
      ANDA NAMSK2
      ACCM TEMP
      MEMA @DRPNT2
      ANDA NAMSK2
      A-MZ TEMP
      JMP LKP2 /DIDN'T MATCH
      MEMA @DRPNT2
      MPOM DRPNT2
      ACCM NAME2
      LLSH 2
      ANDA (3
      A-MZ (1
      ZERMZ ASFLAG /CLEAR IF .B, .C, OR NULL
      MONM ASFLAG /SET IF .A
      MEMA @DIRPNT
      ACCM NAME1
      MEMA @DRPNT2
      ANDA BITS16 /(177777)

```

```

    ACCM      MLA      /MLA(0:15)
    MCPA      BITS16
    ANDA      @DRPNT2
    RISH      1
    ACCM      PSA      /PSA(15:18)
    MPOM      DRPNT2
    MEMA      @DRPNT2 /GET WORD 4
    ANDA      BITS16
    ACCM      SIZE      /SIZE, MOD 2**16
    MCPA      BITS16
    ANDA      @DRPNT2
    RISH      5
    A+MM      PSA      /PSA(11:14)
    MPOM      DRPNT2
    MEMA      @DRPNT2
    RISH      11
    A+MM      PSA      /PSA(0:10)
    MEMA DIRPNT
    A+MA (4
    ACCM DRPNT2
    MEMA @ DRPNT2
    ANDA (777
    ACCM FRSTRK      /FIRST RECORD
    JMS CKBDIR      /IS IT THE BACKUP DIR?
    JMP @ LOOKUP      /YES, DON'T SEND IT
    JMS      PRFLNM /TYPE FOUND FILENAME
    MPOM LOOKUP
    JMP      @LOOKUP /YES, RETURN
/SAY FILENAME BEING SENT
PRFLNM,0
    MPOM NFILES      /BUMP FILE NO.
    MEMZ @ YDIAFL
    JMS @ YCRLF
    JMS @ YCRLF
    JMS @ YUNPCK      /PRINT "SENDING"
    SND
    MEMA NAME1      /PRINT FILENAME
    ANDA PRMSK
    ACCM FILNAM
    MEMA NAME2
    ANDA PRMSK
    ACCM FILNAM+1
    JMS @ YFPCK
    FILNAM
    MEMA NAME2      /PRINT EXTENSION
    LLSH 2
    ANDA (3
    ACCM EXTEN
    RLSH 2      /INSERT INTO FILENAME
    TACMQ
    MEMA FILNAM+1
    INCOR
    ACCM FILNAM+1
    MEMA NAME1      /PRINT "."?
    EXCT AC19
    JMP PNTPRN      /YES
    MEMZ EXTEN
    ZERZ
    JMP SPCOUT      /NO
PNTPRN, MEMA ("      /YES, PRINT THE "."

```

```

JMS @ YTYPE
MEMA EXTEN
  A+MA      ("@
  A-MZ      ("@
  JMS       @ YTYPE
  MEMA      NAME1
SKIP AC19
JMP SPCOUT
  MEMA      ("P      / TYPE P IF PROTECTED
  JMS       @ YTYPE
SPCOUT, MEMA (240      /PRINT SPACE ON THE WAY OUT
  JMS @ YTYPE
  JMP      @PRFLNM
BITS16, 177777
DLTFLG, 0
DIRMOD, 0
DIRFLG, 0
DIRSIZ, 0
DIRPNT, 0
DRPNT2, 0
DRPSAV, 0
BIGMT, 0
PRMSK, 777777
FILNAM, 0
0
007700
EXTEN, 0
/OFF-PAGE POINTERS
YDIAFL, DIAFLG
YNEWNA, NEWNAM
YCRLF, CRLF
YTYPE, TYPE
YREAD, READ
YUNPCK, UNPCK
YFPCK, FPCK
YEXPLA, EXPLAN
YRECSI, RECSIZ
YDECPR, DECPRN
ZZZEND, BLOCK 20      /CHARACTER BUFFER
*115477
/TEXT FOR ERRORS ETC, FITS IN ABOVE BOOTS
ERR1, TEXT %FILE NOT FOUND%
ERR2, TEXT % %
ERR4, TEXT %ONE FILE AT A TIME, PLEASE!%
ERR5, TEXT %NO ROOM%
ERR5A, TEXT % %
ERR6, TEXT %DISK READ ERROR%
ERR8, TEXT %WHAT?%
FILE, TEXT % FILE %
SND, TEXT %SENDING %
/WATCH OUT FOR ///DIR BACKUP DIRECTORY
/IF FOUND, TAKE 1ST RA
/NORMALLY, TAKE 2ND RA
CKBDIR, 0
  MEMA NAME1
  ANDA PRMSK
  A-MZ BDIR
  JMP CKBOUT
  MEMA NAME2
  ANDA PRMSK

```

```

A-MZ BDIR+1
JMP CKBOUT
JMS @ YCRLF
JMS @ YCRLF
JMS @ YUNPCK
NSND
JMS @ YUNPCK
BDIR
JMS @ YCRLF
JMP @ CKBDIR
CKBOUT,MPOM CKBDIR
JMP @ CKBDIR
BDIR, TEXT %///DIR%
NSND, TEXT %NOT SENDING %
PAGSKP
*116000
/PAGE 116000 HOLDS MOST OF THE DISK-INTERACTIVE ROUTINES
/AND GENERAL UTILITIES
/
TOTLORD,0 /TOTAL WORDS LOADED TO DATE
ARWDS, 0 /WORDS PACKETED OUT TO DATE
/
WRDTOT,0 /TOTAL WORDS TO LOAD
WORDS, 0 /WORDS STILL TO LOAD
WRDSTO,0 /TOTAL OF WORDS STORED TO DATE
LASFLG,0 /SET =1 FOR FINAL RECORD
/
/PRINT INTRODUCTORY MESSAGE ON FIRST ENTRY
INTROD,0
MEMZ INTFLG
JMP @ INTROD
ONEM @ ZIGBFL
JMS @ ZCRLF
JMS DSKSET /CONFIGURE FOR DISK
JMS FPPMOV /MOVE FPP72 INTO POSITION
JMS TXPCK
INTMES
ONEM INTFLG
JMS @ ZCRLF
JMP @ INTROD
INTFLG,0
/CONFIGURE NICSYM FOR DISK IN USE
DSKSET,0
JMS DISKID /IDENTIFY F OR D
ACCM DSKTYP
MEMA (DPST1-UNIT /SET COUNTER
ACCM DSKCNT
MEMA FLSRT /SET PICK-UP POINTER
MEMZ DSKTYP
MEMA DLSRT
ACCM DSKPT1
MEMA DPST1 /SET PUT-DOWN POINTER
ACCM DSKPT2
DSKLUP, MEMA @ DSKPT1 /TRANSFER LOOP
ACCM @ DSKPT2
MPOM DSKPT1
MPOM DSKPT2
MMOMZ DSKCNT
JMP DSKLUP
MEMA DISOLV /FIX UP PAGE 110000

```

```

        ACCM @ DISPT
        JMP @ DSKSET
DSKTYP,0          /0=NIC-298, 1=NIC-294
DSKPT1,0
DSKPT2,0
DSKCNT,0
DISPT, WDISSO
/MOVE FPP72 FROM 102000 TO ITS WORKING POSITION AT 6000
FPPMOV,0
        MEMA FPPSRT
        ACCM FPPPIK
        MEMA FPPEND
        ACCM FPPPUT
        MEMA FPPSIZ
        ACCM FPPCNT
FPPLUP, MEMA @ FPPPIK
        ACCM @ FPPPUT
        MPOM FPPPIK
        MPOM FPPPUT
        MMOMZ FPPCNT
        JMP FPPLUP
        JMP @ FPPMOV
FPPSRT,102000
FPPEND,6000
FPPSIZ,1577
FPPPIK,0
FPPPUT,0
FPPCNT,0
/CONFIGURE NICSYM WITH USERNAME AND PASSWORD
CONFIG,0
        ONEM @ ZCMDFL
        MEMA @ FTST1      /CHECK FPP72 IS OK AT 102000
        A-MZ FVAL1
        JMP GETVIR
        MEMA @ FTST2
        A-MZ FVAL2
        JMP GETVIR
        ZERM NEWFLG
RENAM, JMS @ ZCRLF          /GET USERNAME
        JMS @ ZUNPCK
        CON
        JMS @ ZUNPCK
        UWORD
        MEMA USRNAM
        EXCT ZAC
        JMP UBAK
        JMS ASCPRN
        USRNAM
        MEMA (240
        JMS @ ZTYPE
UBAK,  JMS ASCINP
        USRNAM
        SKIP ZAC
        ONEM NEWFLG
        JMS @ ZUNPCK      /GET PASSWORD
        CON
        JMS @ ZUNPCK
        PWORD
        ONEM @ ZNULAF
        JMS ASCINP

```



```

PASWRD
ZERM @ ZNULAF
SKIP ZAC
ONEM NEWFLG
JMS @ ZCRLF
MEMZ NEWFLG      /STORE, IF ALTERED
JMS NVSTO
JMP @ CONFIG
GETVIR,JMS @ ZCRLF
JMS @ ZUNPCK
CONVIR
JMS @ ZCRLF
JMP @ CONFIG
NEWFLG,0          /=1 IF CHANGES MADE
FTST1, 102000
FVAL1, 25
FTST2, 103000
FVAL2, 2000767
CON, TEXT %CONFIGURE %
UWORD, TEXT %USERNAME: %
PWORD, TEXT %PASSWORD: %
CONVIR,TEXT %CONFIGURE *VIRGIN* KERMIT%
/INPUT ROUTINE FOR 8-BIT ASCII
ASCINP,0
MEMA @ ASCINP
MPOM ASCINP
ACCM ASCPNT
ACCM ASCSRT
MEMA (17          /MAXIMUM CHARACTERS
ACCM ASCCNT
JMS @ ZREAD
A-MZ (215
ZERZ
JMP ASCOUT
ZERZ
NEWASC,JMS @ ZREAD
A-MZ ("Q-100      /ABORT ON CTRL/Q
ZERZ
JMP @ ZCOMBA
A-MZ (210         /DELETE TO THE LEFT
ZERZ
JMP DELRIT
A-MZ (377         /OR RUBOUT
ZERZ
JMP RUBLFT
ACCM @ ASCPNT     /ACCEPT CHARACTER
A-MZ (215
ZERZ
JMP FINCOM        /JUMP OUT ON CR
MPOM ASCPNT
MMOMZ ASCCNT
JMP NEWASC
FINCOM,JMS @ ZCRLF
JMP @ ASCINP
ASCOUT,JMS @ ZCRLF
ZERA              /TO SHOW NULL INPUT
JMP @ ASCINP
RUBLFT,MEMA (210   /BACKSPACE ON RUBOUT
JMS @ ZTYPE
DELRIT,MMOMA ASCPNT /DELETE CHAR

```

```

A-MA ASCSRT      /BUT DO NOT GO BEYOND START
SKIP AC19
JMP DELOUT
MPOM ASCPNT      /TOO FAR, SO ADVANCE ONE SPACE
MEMA (240
JMS @ ZTYPE
DELOUT, MEMA (210
JMP NEWASC
ASCPNT, 0        /POINTER
ASCCNT, 0
ASCSRT, 0
/TRANSMIT 8-BIT ASCII OUT ON RS232 B-CHANNEL
/ADDRESS OF TEXT TO BE IN WORD FOLLOWING CALL
ASCTRN, 0
    MEMA @ ASCTRN
    MPOM ASCTRN
    ACCM RDCPNT
    MEMA @ RDCPNT
    MEMA (115
    ACCM REDCNT   /SAFETY STOP
NEWRDC, MEMA @ RDCPNT
    MPOM RDCPNT
    A-MZ (215
    ZERZ
    JMP RDCFIN    /JUMP OUT ON CR
    JMS @ ZACOB
    MMOMZ REDCNT
    JMP NEWRDC
RDCFIN, JMS @ ZACOB
    JMS @ ZEMPTO
    JMP @ ASCTRN
RDCPNT, 0
REDCNT, 0
/PRINT 8-BIT ASCII ON TTY
/ADDRESS OF TEXT TO BE IN WORD FOLLOWING CALL
ASCPRN, 0
    MEMA @ ASCPRN
    MPOM ASCPRN
    ACCM RDCPNT
    MEMA (115
    ACCM REDCNT
NUAPRN, MEMA @ RDCPNT
    MPOM RDCPNT
    A-MZ (215
    ZERZ
    JMP @ ASCPRN
    JMS @ ZTYPE
    MMOMZ REDCNT
    JMP NUAPRN
    JMP @ ASCPRN
/UNPACKING ROUTINE FOR LARGE TEXT BLOCKS
/GIVES A CRLF WHEN A 77 IS FOUND
/AND EXITS WHEN A 74 IS FOUND
TXPCK, 0
    MEMA (26
    ACCM TUNCNT
    MEMA @ TXPCK
    ACCM TX100
    MPOM TXPCK
TX200, MEMA @ TX100

```

```

        JMS TX300
TX600,MPOM TX100
        JMP TX200
TX300,0
        ACCM TX400
        RASH 14
        JMS TX500
        RASH 6
        JMS TX500
        JMS TX500
        JMP @ TX300
TX500,0
        ANDA (77
        A-MZ (77
        ZERZ
        JMP NEXLIN
        A-MZ (74
        ZERZ
        JMP @ TXPCK
        A+MA (240
        JMS @ ZTYPE
        MEMA TX400
        JMP @ TX500
NEXLIN,JMS @ ZCRLF
        MMOMZ TUNCNT
        JMP TX600
KBWAIT,JMS @ ZSREAD
        A-MZ ("Q
        ZERZ
        JMP @ TXPCK
        A-MZ (215
        ZERZ
        JMP ONELIN
        A-MZ (240
        JMP KBWAIT
        MEMA (26
        ACCM TUNCNT
        JMP TX600
ONELIN,ONEM TUNCNT
        JMP TX600
TX100,0
TX400,0
TUNCNT,0
/LOAD THE DIRECTORY AT 3000 (D) OR 4000 (F)
/USED IN TRANSMIT MODE
DIRLOD,0
        ZERM @ ERRFLG
        MEMA UNIT           /SET UNIT & RECORD
        A+MA (3
        ACCM SDIREC
        MEMA RECSIZ         /SET NIMBER OF WORDS
        ACCM SDWRDS
        MEMA SAVSRT         /SET DIR START
        ACCM DIRSRT
        ZERA
        JMS @ DISK          /DO THE LOAD
SDIREC,0
SDWRDS,0
DIRSRT,0
        MEMA DIRSRT         /INITIALISE DIR POINTER

```

```

A-MA (5
ACCM @ ZDIRPN
MEMA @ ERRFLG      /CHECK FOR ERRORS
EXCT ZAC
JMP @ DIRLOD      /ALL IS WELL
JMS @ ZCRLF        /REPORT ERROR
JMS @ ZUNPCK
ERR6
ZERM @ ERRFLG
JMP @ ZCOMBA
/PREPARE TO LOAD FILE
LODSET,0
    ZERM LASFLG
    MEMA @ ZSIZE
    ACCM WORDS
    ACCM WRDTOT
    ZERM WRDSTO
    ZERM TOTLOD
    ZERM ARWDS
    MEMA RECSIZ
    ACCM REDWRD
FINSET, MEMA UNIT
    A+MA @ ZFRSTR
    AMOM REDREC
    JMP @ LODSET
/FIND LARGEST EMPTY SPACE ON DISK
GETPAD,0
    JMS DIRFIN
    ZERM @ OARG2
    MONM @ DISOLV
    JMS @ DIRFUN
    1
    2
    NULNAM
    ZERZ
    STOP
    MEMA @ OARG2
    EXCT AC19
    ANGA
    ACCM PADSIZ
    MEMA @ OARG1
    ACCM PADST
    A+MA UNIT
    ACCM RITREC
    ZERM @ ERRFLG
    JMP @ GETPAD
PADST, 0      /STARTING RECORD OF SPACE
PADSIZ,0      /SIZE OF SPACE
NULNAM,0
/MAKE THE DIRECTORY ENTRY
DIRENT,0
    JMS DIRFIN
    MEMA PADST
    ACCM @ OARG1
    MEMA WRDSTO
    ACCM @ OARG2
    MEMA @ ZMLA
    ACCM @ OARG3
    MEMA @ ZPSA
    ACCM @ SYSTRT

```

```

MEMA @ ZFILN1
LLSH 2
SKIP AC0
APOA
RLSH 2
MEMZ @ ZASFLA
ACCM @ ZFILN1
JMS @ DIRFUN
1
1
FILNAM
STOP
JMP @ DIRENT
DATMEM,100000
/CALL DIRFUN INTO CORE
/USED ONLY IN RECEIVE MODE
DIRFIN,0
MEMA (5
ACCM RETRYS
RETRY, ZERM @ ERRFLG
MEMA UNIT
A+MA (7
ACCM DFREC
ZERA
JMS @ DISK
DFREC, 0
600
7000
ZERM @ DEVDIR
MMOZ @ ERRFLG
JMP @ DIRFIN
MMOMZ RETRYS
JMP RETRY
JMS @ ZUNPCK
ERR6
JMS @ ZCRLF
ZERM @ ERRFLG
JMP @ ZCOMBA
RETRYS,0
/PREPARE TO LOAD <=1 RECORD
NEXREC,0
MPOM REDREC          /BUMP RECORD
MEMA WORDS           /MORE THAN RECSIZ TO GO?
A-MA RECSIZ
EXCT AC19 ZAC
JMP LASMOV           /NO, LAST LOAD
MEMA RECSIZ          /YES
M-AM WORDS           /DECREMENT WORDS
JMP @ NEXREC
LASMOV,ONEM LASFLG    /FLAG FINAL TRANDFER
MEMA WORDS           /SET WORD COUNT
ACCM REDWRD
JMS CLDBUF           /CLEAR DISK BUFFER
JMP @ NEXREC
/LOAD UP TO ONE RECORD AT 0
LOAD, 0
MEMA (5
ACCM NUTRYS
NEXTRY,ZERM @ ERRFLG
ZERA

```

```

        JMS @ DISK
REDREC,0
REDWRD,0
        0
        MEMA REDWRD
        A+MM TOTLOD
        MEMA @ ERRFLG
        EXCT ZAC
        JMP @ LOAD           /ALL IS WELL
        MMOMZ NUTRYS        /DISK READ ERROR
        JMP NEXTRY          /FOUR MORE TRIES
        JMS @ ZUNPCK
        ERR6
        JMS @ ZCRLF
        ZERM @ ERRFLG
        JMP @ LOAD
NUTRYS,0
/WRITE UP TO ONE RECORD FROM 0, DON'T OVERRUN PADSIZ
/BUMP THE RECORD NUMBER
WRITE, 0
        MEMA PADSIZ
        A-MA WRDSTO
        EXCT AC19 ZAC
        JMP FULLUP
        ONEA
        JMS @ DISK
RITREC, 0
RITWRD, 0
        0
        MEMA RITWRD
        A+MM WRDSTO
        MPOM RITREC
        JMP @ WRITE
FULLUP, JMS @ ZCRLF
        JMS @ ZUNPCK
        FU
        JMS @ ZCRLF
        ONEM @ ZCZFLG
        JMP @ WRITE
FU,      TEXT %DISK FULL%
/SAVE THE MSA
SAVE, 0
        MPOA UNIT
        ACCM KUNIT
        MEMA SAVWRD
        ACCM SWRD
        MEMA SAVSRT
        ACCM SSRT
        ONEA
        JMS @ DISK
KUNIT,0
SWRD, 0
SSRT, 0
        JMP @ SAVE
/IDENTIFY DISK UNIT CURRENTLY IN USE
/RETURNS WITH RESULT IN ACC: 0=FLOPPY, 1=DIABLO
DISKID,0
        MEMA @ IDPNT
        A-MZ IDNUM
        JMP DIABID

```

```

        ZERA
        JMP @ DISKID      /FLOPPY DISK
DIABID,ONEA
        JMP @ DISKID      /DIABLO
IDPNT, 7602
IDNUM, 1001
/DO A SUMMARY DIRECTORY LISTING
DIRLST,0
        ONEM @ ZCMDFL
        JMS @ ZCRLF
        JMS @ ZCRLF
        JMS @ ZUNPCK
        DLST
        JMS SAVE
        JMS DIRFIN
        JMS @ DIRFUN
        1
        0
        1
        ACCA
        JMS @ DISOLV
        JMP @ DIRLST
DLST, TEXT % - DIRECTORY -%
USRNAM,BLOCK 20
PASWRD,BLOCK 20
/RE-STORE KERMIT AFTER CONFIGURATION
NVSTO, 0
        ZERM INTFLG
        JMS SAVE
        JMS DIRFIN      /FIND KERMIT ON DISK
        JMS @ DIRFUN
        1
        2
        NVNAM
        JMP NONSYM
        JMS @ DISOLV
        MEMA @ OARG1
        A+MA UNIT
        ACCM NVREC
        ONEA      /REPLACE WITH NEW VERSION
        JMS @ DISK
NVREC, 0
        16000
        102000
        ONEM INTFLG
        JMP @ NVSTO
NONSYM,JMS @ DISOLV
        JMS @ ZCRLF
        JMS @ ZUNPCK
        NVNAM
        JMS @ ZUNPCK
        MISING
        JMS @ ZCRLF
        ONEM INTFLG
        JMP @ ZCOMBA
MISING,TEXT % NOT FOUND, SO CANNOT CONFIGURE%
NVNAM, TEXT %KERMIT%
/CLEAR DISK BUFFER 0-2000/3000
CLDBUF,0
        ZERM CLBPNT

```

```

        MEMA RECSIZ
        ACCM CLBCNT
CBLUP, ZERM @ CLBPNT
        MPOM CLBPNT
        MMOMZ CLBCNT
        JMP CBLUP
        JMP @ CLDBUF
CLBPNT, 0
CLBCNT, 0
/INITIALISE ARROW POSITION ON LHS
/WITH ARWDS=0 AND A DEFAULT WRDTOT
SARROW, 0
        ZERM ARWDS
        MPOA (1777
        ACCM WRDTOT
        JMP @ SARROW
/DISPLAY AN ARROW ON THE CRT TO INDICATE PROGRESS, IF RTFLG=1
/A DIFFERENT ELEMENT GETS DISPLAYED ON EACH ENTRY
ARROW, 0
        MPOMA ARNUM
        A-MZ (1
        ZERZ
        JMP ARCALC
        A-MZ (2
        ZERZ
        JMP LHSDOT
        A-MZ (3
        ZERZ
        JMP SHAFT
        A-MZ (4
        ZERZ
        JMP HEAD
        ZERM ARNUM
        JMP RHSDOT
ARNUM, 0
/CALCULATE POSITION OF TIP OF ARROW FROM
/  ARTIP = SHFSIZ + (ARWDS/WRDTOT)*(TOPHOR - SHFSIZ)
ARCALC, MEMA WRDTOT
        ACCM ARDIV
        MEMA TOPHOR
        A-MA SHFSIZ
        ACCM ARMUL
        MEMA ARWDS
        TACMQ
        MULT
ARMUL, 0
        CLL
        ACCM ARHIGH
        TMQAC
        ACCM ARLOW
        EXCT AC19
        STL
        LASH 1
        TACMQ
        MEMA ARHIGH
        LASH 1
        EXCT L
        APOA
        DIVD
ARDIV, 0

```



```

    TMQAC
    A+MA SHFSIZ
    ACCM ARTIP
    JMP @ ARROW
TOPHOR,37774
ARHIGH,0
ARLOW, 0
ARTIP, 0
/DISPLAY A VERTICAL LINE ON THE LHS TO SHOW START
LHSDOT,ZERA
    JMS MIDDOT
    JMP @ ARROW
/DISPLAY A VERTICAL LINE ON THE RHS TO SHOW DESTINATION
RHSDOT,MEMA TOPHOR
    JMS MIDDOT
    JMP @ ARROW
/DISPLAY VERTICAL LINE AT X-POSITION IN ACC ON ENTRY
MIDDOT,0
    TACXD
    MEMA DOTSRT
    ACCM DOTHIT
    MEMA (10
    ACCM DOTCNT
    ZERA
DOTLUP,MEMA DOTINC
    A+MMA DOTHIT
    TACYD INTENS
    MMOMZ DOTCNT
    JMP DOTLUP
    JMP @ MIDDOT
DOTCNT,0
DOTHIT,0
DOTINC,10000
DOTSRT,140000
/DRAW SHAFT OF ARROW
/RHS X-CRT-VALUE TO BE AR ARPOS
SHAFT, MEMA ARTIP
    A-MA SHFSIZ
    ACCM SHFPNT
    TACXD
    MEMA SHFNUM
    ACCM SHFCNT
SHFLUP,MEMA YHITE
    TACYD INTENS
    MEMA SHFINC
    A+MMA SHFPNT
    TACXD
    MMOMZ SHFCNT
    JMP SHFLUP
    JMP @ ARROW
SHFNUM,10
SHFCNT,0
SHFSIZ,1200
SHFPNT,0
SHFINC,120
/DRAW HEAD OF ARROW WITH ITS APEX ON X-AXIS AT ARPOS
HEAD, MEMA HEDNUM
    ACCM HEDCNT
    MEMA ARTIP
    ACCM XHEDPT

```

```

TACXD
MEMA YHITE
ACCM YHEDP1
ACCM YHEDP2
TACYD INTENS
HEDLUP, MEMA HXINC
M-AMA XHEDPT
TACXD
MEMA HYINC
A+MMA YHEDP1
TACYD INTENS
MEMA HYINC
M-AMA YHEDP2
TACYD INTENS
MMOMZ HEDCNT
JMP HEDLUP
JMP @ ARROW

HEDNUM, 5
HEDCNT, 0
XHEDPT, 0
YHEDP1, 0
YHEDP2, 0
HXINC, 60
HYINC, 10000
YHITE, 200000
/ARROW TESTER FOR NICBUG
ARWTST, 0
MEMA AWTNUM
ACCM AWCNT
AWTLUP, JMS ARROW
MMOMZ AWCNT
JMP AWTLUP
JMP @ ARWTST
AWTNUM, 100000
AWCNT, 0
/OFFER HELP
EXPLAN, 0
JMS @ ZCRLF
JMS TXPCK
HLPPLP
JMP @ EXPLAN
/PRINT INSTRUCTIONS
HELP, 0
ONEM @ ZCMDLFL
JMS @ ZCRLF
JMS @ ZCRLF
JMS TXPCK
COMSUM
JMS @ ZCRLF
JMP @ HELP
/PRINT NOTES ON OPERATION
NOTPRN, 0
ONEM @ ZCMDLFL
JMS @ ZCRLF
JMS @ ZCRLF
JMS TXPCK
NOTES
JMS @ ZCRLF
JMP @ NOTPRN
/TYPE DECIMAL INTEGER

```

/NUMBER TO BE IN ACC ON ENTRY

DECPRN,0

ZERM DIGCNT

DECOU1, LASH 1

TACMQZ

DIVD

12 /NUMBER BASE = DECIMAL

RISH 1

ACCM @DIGPTR

MPOM DIGPTR /PUSH A DIGIT

MPOM DIGCNT

TMQAC

SKIP ZAC

JMP DECOU1

DECOU2, MMOM DIGPTR

MEMA @DIGPTR /POP A DIGIT

A+MA (260

JMS @ ZTYPE

MMOMZ DIGCNT

JMP DECOU2

JMP @DECPRN

DIGCNT,0

DIGPTR, #+1

BLOCK 7 /SPACE FOR 7 DIGITS

/FLOPPY DISK PARAMETERS

FUNIT, 1000

FRECSZ,2000

FMAXRC,231

FDISOL,7747

FERRFL,7735

FDISK, 7613

FSVWRD,3600

FSVSRT,4000

FLSRT, FUNIT

/DIABLO DISK PARAMETERS

DUNIT, 100000

DRECSZ,3000

DMAXRC,624

DDISOL,7751

DERRFL,7704

DDISK, 7612

DSVWRD,4600

DSVSRT,3000

DLSRT, DUNIT

/PARAMETERS FOR DISK IN USE

UNIT, 0 /1000 (OR 100000)

RECSIZ,0 /2000 (OR 3000)

MAXREC,0 /231 (OR 624)

DISOLV,0 /7747 (OR 7751)

ERRFLG,0 /7735 (OR 7704)

DISK, 0 /7613 (OR 7612)

SAVWRD,0 /NO. OF WORDS TO SAVE

SAVSRT,0 /START OF SAVE AREA

DPST1, UNIT

/COMMON DISK PARAMETERS

MONITOR,7600

OARG1, 7770

OARG2, 7771

OARG3, 7772

SYSTRT,7760