```
(RPAQQ LISPDIAGNOSTICSCOMS
       [(COMS (* "This would be a good one for the system to have")
              (FNS UNSHADEALLITEMS printout.SUBR))
        (INITVARS (DIAGNOSTICSRECORDSTREAM T)
               (DIAGNOSTICSCONTROLWINDOW.POSITION (create POSITION XCOORD _ 800 YCOORD _ 80)))
        (VARS (EXERCISE.RUNNER NIL)
              (EXERCISE.STATE (QUOTE STOP))
              EXERCISE.POSSIBILITIES)
        (CONSTANTS \LD.INFOSHADE)
        (GLOBALVARS DIAGNOSTICSRECORDSTREAM DIAGNOSTICSCONTROLWINDOW.POSITION EXERCISE.RUNNER EXERCISE.STATE
               EXERCISE.POSSIBILITIES)
        (FNS EXERCISE \LD.BLOCKCHECK \LD.STOPPROCS)
        (COMS (* "Some user-interface, menu-like things")
              (CONSTANTS (\LD.DPM.MENUBORDERSIZE 3))
              (INITVARS (\LD.DCW.WINDOW NIL)
                     (\LD.DPM.MENU NIL)
                     (\LD.DPM.WINDOW NIL)
                     (\LD.DPM.WINDOWBORDERSIZE NIL)
                     (\LD.DPM.ITEMS NIL)
                     (\LD.DPM.SPACEWIDTH NIL))
              (FNS MAKEDIAGNOSTICSMENU \LD.DCW.WHENSELECTED \LD.DPM.WHENSELECTED)
              (GLOBALVARS \LD.DCW.WINDOW \LD.DPM.MENU \LD.DPM.WINDOW \LD.DPM.WINDOWBORDERSIZE \LD.DPM.FONTHEIGHT
                     \LD.DPM.ITEMS \LD.DPM.SPACEWIDTH))
        (DECLARE: EVAL@COMPILE DONTCOPY (RECORDS EXERCISE.POSSIBILITIES)
               (MACROS ldprintout))
        (COMS (* "Various background activities to stress hardware")
              (FNS DSKPROC DSKPROC.AUX DSKPROC.DO1COPY ETHERPROC DAEMONPROC))
        (COMS (* "Various diagnostic and benchmark activities")
              (FNS EMUPROC 20RECLAIM)
              (* "After the TANSPEED benchmark")
              (FNS \LD.TANSPEED)
              (* "Extraction from Gabriel's BROWSE benchmark")
              (FNS \LD.BROWSE \LD.BROWSEINIT \LD.BROWSEMATCH)
              (VARS (!BROWSEINIT NIL))
              (GLOBALVARS !BROWSEINIT)
              (DECLARE: EVAL@COMPILE DONTCOPY (MACROS CHAR1)))
        (DECLARE: DONTEVAL@COMPILE DONTEVAL@LOAD DOCOPY (* "PPLossage")
              (ADDVARS (DISPLAYFONTEXTENSIONS STRIKE)
                     (DISPLAYFONTDIRECTORIES {FLOPPY})
                     (LISPUSERSDIRECTORIES {FLOPPY}))
              (VARS (!MTUSERAIDFLG NIL))
              (FILES (COMPILED FROM VALUEOF LISPUSERSDIRECTORIES)
                     PAGEHOLD MACROTESTAUX MACROTEST PLURAL)
              (P (MAKEDIAGNOSTICSMENU)))
        (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                       (NLAML)
                                                                       (LAMA printout.SUBR)])
```

;; This would be a good one for the system to have

```
(DEFINEQ
```

(UNSHADEALLITEMS
```
  [LAMBDA (MENU)                                              (* JonL " 1-Dec-84 18:19")
    (PROG ((ITEMS (fetch (MENU ITEMS) of MENU)))
          (MAPC ITEMS (FUNCTION (LAMBDA (ITEM)
                                   (SHADEITEM ITEM MENU WHITESHADE])
```

(printout.SUBR
```
  [LAMBDA N                                                   (* JonL " 1-Dec-84 22:52")

          (* * Temporarily, this prints out to \TopLevelTtyWindow until we can also make a broadcast stream.)

    (OR (IGEQ N 2)
        (SHOULDNT "Too few args"))
    (bind X (STREAM _ (ARG N 1)) for I from 2 to N do (SELECTQ (SETQ X (ARG N I))
                                                          (T (TERPRI STREAM)
                                                             (TERPRI \TopLevelTtyWindow))
                                                          ((, -1)
                                                             (SPACES 1 STREAM)
                                                             (SPACES 1 \TopLevelTtyWindow))
```

```
                                                                    (PROGN (PRIN1 X STREAM)
                                                                           (PRIN1 X \TopLevelTtyWindow])
)

(RPAQ? DIAGNOSTICSRECORDSTREAM T)

(RPAQ? DIAGNOSTICSCONTROLWINDOW.POSITION (create POSITION XCOORD _ 800 YCOORD _ 80))

(RPAQQ EXERCISE.RUNNER NIL)

(RPAQQ EXERCISE.STATE STOP)

(RPAQQ EXERCISE.POSSIBILITIES
        ((T (EMUPROC)
            "BenchMarks")
         ((AND (HOSTNAMEP (QUOTE DSK))
               (IGEQ (DISKFREEPAGES)
                     500))
          (DSKPROC)
          "Disk Activity")
         ((START.CLEARINGHOUSE T)
          (ETHERPROC)
          "Ethernet Activity")
         (T (DAEMONPROC)
            "Swap-out WorkingSet")))

(DECLARE: EVAL@COMPILE

(RPAQQ \LD.INFOSHADE 16920)

(CONSTANTS \LD.INFOSHADE)
)

(DECLARE: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DIAGNOSTICSRECORDSTREAM DIAGNOSTICSCONTROLWINDOW.POSITION EXERCISE.RUNNER EXERCISE.STATE
       EXERCISE.POSSIBILITIES)
)

(DEFINEQ

(EXERCISE
  [LAMBDA (N)                                                      (* JonL "16-Dec-84 17:09")
    (SETQ EXERCISE.STATE NIL)
    (RESETLST
        (PROG ((OEMS ERRORMESSAGESTREAM)
               (ODRS DIAGNOSTICSRECORDSTREAM)
               (DPM.FONT MENUFONT)
               RUNNINGPROCS MENUSTATE DCWMENU DCWITEMS)
              [if (AND (EQ DIAGNOSTICSRECORDSTREAM T)
                       (HOSTNAMEP (QUOTE DSK)))
                  then (RESETSAVE (SETQ DIAGNOSTICSRECORDSTREAM (OPENSTREAM (QUOTE {DSK}DIAGNOSTICSRECORD)
                                                                           (QUOTE OUTPUT)))
                                  (QUOTE (PROGN (CLOSEF? OLDVALUE)
                                                (SETQ DIAGNOSTICSRECORDSTREAM T]
              (if (EQ ERRORMESSAGESTREAM T)
                  then (SETQ ERRORMESSAGESTREAM DIAGNOSTICSRECORDSTREAM))
              (if (WINDOWP \LD.DCW.WINDOW)
                  then (DETACHALLWINDOWS \LD.DCW.WINDOW)
                else (MAKEDIAGNOSTICSMENU))
              (OPENW \LD.DCW.WINDOW)
              [UNSHADEALLITEMS (SETQ DCWMENU (CAR (WINDOWPROP \LD.DCW.WINDOW (QUOTE MENU]
              (SHADEITEM (ASSOC (QUOTE StartExercise)
                                (SETQ DCWITEMS (fetch (MENU ITEMS) of DCWMENU)))
                         DCWMENU \LD.INFOSHADE)
              (SETQ RUNNINGPROCS (SETQ \LD.DPM.ITEMS))
              (SETQ \LD.DPM.SPACEWIDTH (STRINGWIDTH " " DPM.FONT))
              (SETQ \LD.DPM.FONTHEIGHT (FONTHEIGHT DPM.FONT))
              (bind PROC for \Possibility in (PROG1 EXERCISE.POSSIBILITIES
                                                    (* Comment PPLossage))
                as THISHEIGHT from 0 by (IPLUS \LD.DPM.FONTHEIGHT \LD.DPM.MENUBORDERSIZE)
                eachtime (if (EQ EXERCISE.STATE (QUOTE STOP))
                            then (RETURN (SETQ RUNNINGPROCS)))
                when [SETQ PROC (CAR (LISTP (NLSETQ (PROGN [NLSETQ (DEL.PROCESS (CAR (fetch (
                                                                                 EXERCISE.POSSIBILITIES
                                                                                 PROCFORM)
                                                                            of \Possibility]
                                                        (AND (EVAL (fetch (EXERCISE.POSSIBILITIES TESTFORM)
                                                                   of \Possibility))
                                                            (ADD.PROCESS (fetch (EXERCISE.POSSIBILITIES
                                                                                 PROCFORM)
                                                                          of \Possibility)
                                                                (QUOTE SUSPEND)
                                                                T]
                do (DECLARE (SPECVARS \Possibility))
                   (push \LD.DPM.ITEMS (LIST (CONCAT " " (fetch (EXERCISE.POSSIBILITIES MENUITEMFORM)
```

```
                                                                    of \Possibility))
                                        NIL "LEFT to WakeUp, MIDDLE to Suspend" (LIST (PROCESS.NAME PROC)
                                                                                      THISHEIGHT)))
                      (push RUNNINGPROCS PROC))
                (if (OR (NULL RUNNINGPROCS)
                        (EQ EXERCISE.STATE (QUOTE STOP)))
                    then (UNSHADEALLITEMS DCWMENU)
                         (RETURN (QUOTE ABORT)))
                (SETQ \LD.DPM.MENU
                 (create MENU
                         ITEMS _ \LD.DPM.ITEMS
                         MENUFONT _ DPM.FONT
                         MENUBORDERSIZE _ 1
                         MENUCOLUMNS _ 1
                         WHENSELECTEDFN _ (FUNCTION \LD.DPM.WHENSELECTED)))
                (SETQ \LD.DPM.ITEMS (fetch (MENU ITEMS) of \LD.DPM.MENU))
                                                    (* Just to be sure of the right EQality!)
                (ATTACHMENU \LD.DPM.MENU \LD.DCW.WINDOW (QUOTE TOP)
                       (QUOTE RIGHT))
                (SETQ \LD.DPM.WINDOW (OR (CAR (ATTACHEDWINDOWS \LD.DCW.WINDOW))
                                        (SHOULDNT)))
                (SETQ \LD.DPM.WINDOWBORDERSIZE (WINDOWPROP \LD.DPM.WINDOW (QUOTE BORDER)))
                (SETQ EXERCISE.RUNNER)
                (SETQ EXERCISE.STATE (QUOTE RUNNING))
                (printout.SUBR DIAGNOSTICSRECORDSTREAM T "            Legend" T T "! -> Completed !DIAGNOSE of
                       MACROTEST" T "@ -> Completed TANSPEED benchmark" T "# -> Completed BROWSE benchmark" T
                       "$ - > Found a Clearing House on the EtherNet" T "- -> Looked, but faild to find a Clearing
                       House" T "[xxx] -> Tried 32 retrievals from CH and got xxx failures" T T "{xxx} - > Copied
                       and deleted xxx copies of the Disk file" "(xxx) -> Finished with xxx'th run of the EMUPROC
                       loop." T "GDATE on new line marks release of working set pages." T T T)
                (MAPC RUNNINGPROCS (FUNCTION WAKE.PROCESS))
                (SHADEITEM (ASSOC (QUOTE StartExercise)
                                  DCWITEMS)
                       DCWMENU WHITESHADE)
            LP                                               (* Just let the processes run until we STOP them)
                (if (OR (EQ (SETQ MENUSTATE (BLOCK 10000))
                            (QUOTE STOP))
                        (EQ EXERCISE.STATE (QUOTE STOP)))
                    then (PROG ([MENU (CAR (WINDOWPROP \LD.DCW.WINDOW (QUOTE MENU]
                                ITEM)
                               (AND (SETQ ITEM (ASSOC (QUOTE StopExercise)
                                                      (fetch (MENU ITEMS) of MENU)))
                                    (SHADEITEM ITEM MENU \LD.INFOSHADE))
                               (\LD.STOPPROCS RUNNINGPROCS MENU)
                               (DETACHALLWINDOWS \LD.DCW.WINDOW)
                               (AND (NEQ DIAGNOSTICSRECORDSTREAM T)
                                    (CLOSEF DIAGNOSTICSRECORDSTREAM))
                               (AND ITEM (SHADEITEM ITEM MENU WHITESHADE)))
                         (SETQ ERRORMESSAGESTREAM OEMS)
                         (SETQ DIAGNOSTICSRECORDSTREAM ODRS)
                         (RETURN (QUOTE STOP))
                  elseif MENUSTATE
                    then                                     (* Someday, look for more interesting signals))
                (GO LP)))])


(\LD.BLOCKCHECK
  [LAMBDA (TIME POS NAME)                                    (* JonL "16-Dec-84 17:10")
    (PROG ((SIGNAL (if (DISMISS TIME)
                     elseif (NULL TIME)
                       then

                  (* Allow two passes around the scheduler loop when blocking for 0 time;
                  this helps the MOUSE tracker)

                            (BLOCK)))
            NEWP OLDP)
           (if (OR (EQ SIGNAL (QUOTE STOP))
                   (EQ EXERCISE.STATE (QUOTE STOP)))
               then (RETFROM POS (QUOTE STOP))
             elseif SIGNAL
               then                                          (* Do some action precipitaed by the menu))
           (if (EQ NAME (QUOTE DON'T))
               then (RETURN)
             elseif (AND NAME (LITATOM NAME))
             elseif (AND POS (LITATOM POS))
               then (SETQ NAME POS)
             else (SHOULDNT))
           (if [SETQ NEWP (find ITEM in \LD.DPM.ITEMS suchthat (EQ NAME (CAR (CADDR ITEM]
               then (if [AND EXERCISE.RUNNER (SETQ OLDP (find ITEM in \LD.DPM.ITEMS
                                                          suchthat (EQ EXERCISE.RUNNER (CAR (CADDR ITEM]
                        then                                 (* First, take away the "baton" from the old process)
                            (BITBLT NIL NIL NIL \LD.DPM.WINDOW \LD.DPM.WINDOWBORDERSIZE (CADR (CADDDR OLDP))
                                   \LD.DPM.SPACEWIDTH \LD.DPM.FONTHEIGHT (QUOTE TEXTURE)
                                   (QUOTE REPLACE)
                                   WHITESHADE))
                      (BITBLT NIL 0 0 \LD.DPM.WINDOW \LD.DPM.WINDOWBORDERSIZE (CADR (CADDDR NEWP))
```

```
                             \LD.DPM.SPACEWIDTH \LD.DPM.FONTHEIGHT (QUOTE TEXTURE)
                             (QUOTE REPLACE)
                             BLACKSHADE)

          (* Here, we give the "baton" to the process NAME -- by turning on some bits in the menuitem reflecting that process)

                    (SETQ EXERCISE.RUNNER NAME])
```

(**LD.STOPPROCS**
```
  [LAMBDA (PROCS MENU)                                             (* JonL " 1-Dec-84 21:13")

          (* When we want to stop, the find out who is still alive, and give them the STOP signal;
          then wait around for 60 seconds so to see if they "give up" gracefully.)

    [MAPC [SETQ PROCS (MAPCONC (MKLIST PROCS)
                               (FUNCTION (LAMBDA (PROC)
                                           (AND (PROCESSP PROC)
                                                (LIST PROC]
          (FUNCTION (LAMBDA (PROC)
                      (WAKE.PROCESS PROC (QUOTE STOP]
    [bind STOPPERS PN forDuration 60 timerUnits (QUOTE SECONDS) until (NULL PROCS) eachtime (BLOCK)
       when (SETQ STOPPERS (for P in PROCS when (PROCESS.FINISHEDP P) collect P))
       do (SETQ PROCS (LDIFFERENCE PROCS STOPPERS))
          (for P ITEM in STOPPERS do (SETQ PN (PROCESS.NAME P))
                                     (if [SETQ ITEM (find ITEM in \LD.DPM.ITEMS
                                                       suchthat (EQ PN (CAR (CADDDR ITEM]
                                         then (SHADEITEM ITEM MENU DARKBITSHADE]
    (if PROCS
        then (printout.SUBR DIAGNOSTICSRECORDSTREAM T "Processes which didn't terminate normally: "
               (MAPCAR PROCS (QUOTE PROCESS.NAME))
               T)
          (MAPC PROCS (FUNCTION DEL.PROCESS)))
)
```

;; Some user-interface, menu-like things

(DECLARE: EVAL@COMPILE

(RPAQQ **LD.DPM.MENUBORDERSIZE** 3)

(CONSTANTS (\LD.DPM.MENUBORDERSIZE 3))
)

(RPAQ? **LD.DCW.WINDOW** NIL)

(RPAQ? **LD.DPM.MENU** NIL)

(RPAQ? **LD.DPM.WINDOW** NIL)

(RPAQ? **LD.DPM.WINDOWBORDERSIZE** NIL)

(RPAQ? **LD.DPM.ITEMS** NIL)

(RPAQ? **LD.DPM.SPACEWIDTH** NIL)

(DEFINEQ

(**MAKEDIAGNOSTICSMENU**
```
  [LAMBDA NIL                                                      (* JonL "16-Dec-84 17:06")
    (AND (WINDOWP \LD.DCW.WINDOW)
         (CLOSEW \LD.DCW.WINDOW))
    (SETQ \LD.DCW.WINDOW (ADDMENU (create MENU
                                    ITEMS _ (QUOTE ((StartExercise [PROG ((\INTERRUPTABLE NIL))
                                                                     (SETQ EXERCISE.STATE (QUOTE RUN))
                                                                     (ADD.PROCESS (QUOTE (EXERCISE]
                                                   "Begins diagnostic suite processes")
                                                   (StopExercise (WAKE.PROCESS (QUOTE EXERCISE)
                                                                   (SETQ EXERCISE.STATE (QUOTE STOP))
                                                                   )
                                                   "Stops and deletes the diagnostic processes (if
                                                   any)")))
                                    WHENSELECTEDFN _ (FUNCTION \LD.DCW.WHENSELECTED)
                                    MENUFONT _ (FONTCREATE (QUOTE HELVETICA)
                                                 18)
                                    MENUTITLEFONT _ (FONTCREATE (QUOTE TIMESROMAN)
                                                      12)
                                    TITLE _ "Diagnostics Control"
                                    CENTERFLG _ T
                                    ITEMHEIGHT _ 30
                                    ITEMWIDTH _ [IMAX 150 (IPLUS 20 (STRINGWIDTH "StartExercise"
                                                                       (FONTCREATE (QUOTE HELVETICA)
                                                                         18]
                                    MENUBORDERSIZE _ \LD.DPM.MENUBORDERSIZE)
                             NIL DIAGNOSTICSCONTROLWINDOW.POSITION))
    [WINDOWPROP \LD.DCW.WINDOW (QUOTE AFTERMOVEFN)
```

```
                   (FUNCTION (LAMBDA (WINDOW)
                               (OR (EQ WINDOW \LD.DCW.WINDOW)
                                   (SHOULDNT))
                             (PROG [[REG (WINDOWPROP WINDOW (QUOTE REGION]
                                    (SETQ DIAGNOSTICSCONTROLWINDOW.POSITION (create POSITION
                                                            XCOORD _ (fetch (REGION LEFT)
                                                                        of REG)
                                                            YCOORD _ (fetch (REGION BOTTOM)
                                                                        of REG]
          (WINDOWPROP \LD.DCW.WINDOW (QUOTE RESHAPEFN)
                 (QUOTE DON'T))
          [WINDOWPROP \LD.DCW.WINDOW (QUOTE CLOSEFN)
                 (FUNCTION (LAMBDA (WINDOW)
                             (SETQ \LD.DCW.WINDOW]
          \LD.DCW.WINDOW])
```

(**LD.DCW.WHENSELECTED**
```
  [LAMBDA (ITEM MENU BUTTON)                                        (* JonL " 1-Dec-84 18:29")
    (SHADEITEM ITEM MENU GRAYSHADE)
    (EVAL (CADR ITEM])
```

(**LD.DPM.WHENSELECTED**
```
  [LAMBDA (ITEM MENU BUTTON)                                        (* JonL " 1-Dec-84 00:08")
    (PROG ((STUFF (CADDDR ITEM)))
```

          (* This STUFF in the 4th slot of the menuitem should be a list of the process name and the height in the menuwindow of this
          item.)

```
          (APPLY* (SELECTQ BUTTON
                      (LEFT (FUNCTION WAKE.PROCESS))
                      (MIDDLE (FUNCTION SUSPEND.PROCESS))
                      (RETURN))
                  (OR (FIND.PROCESS (CAR STUFF))
                      (RETURN])
)

(DECLARE: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \LD.DCW.WINDOW \LD.DPM.MENU \LD.DPM.WINDOW \LD.DPM.WINDOWBORDERSIZE \LD.DPM.FONTHEIGHT \LD.DPM.ITEMS
      \LD.DPM.SPACEWIDTH)
)

(DECLARE: EVAL@COMPILE DONTCOPY

(DECLARE: EVAL@COMPILE

(RECORD EXERCISE.POSSIBILITIES (TESTFORM PROCFORM MENUITEMFORM))
)

(DECLARE: EVAL@COMPILE

(PUTPROPS ldprintout MACRO ((STREAM . REST)
                            (PROGN                                (* How we'd like a broadcast stream here!)
                                (printout STREAM . REST)
                                (printout \TopLevelTtyWindow . REST))))
)
)
```

;; Various background activities to stress hardware

```
(DEFINEQ
```

(**DSKPROC**
```
  [LAMBDA (N)                                                       (* JonL "16-Dec-84 18:27")
                                                                   (* Cause a lot of DSK activity)

    (RESETLST
        [RESETSAVE NIL (QUOTE (PROGN (CLOSEF? (QUOTE {DSK}RANDOMDATA))
                                     (CLOSEF? (QUOTE {CORE}RANDOMDATA))
                                     [until (NULL (DELFILE (QUOTE {DSK}RANDOMDATA]
                                     [until (NULL (DELFILE (QUOTE {CORE}RANDOMDATA]
                                     (RESETLST
                                         (PROG1 (* Comment PPLossage))
                                         (RESETSAVE (CNDIR (QUOTE {DSK}))
                                                (LIST (FUNCTION CNDIR)
                                                        (DIRECTORYNAME T)))
                                         ([LAMBDA (X)
                                                (MAPC X (FUNCTION CLOSEF?))
                                                (MAPC X (FUNCTION DELFILE]
                                           (DIRECTORY (QUOTE TEMPRANDOMCOPY*)))))]
          (DSKPROC.AUX N))])
```

(**DSKPROC.AUX**
```
  [LAMBDA (N)                                                       (* JonL "16-Dec-84 18:44")
```

```
        (OR (FIXP N)
            (SETQ N MAX.SMALLP))
        (PROG (SOURCE CORESOURCE NFILES)
                [until (NULL (DELFILE (QUOTE {DSK}RANDOMDATA]
                [until (NULL (DELFILE (QUOTE {CORE}RANDOMDATA]
                (SETQ CORESOURCE (OPENFILE (QUOTE {CORE}RANDOMDATA)
                                       (QUOTE OUTPUT)))              (* First, create a moderately large file to copy)
                 (for I from 0 to 99 do (\LD.BLOCKCHECK 100 (QUOTE DSKPROC.AUX)
                                                (QUOTE DSKPROC))
                                    (for J from 0 to 9 do (printout CORESOURCE "Now here's yet another line, the "
                                                            (IPLUS (ITIMES I 10)
                                                                   J 1)
                                                          "'th one." T)))
                (TERPRI CORESOURCE)
                (CLOSEF CORESOURCE)
                (SETQ SOURCE (DSKPROC.DO1COPY CORESOURCE (QUOTE {DSK}RANDOMDATA)
                              CORESOURCE))
                (printout.SUBR DIAGNOSTICSRECORDSTREAM T "Finished initializing the source file for DSKPROC" T)
                (\LD.BLOCKCHECK 250 (QUOTE DSKPROC.AUX)
                        (QUOTE DSKPROC))                            (* Then make a random number of copies, and then delete them)
                (to N do [MAPC (bind SS DS to (SETQ NFILES (RAND 3 7))
                               collect (\LD.BLOCKCHECK (ITIMES 1000 (RAND 1 5))
                                            (QUOTE DSKPROC.AUX)
                                            (QUOTE DSKPROC))
                                        (DSKPROC.DO1COPY SOURCE (PACK* (QUOTE {DSK}TEMPRANDOMCOPY)
                                                            (SELECTC (RAND 1 5)
                                                                (1 "25.TXT")
                                                                (2 ".TXT")
                                                                (3 "MUMBLE")
                                                                (4 "2345.MUMBLE")
                                                                (5 "")
                                                              (SHOULDNT)))
                                          CORESOURCE))
                                (FUNCTION (LAMBDA (FILE)
                                        (\LD.BLOCKCHECK 250 (QUOTE DSKPROC.AUX)
                                                (QUOTE DSKPROC))
                                        (DELFILE FILE]
                    (printout.SUBR DIAGNOSTICSRECORDSTREAM (QUOTE {)
                        NFILES
                        (QUOTE }])
```

(**DSKPROC.DO1COPY**
```
  [LAMBDA (SOURCE DESTINATION CORESOURCE)                          (* JonL "19-Dec-84 18:53")
    (LET ((SS (OPENSTREAM SOURCE (QUOTE INPUT)))
          (DS (OPENSTREAM DESTINATION (QUOTE OUTPUT)))
          (FILELEN 0)
          THISROUND)
        (SETQ FILELEN (GETFILEINFO SS (QUOTE LENGTH)))
        (bind (NBYTES _ FILELEN) while (IGREATERP NBYTES 0) do (\LD.BLOCKCHECK 250 (QUOTE DSKPROC.AUX)
                                                                    (QUOTE DSKPROC))
                                                               (to (SETQ THISROUND (IMIN NBYTES 512))
                                                                 do (BOUT DS (BIN SS)))
                                                               (add NBYTES (IMINUS THISROUND)))
        (CLOSEF DS)
        (\LD.BLOCKCHECK 250 (QUOTE DSKPROC.AUX)
               (QUOTE DSKPROC))
        (if CORESOURCE
            then (CLOSEF SS)
                 (SETQ SS (OPENSTREAM CORESOURCE (QUOTE INPUT)))
          else (SETFILEPTR SS 0))
        (SETQ DS (OPENSTREAM DS (QUOTE INPUT)))                    (* Now compare the file to see that it "made it")
        [if (NEQ FILELEN (GETFILEINFO DS (QUOTE LENGTH)))
            then (!MRAID DESTINATION "DSK copy failure -- wrong length")
          else (bind (NBYTES _ FILELEN)
                     (I _ -1) while (IGREATERP NBYTES 0) do (\LD.BLOCKCHECK 250 (QUOTE DSKPROC.AUX)
                                                                (QUOTE DSKPROC))
                                                           (to (SETQ THISROUND (IMIN NBYTES 512))
                                                             eachtime (PROG1 (add I 1)
                                                                 (* Comment PPLossage)
                                                                  )
                                                             when (NEQ (BIN SS)
                                                                       (BIN DS))
                                                             do (!MRAID DESTINATION (CONCAT "DSK copy failure
                                                                              -- data different at
                                                                              filepos " I))
                                                                (RETURN (SETQ THISROUND)))
                                                           (add NBYTES (IMINUS (OR THISROUND (RETURN]
        (CLOSEF SS)
        (CLOSEF DS)
        (\LD.BLOCKCHECK 500 (QUOTE DSKPROC.AUX)
               (QUOTE DSKPROC))
        (FULLNAME DS])
```

(**ETHERPROC**
```
  [LAMBDA (N)                                                      (* JonL " 9-Dec-84 13:45")
```

```
    (DECLARE (GLOBALVARS LOCAL.CLEARINGHOUSE))
    (OR (FIXP N)
        (SETQ N MAX.SMALLP))
    (for I to N bind (J _ 0)
                    (JF _ 0)
       do [if (EVENP I 64)
             then (\LD.BLOCKCHECK 1000 (QUOTE ETHERPROC))
                  (SETQ LOCAL.CLEARINGHOUSE)
                  (printout.SUBR DIAGNOSTICSRECORDSTREAM (if [CAR (LISTP (NLSETQ (START.CLEARINGHOUSE T]
                                                              then (QUOTE $)
                                                            else (QUOTE -]
           (if LOCAL.CLEARINGHOUSE
              then (\LD.BLOCKCHECK (if (EVENP I 8)
                                      then 1000
                                    else NIL)
                        (QUOTE ETHERPROC))
                   (if [CAR (LISTP (NLSETQ (CH.LIST.ORGANIZATIONS]
                      else (add JF 1))
                   (if (IGEQ (add J 1)
                             32)
                      then (SETQ J 1)
                           (printout.SUBR DIAGNOSTICSRECORDSTREAM (QUOTE %[)
                               JF
                               (QUOTE %]))
                           (SETQ JF 0))
            else (\LD.BLOCKCHECK 20000 (QUOTE ETHERPROC))
                 (SETQ LOCAL.CLEARINGHOUSE)
                 (NLSETQ (START.CLEARINGHOUSE T]))
```

(**DAEMONPROC**
```
  [LAMBDA (N)                                                           (* JonL " 7-Dec-84 01:37")
    (OR (FIXP N)
        (SETQ N MAX.SMALLP))
    (to N do (to (CONSTANT (QUOTIENT (TIMES 10 60)
                                    10))
                do (\LD.BLOCKCHECK 10000 (QUOTE DAEMONPROC)
                        (QUOTE DON'T)))
```

          (* * Random perturbation of the working set every 10 minutes +or-
          a few seconds)

```
             (\LD.BLOCKCHECK (ITIMES 1000 (RAND 5 10))
                   (QUOTE DAEMONPROC))
             (\RELEASEWORKINGSET)
             (printout.SUBR DIAGNOSTICSRECORDSTREAM T (GDATE)
                  T])
)
```

;; Various diagnostic and benchmark activities

(DEFINEQ

(**EMUPROC**
```
  [LAMBDA (N)                                                           (* JonL " 7-Dec-84 01:23")
                                                                        (* Basically, just runs a lot of test of emulator instructons)
    (OR (FIXP N)
        (SETQ N MAX.SMALLP))                                            (* First, run the standard MACROTEST diagnostics)
    (for RUN# to N
       do (if (AND (GETD (QUOTE !NUMBERTEST))
                   (GETD (QUOTE CHECKFREELISTS)))
             then (for TEST in (QUOTE (!NUMBERTEST !FNUMTEST !MIXNUMTEST !GCTEST !CONSTEST !FVARTEST !INTERPTEST
                                       CHECKCONSPAGES CHECKFREELISTS 20RECLAIM CHECKFREELISTS))
                     do (\LD.BLOCKCHECK 500 (QUOTE EMUPROC))
                        (APPLY* TEST))
                  (printout.SUBR DIAGNOSTICSRECORDSTREAM (QUOTE !)))
          (\LD.BLOCKCHECK (ITIMES 1000 (RAND 1 5))
               (QUOTE EMUPROC))
          (\LD.TANSPEED)
          (\LD.BLOCKCHECK (ITIMES 1000 (RAND 1 5))
               (QUOTE EMUPROC))
          (\LD.BROWSE)
          (\LD.BLOCKCHECK 1000 (QUOTE EMUPROC))
          (printout.SUBR DIAGNOSTICSRECORDSTREAM (QUOTE %()
               RUN#
               (QUOTE %)])
```

(**20RECLAIM**
```
  [LAMBDA NIL                                                           (* JonL "28-Nov-84 13:52")
    (FRPTQ 20 (RECLAIM])
)
```

;; After the TANSPEED benchmark

```
(DEFINEQ

(\LD.TANSPEED
  [LAMBDA NIL                                                    (* JonL " 7-Dec-84 01:22")
                                                                 (* TANSPEED benchmark is a fairly good test of floating point
                                                                 arithmetic also)

    (for I F (A _ 1.0) from 0 to 2498 do (AND (EVENP I 32)
                                                  (\LD.BLOCKCHECK 500 (QUOTE EMUPROC)))
                                             (SETQ A (FPLUS (TAN (ARCTAN [ANTILOG (LOG (SQRT (FTIMES A A]
                                                                                          T)
                                                              T)
                                                           1.0))
       finally (if [LESSP 25.0 (ABS (SETQ F (DIFFERENCE A 2500.0]
                    then (!MRAID (LIST (QUOTE (TANSPEED))
                                        (QUOTE =>)
                                        F
                                        (QUOTE should-have-been)
                                        2476.246))
                 else (printout.SUBR DIAGNOSTICSRECORDSTREAM (QUOTE @])

)


;; Extraction from Gabriel's BROWSE benchmark

(DEFINEQ

(\LD.BROWSE
  [LAMBDA NIL                                                    (* JonL " 7-Dec-84 01:23")

         (* Unfortunately, this has to be a copy of the code in BROWSE since we want to do the init phase only once)

    [if !BROWSEINIT
        else (SETQ !BROWSEINIT
                   (\LD.BROWSEINIT 100 10 4
                        (QUOTE ((A A A B B B B A A A A A B B A A A)
                                (A A B B B B A A (A A)
                                  (B B))
                                (A A A B (B A)
                                  B A B A]
    [for UNITS
       on (bind A N (RAND _ 21)
                 (L _ !BROWSEINIT) while L
              do (\LD.BLOCKCHECK NIL (QUOTE EMUPROC))
                 [if [EQ 0 (SETQ N (IMOD (SETQ RAND (IMOD (ITIMES RAND 17)
                                                             251))
                                           (LENGTH L]
                     then (push A (pop L))
                     else (for N from N to 2 by -1 as X on L do NIL finally (PROGN (push A (CADR X))
                                                                                    (RPLACD X (CDDR X]
              finally (RETURN A))
       do (for PATS on (QUOTE ((*A ?B *B ?B A *A A *B *A)
                               (*A *B *B *A (*A)
                                 (*B))
                               (? ? * (B A)
                                 * ? ?)))
             do (for P on (GETP (CAR UNITS)
                                 (QUOTE PATTERN))
                   do (\LD.BLOCKCHECK NIL (QUOTE EMUPROC))
                      (\LD.BROWSEMATCH (CAR PATS)
                          (CAR P)
                          NIL]
    (printout.SUBR DIAGNOSTICSRECORDSTREAM (QUOTE #])


(\LD.BROWSEINIT
  [LAMBDA (N M NPATS IPATS)                                      (* JonL "30-Nov-84 22:50")
    (SETQ IPATS (SUBST NIL NIL IPATS))
    (bind (A _ NIL)
          (LOSER _ (LAST IPATS)) first (RPLACD LOSER IPATS) for old N from N to 1 by -1
       as (I _ M) by (if (ZEROP I)
                          then M
                          else (SUB1 I))
       as (NAME _ (GENSYM)) by (GENSYM)
       do (\LD.BLOCKCHECK NIL (QUOTE EMUPROC))
          (push A NAME)
          (RPTQ I (PUTPROP NAME (GENSYM)
                           NIL))
          (PUTPROP NAME (QUOTE PATTERN)
                 (bind (A _ NIL) for I from NPATS to 1 by -1 as IPATS on IPATS do (push A (CAR IPATS))
                    finally (RETURN A)))
          (RPTQ (DIFFERENCE M I)
                (PUTPROP NAME (GENSYM)
                       NIL))
       finally (PROGN                                            (* Just to break the circularity)
                      (RPLACD LOSER NIL)
                      (RETURN A])
```

```
(\LD.BROWSEMATCH
   [LAMBDA (PAT DAT ALIST)                                    (* JonL "25-FEB-83 13:38")
     (COND
        ((NULL PAT)
         (NULL DAT))
        ((NULL DAT)
         NIL)
        ((OR (EQ (CAR PAT)
                 (QUOTE ?))
             (EQ (CAR PAT)
                 (CAR DAT)))
         (\LD.BROWSEMATCH (CDR PAT)
                (CDR DAT)
                ALIST))
        ((EQ (CAR PAT)
             (QUOTE *))
         (OR (\LD.BROWSEMATCH (CDR PAT)
                DAT ALIST)
             (\LD.BROWSEMATCH (CDR PAT)
                 (CDR DAT)
                 ALIST)
             (\LD.BROWSEMATCH PAT (CDR DAT)
                 ALIST)))
        (T (COND
             [(NLISTP (CAR PAT))
              (COND
                [(EQ (CHAR1 (CAR PAT))
                     (QUOTE ?))
                 (PROG ((VAL (FASSOC (CAR PAT)
                                 ALIST)))
                       (RETURN (COND
                                 (VAL (\LD.BROWSEMATCH (CONS (CDR VAL)
                                                          (CDR PAT))
                                            DAT ALIST))
                                 (T (\LD.BROWSEMATCH (CDR PAT)
                                          (CDR DAT)
                                          (CONS (CONS (CAR PAT)
                                                    (CAR DAT))
                                              ALIST]
                ((EQ (CHAR1 (CAR PAT))
                     (QUOTE *))
                 (PROG ((VAL (FASSOC (CAR PAT)
                                 ALIST)))
                       (RETURN (COND
                                 (VAL (\LD.BROWSEMATCH (APPEND (CDR VAL)
                                                           (CDR PAT))
                                            DAT ALIST))
                                 (T (for (L _ NIL) by (NCONC L (LIST (CAR D))) as E
                                        on (CONS NIL DAT) as (D _ DAT) by (CDR D)
                                        do (COND
                                             ((\LD.BROWSEMATCH (CDR PAT)
                                                     D
                                                 (CONS (CONS (CAR PAT)
                                                           L)
                                                     ALIST))
                                              (RETURN T]
             (T (AND (NOT (NLISTP (CAR DAT)))
                     (\LD.BROWSEMATCH (CAR PAT)
                           (CAR DAT)
                           ALIST)
                     (\LD.BROWSEMATCH (CDR PAT)
                           (CDR DAT)
                           ALIST])
)

(RPAQQ !BROWSEINIT NIL)

(DECLARE: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS !BROWSEINIT)
)

(DECLARE: EVAL@COMPILE DONTCOPY

(DECLARE: EVAL@COMPILE

(PUTPROPS CHAR1 MACRO ((X)
                        (NTHCHAR X 1)))
)
)

(DECLARE: DONTEVAL@COMPILE DONTEVAL@LOAD DOCOPY

(ADDTOVAR DISPLAYFONTEXTENSIONS STRIKE)
```

```
(ADDTOVAR DISPLAYFONTDIRECTORIES {FLOPPY})

(ADDTOVAR LISPUSERSDIRECTORIES {FLOPPY})

(RPAQQ !MTUSERAIDFLG NIL)

(FILESLOAD (COMPILED FROM VALUEOF LISPUSERSDIRECTORIES)
       PAGEHOLD MACROTESTAUX MACROTEST PLURAL)

(MAKEDIAGNOSTICSMENU)
)

(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA printout.SUBR)
)

(PUTPROPS LISPDIAGNOSTICS COPYRIGHT ("Xerox Corporation" 1984))
```

FUNCTION INDEX

VARIABLE INDEX

MACRO INDEX

CONSTANT INDEX

RECORD INDEX