```
↑
P$

//  STARTSLAVE.....................................................
//
//
//
//
//
//  START UP SLAVE SYSTEM
//
LET STARTSLAVE( HOWMANY ) BE [ LET I=Ø AND BUFBASE=Ø AND
        BUF=VEC 1Ø
//
//          DESTROY POSSIBLE EXISTING SLAVE DATA STRUCT
     KILLSTRUCT()
//
//          CREATE SLAVE SYSTEM DATA STRUCTURE
//
//          CREATE MAIN CLIST FOR SLAVES
     SXJ( CRECLIST, ALLOC, GETCLIST('SCLIST'), SLAVECHAN+2*HOWMANY )
//
//          CREATE DATA FILE
     I ← PTRBASE + HOWMANY * (BUFLEN + 4)
     SXJ( CREFILE, ALLOC, GETCLIST('TEMP'), 1, LV I )
     SXJ( CREBLK, GETCLIST('TEMP'), Ø )
     SXJ( CAPOUT, GETCLIST('SCLIST'), SLAVEFILE, GETCLIST('TEMP') )
//
//          CREATE SLAVE LOGON COUNTER
     SXJ( CREVENT, ALLOC, GETCLIST('TEMP'), 2 )
     SXJ( SENDEV, GETCLIST('TEMP'), Ø )
     SXJ( CAPOUT, GETCLIST('SCLIST'), SLAVECOUNT, GETCLIST('TEMP') )
//
//          CREATE MASTER WAKE-UP CHANNEL
     SXJ( CREVENT, ALLOC, GETCLIST('TEMP'), HOWMANY+2 )
     SXJ( CAPOUT, GETCLIST('SCLIST'), MASTER, GETCLIST('TEMP') )
//
//          CREATE SLAVE EVENT CHANS AND INNITIAL SLAVE BUFFER POINTERS
     I ← HOWMANY
     WHILE ( I > Ø ) DO
          [ I ← I - 1
          SXJ( CREVENT, ALLOC, GETCLIST('TEMP'), 2 )
          SXJ( CAPOUT, GETCLIST('SCLIST'), SLAVECHAN+ I*2 + WAKEUP,
               GETCLIST('TEMP') )
          SXJ( CREVENT, ALLOC, GETCLIST('TEMP'), 2 )
          SXJ( CAPOUT, GETCLIST('SCLIST'), SLAVECHAN + I*2 + REPORT,
               GETCLIST('TEMP') )
//
          BUFBASE ← PTRBASE + HOWMANY*4 + I*BUFLEN
          POINTER.FIRST ← BUFBASE
          POINTER.IN ← BUFBASE
          POINTER.OUT ← BUFBASE
          POINTER.LIMIT ← BUFBASE + BUFLEN
          SXJ( WRITE, IND(GETCLIST('SCLIST'), SLAVEFILE),
               PTRBASE + I*4, LV(POINTER.FIRST), 4 )
          ]
```

```
//
//        STORE NUMBER OF SLAVES
      $ SCOUNT ← HOWMANY
      SXJ( WRITE, IND(GETCLIST('SCLIST'), SLAVEFILE), NUMSLAVES,
           SCOUNT, 1 )
//
//        MOVE SLAVE C-LIST CAP TO MAINCL
      SXJ( CAPOUT, MAINCL, MSLAVE, GETCLIST('SCLIST') )
      $ RPTINDX ← Ø
//
//        SEND REPORT COMMAND TO ALL SLAVES
//
//        COLLECT REPORTS FROM ALL SLAVES
      I ← Ø
      WHILE ( I < HOWMANY ) DO
           [ ($ SLAVON).I ← TRUE; I ← I + 1 ]
           L1: IF NOT GETREPORT($ SLAVON) DO
           [ OUTPUT("NOT ALL SLAVES ON *N")
           L2: SWITCHON ( GETCOMMAND (BUF) ) INTO
                [ CASE 'RETRY': GOTO L1
                CASE 'SKIP': GOTO NXTCMD
                DEFAULT: OUTPUT("TYPE 'SKIP' OR 'RETRY' *N")
                     GOTO L2
                ]
           ]
//
      OUTPUT("SLAVE SYSTEM UP *N")
      RETURN
      ]
//
//   ...STARTSLAVE
//
//
//   KILLSTRUCT.....................................................
//
//   DESTROY SLAVE DATA STRUCTURE
//
LET KILLSTRUCT () BE [ LET COUNT = Ø
//
//        CHECK THAT STRUCTURE EXISTS
      IF SLAVEUP() DO
//
//        READ NUMBER OF SLAVES
           [ SXJ( READ, IND(GETCLIST('SCLIST'), SLAVEFILE), NUMSLAVES
                LV COUNT, 1 )
//
//        DESTROY SLAVE EVENT CHANNELS
           I ← COUNT
           WHILE ( I > Ø ) DO
                [ I ← I + 1
                SXJ( DESTEVENT, IND(GETCLIST('SCLIST'), SLAVECHAN + 2*I )
                SXJ( DESTEVENT, IND(GETCLIST('SCLIST'), SLAVECHAN + 2*I+1 )
                 ]
//
//         DESTROY SLAVE COUNTER
           SXJ(DESTEVENT, IND(GETCLIST('SCLIST'), SLAVECOUNT) )
//
//         DESTROY MASTER WAKE UP CHANNEL
           SXJ( DESTEVENT, IND(GETCLIST('SCLIST'), MASTER) )
```

```
//
//          DESTROY SLAVE DATA FILE
            SXJ( DESTBLK, IND(GETCLIST('SCLIST'), SLAVEFILE), Ø )
            SXJ( DESTFILE, IND(GETCLIST('SCLIST'), SLAVEFILE) )
//
//          DESTROY SLAVE C-LIST
            SXJ( DESCLIST, GETCLIST('SCLIST') )
//
//          CLEAR SLAVE C-LIST CAP IN MAINCL
            SXJ( ZEROCAP, GETCLIST('SCLIST') )
            SXJ( CAPOUT, MAINCL, MSLAVE, GETCLIST('SCLIST') )
            ]
//
     OUTPUT("SLAVE DATA STRUCTURE KILLED *N")
     RETURN
     ]
//
//   ...KILLSTRUCT
//
//   SLAVEUP.....................................................
//
//
//
//
//
//   CHECK FOR SLAVE SYSTEM UP
//
LET SLAVEUP () = VALOF
     [ SXJ( CAPIN, MAINCL, MSLAVE, GETCLIST('SCLIST') )
     SXJ( DISPCAP, GETCLIST('SCLIST') )
     TEST ( $ XREG6 = Ø ) THEN RESULTIS FALSE OR RESULTIS TRUE
     ]
//
//   ...SLAVEUP
//
//   ENSLAVE.....................................................
//
//
//
//
//
//   BECOME A SLAVE
//
LET ENSLAVE() BE [ LET BUF = VEC 1Ø
     $ EXIT ← DONE
//
//          CHECK FOR MASTER AT HOME
     IF NOT SLAVEUP() DO [ OUTPUT("MASTER NOT HOME *N"); RETURN ]
//
//          SIGN ON
     SXJ( READ, IND(GETCLIST('SCLIST'), SLAVEFILE), NUMSLAVES,
          SCOUNT, 1 )
     SXJ( GETEVH, IND( GETCLIST('SCLIST'), SLAVECOUNT ) )
     IF ($ XREG7 = $ SCOUNT ) DO
          [ SXJ( SENDEV, IND(GETCLIST('SCLIST'), SLAVECOUNT),
               $ SCOUNT )
          OUTPUT("MASTER'S HOUSE IS FULL *N")
          RETURN
          ]
     $ MYINDEX ← $ XREG7
     SXJ( SENDEV, IND( GETCLIST('SCLIST'), SLAVECOUNT ), $ MYINDEX + 1 )
//
     OUTPUT("I AM SLAVE NUMBER.. *N")
     OUTPUT(OCTALTOCHAR( BUF, $ MYINDEX ) )
```

```
//
//        GO TO WORK
     SLAVEAWAY()
     DONE: OUTPUT("I'M FREE..GLORY BE *N")
     RETURN
//
     ]
//
//   ...ENSLAVE
//
//   SLAVEAWAY...
//
//
//
//
//
//   SLAVE STAYS IN HERE AS LONG AS HE IS UP
//
LET SLAVEAWAY() BE [ LET PTR=VEC 4 AND BUFFER = VEC BUFLEN AND
          I=0 AND COUNT=0 AND ACTION=VEC 10
//
//   WAIT FOR WORK
     WAIT: SXJ( GETEVH, IND(GETCLIST('SCLIST'), SLAVECHAN +
          $ MYINDEX * 2 + WAKEUP )
//
//        GOT WORK.. READ BUFFER POINTERS
     NEXT: SXJ( READ, IND(GETCLIST('SCLIST'), SLAVEFILE), PTRBASE +
          $ MYINDEX * 4, LV(PTR.FIRST), 4 )
     SXJ( READ, IND(GETCLIST('SCLIST'), SLAVEFILE ), PTR.FIRST,
          BUFFER, BUFLEN )
//
//        CHECK FOR BUFFER EMPTY
     IF (PTR.IN = PTR.OUT) DO GOTO WAIT
//
//        PROCESS ONE COMMAND
     I ← 0
     COUNT ← BUFFER.(PTR.OUT - PTR.FIRST)
     IF (COUNT > BUFLEN) DO
          [ OUTPUT("BUFFER ERROR *N"); GOTO $ SEXIT ]
     PTR.OUT ← PTR.OUT + 1
     IF (PTR.OUT = PTR.LIMIT) DO PTR.OUT ← PTR.FIRST
     WHILE ( COUNT > 0 ) DO
          [ ACTION.I ← BUFFER.(PTR.OUT - PTR.FIRST)
          PTR.OUT ← PTR.OUT + 1
          IF (PTR.OUT = PTR.LIMIT) DO PTR.OUT ← PTR.FIRST
          I ← I+1
          COUNT ← COUNT - 1
          ]
//
//        WRITE BACK BUFFER POINTER
     SXJ( WRITE, IND(GETCLIST('SCLIST'), SLAVEFILE), PTRBASE +
          MYINDEX*4 + OUT, LV (PTR.OUT), 1 )
     SXJ( SENDEV, IND(GETCLIST('SCLIST'), SLAVECHAN + $MYINDEX*2 +
          REPORT, $ MYINDEX )
//
//        DO THE ACTION
     ACTION.0( ACTION.1, ACTION.2, ACTION.3, ACTION.4, ACTION.5,
          ACTION.6, ACTION.7, ACTION.8, ACTION.9, ACTION.10 )
//
//
     GOTO NEXT
     ]
//
//   ...SLAVEAWAY
```

```
//
//
//  STOPSLAVE.....................................
//
//
//
//
//
//   TURN OFF SLAVES
//
LET STOPSLAVE () BE [ LET I=Ø AND PTR=VEC 4 AND CHAN=VEC 2
          AND BUF=VEC 1Ø
//
     IF NOT SLAVEUP() DO
          [ OUTPUT("SLAVE SYSTEM NOT UP *N"); RETURN ]
//
//        SEND QUIT COMMAND AND WAIT FOR BUFFER TO CLEAR
     I ← Ø
     WHILE ( I < $ SCOUNT ) DO
          [ IF ($ SLAVON).I) DO
               [ CMD( I, FREESLAVE, 'STOP' )
               SXJ( READ, IND(GETCLIST('SCLIST'), SLAVEFILE), PTRBASE
                    + I*4, LV(PTR.FIRST), 4 )
               IF ( PTR.IN NE PTR.OUT ) DO
                    [ CHAN.Ø ← IND(OPERCL, ONEMIN)
                    CHAN.1 ← IND(GETCLIST('SCLIST'), SLAVECHAN +
                         I*2 + REPORT )
                    SXJ( GETEVMH, CHAN, 2 )
                    IF ( UNPACK ($ XREG6 ) = 1 ) DO
                         [ OUTPUT("SLAVE SLEEPY...*N")
                         OUTPUT(OCTALTOCHAR( BUF, I ) )
                         L2: SWITCHON( GETCOMMAND( BUF ) ) INTO
                              [ CASE 'SKIP': GOTO L3
                              CASE 'RETRY': GOTO L1
                              DEFAULT: OUTPUT("TYPE 'SKIP' OR 'RETRY'*N")
                               ]
                         ]
                    SXJ( READ, IND(GETCLIST('SCLIST'), SLAVEFILE),
                         PTRBASE + I*4, LV(PTR.FIRST), 4 )
                    ]
               ]
          L3: I ← I + 1
          ]
//
//        DESTROY SLAVE DATA STRUCTURE
     KILLSTRUCT ()
     RETURN
     ]
//
//  ...STOPSLAVE
// FREESLAVE.....................................................
//
//
//
//
//
//   ALL DONE BEING A SLAVE
//
//
LET FREESLAVE() BE GOTO $ SEXIT
//
//  ...FREESLAVE
```

```
//  GETREPORT.....................................................
//
//
//
//
//
//    GET REPORT FROM A SET OF SLAVES
//
LET GETREPORT( ROLLCALL ) = VALOF [  LET BUF=VEC 10 AND CHAN=VEC 2
        AND I=0 AND COUNT=0
//
//          CLEAR MASTER REPORTING CHANNEL
     FRETAD ← L2
     L1: SXJ( GETEVF, IND(GETCLIST('SCLIST'), MASTER) )
     OUTPUT("EXTRA REPORTS *N")
     GOTO L1
     L2: FRETAD ← 0
//
//          SEND REPORT COMMAND TO ALL
     I ← 0; COUNT ← 0
     WHILE ( I < $ SCOUNT ) DO
          [ IF ( ROLLCALL.I AND ($ SLAVON).I ) DO
               [ CMD( I, DOREPORT, $ RPTINDX, 'STOP' )
               COUNT ← COUNT + 1
               ]
          I ← I + 1
          ]
//
//          COLLECT REPORTS
     $ RPTINDX ← $ RPTINDX + 1
     CHAN.0 ← IND( OPERCL, ONEMIN )
     CHAN.1 ← IND( GETCLIST('SCLIST'), MASTER )
     WHILE ( COUNT > 0 ) DO
          [ L1: SXJ( GETEVMH, CHAN, 2 )
          IF( UNPACK($ XREG6) = 1 ) DO
               [ OUTPUT("WAITING FOR REPORT *N")
               L2: SWITCHON( GETCOMMAND ( BUF ) ) INTO
                    [ CASE 'SKIP': RESULTIS FALSE
                    CASE 'RETRY': GOTO L1
                    DEFAULT: OUTPUT("TYPE 'RETRY' OR 'SKIP' *N")
                         GOTO L2
                    ]
               ]
          TEST (($ XREG7) RSHIFT 18 = $ RPTINDX-1 )
               THEN COUNT ← COUNT+1
               OR OUTPUT("FALSE REPORT *N")
          ]
     RESULTIS TRUE
     ]
//
//
//    ...GETREPORT
```

```
//
//  DOREPORT.....................................
//
//
//
//
//
//  SEND REPORT TO MASTER
//
LET DOREPORT (DATUM) BE
     [ SXJ( SENDEV, IND(GETCLIST('SCLIST'), MASTER),
          ( DATUM ↑ 18 ) +. $ MYINDEX )
     RETURN
     ]
//
//  ...DOREPORT
//  CMD..........................................
//
//
//
//
//
//  SEND COMMAND TO A SLAVE
//
LET CMD( INDEX, WHOM, P1, P2, P3, P4, P5, P6, P7, P8, P9, P10 ) BE [
          LET COUNTLOC=0 AND PTR=VEC4 AND BUFFER=VEC BUFLEN
          AND SCR=VEC 10
//
//  CHECK INDEX OF SLAVE
     IF( (INDEX < 0) OR (INDEX GE $ SCOUNT) ) DO
          [ OUTPUT("BAD SLAVE INDEX *N"); GOTO NXTCMD ]
     IF ( NOT ($ SLAVON).INDEX ) DO RETURN
//
//        READ BUFFER POINTERS AND BUFFER
     SXJ( READ, IND(GETCLIST('SCLIST'), SLAVEFILE), PTRBASE +
          INDEX*4, LV( PTR.FIRST ), 4 )
     SXJ( READ, IND(GETCLIST('SCLIST'), SLAVEFILE), PTR.FIRST,
          BUFFER, BUFLEN )
//
//        COPY COMMAND TO BUFFER
     COUNTLOC ← PTR.IN
     NEXT ← LV INDEX
     COUNT ← 0
     WHILE ( NEXT.COUNT NE 'STOP' ) DO
          [ BUFFER.(PTR.IN - PTR.FIRST) ← NEXT.COUNT
          PTR.IN ← PTR.IN +1
          IF (PTR.IN = PTR.LIMIT) DO PTR.IN ← PTR.FIRST
//
//        CHECK FOR FULL BUFFER
          WHILE ( PTR.IN = PTR.OUT ) DO
               [ CHAN.0 ← IND(OPERCL, ONEMIN)
               CHAN.1 ← IND(GETCLIST('SCLIST'), SLAVECHAN + INDEX*2
                    + REPORT )
               L1: SXJ( GETEVMH, CHAN, 2 )
               IF ( UNPACK ($ XREG6 ) = 1 ) DO
                    [ OUTPUT("BUFFER FULL: NUM= *N")
                    OUTPUT(OCTALTOCHAR( SCR, INDEX ) )
                    L2: SWITCHON ( GETCOMMAND ( SCR ) ) INTO
                         [ CASE 'SKIP': ($ SLAVON).INDEX ← FALSE
                              RETURN
                         CASE 'RETRY': GOTO L1
                         DEFAULT: OUTPUT("TYPE 'SKIP' OR 'RETRY' *N)
                              GOTO L2
                         ]
                    ]
```

```
//
                SXJ( READ, IND(GETCLIST('SCLIST'), SLAVEFILE), PTRBASE +
                     INDEX*4 + OUT, LV(PTR.OUT), 1 )
                ]
//
          COUNT ← COUNT +1
          IF (COUNT > BUFLEN) DO
               [ OUTPUT("NO 'STOP' *N"); GOTO NXTCMD ]
          ]
//
//        DONE..SET COUNT AND WRITE BUFFER AND POINTER AND SEND WAKE UP
     BUFFER.(COUNTLOC - PTR.FIRST) ← COUNT - 1
     SXJ( WRITE, IND(GETCLIST('SCLIST'), SLAVEFILE), PTR.FIRST,
          BUFFER, BUFLEN )
     SXJ( WRITE, IND(GETCLIST('SCLIST'), SLAVEFILE), PTRBASE +
          INDEX*4 + IN, LV(PTR.IN), 1 )
     SXJ( SENDEV, IND(GETCLIST('SCLIST'), SLAVECHAN + INDEX*2
          + WAKEUP, Ø )
//
     RETURN
     ]
//
//  ...CMD
```