File created: 22-May-90 10:55:20 **{DSK}/home/neptune/jds/TCPOPS.;17**

changes to: (FNS TCP-ACCEPT TCP-LISTEN UDP-RECV)

previous date: 3-May-90 11:40:39 {DSK}/home/neptune/jds/TCPOPS.;16

Read Table: XCL

Package: INTERLISP

Format: XCCS


```
(RPAQQ TCPOPSCOMS
       ((FILES CHARDEVICE)
        (ADDVARS (\\INITSUBRS (TCP 144)))
        (COMS ;; TCP Streams
              (FNS \\TCP-DEV-INIT \\TCP-OPENFILE \\TCP-FORCEOUTPUT \\TCP-GETNEXTBUFFER \\TCP-EOFP
                   \\TCP-CLOSEFILE \\TCP-EVENTFN \\TCP.BUFFERED.BOUTS)
              (P (\\TCP-DEV-INIT)))
        (COMS ;; User-level TCP operations
              (FNS TCP OPENTCPSTREAM TCP-ACCEPT TCP-LISTEN TCP-CLOSE)
              (FNS UDP-LISTEN UDP-SEND UDP-RECV)
              (FNS GETHOSTFROMNAME GETHOSTFROMADDR GETHOSTFROMSOCKET GETHOSTNAME))
        (DECLARE\: EVAL@LOAD DONTCOPY (COMS ;; Debugging functions &c
                                            (VARS (BUFFER (\\ALLOCBLOCK 100)))
                                            (FNS TCPRECV TCPSEND SEEBUFFER FOON)))))

(FILESLOAD CHARDEVICE)

(ADDTOVAR \\INITSUBRS (TCP 144))

;; TCP Streams

(DEFINEQ
```

(\\TCP-DEV-INIT

```
(LAMBDA NIL                                                    ; Edited 20-Feb-90 12:51 by jds
    ;; Initialization for buffered Unix-character-oriented device (e.g. for TCP streams on SUN)
    (SETQ \\TCP-FDEV (|create| FDEV
                             DEVICENAME _ "TCP"
                             FDBINABLE _ T
                             FDBOUTABLE _ T
                             BUFFERED _ T
                             BIN _ (FUNCTION \\BUFFERED.BIN)
                             BOUT _ (FUNCTION \\BUFFCHAR-OTHER-BOUT)
                             OPENFILE _ (FUNCTION \\BUFFCHAR-DEV-OPENFILE)
                             EVENTFN _ (FUNCTION \\CHAR-DEV-EVENTFN)
                             REOPENFILE _ (FUNCTION \\BUFFCHAR-DEV-OPENFILE)
                             CLOSEFILE _ (FUNCTION \\TCP-CLOSEFILE)
                             FORCEOUTPUT _ (FUNCTION \\TCP-FORCEOUTPUT)
                             EOFP _ (FUNCTION \\TCP-EOFP)
                             BLOCKIN _ (FUNCTION \\BUFFERED.BINS)
                             BLOCKOUT _ (FUNCTION \\TCP.BUFFERED.BOUTS)
                             READP _ (FUNCTION \\GENERIC.READP)
                             PEEKBIN _ (FUNCTION \\BUFFERED.PEEKBIN)
                             GETNEXTBUFFER _ (FUNCTION \\TCP-GETNEXTBUFFER)))
    (\\DEFINEDEVICE 'TCP \\TCP-FDEV)))
```

(\\TCP-OPENFILE

```
(LAMBDA (NAME ACCESS RECOG OTHERINFO FDEV)                     ; Edited  7-Mar-90 10:11 by jds
    (LET ((UNIX-NAME (SUBSTRING NAME (ADD1 (STRPOS "}" NAME))))
          (ERRNO (CREATECELL \\FIXP))
          IODESCRIPTOR ACCESS-VALUE STREAM OTHER-STREAM)
         (SETQ STREAM
               (|create| STREAM
                       BINABLE _ T
                       BOUTABLE _ NIL
                       DEVICE _ FDEV
                       FULLFILENAME _ NAME
                       USERCLOSEABLE _ T
                       USERVISIBLE _ T
                       EOLCONVENTION _ LF.EOLC))
         (SELECTQ ACCESS
              (INPUT (|replace| (STREAM STRMBINFN) |of| STREAM |with| (FUNCTION \\BUFFERED.BIN))
                     (SETQ ACCESS-VALUE 0))
              (OUTPUT (|replace| (STREAM STRMBOUTFN) |of| STREAM |with| (FUNCTION \\BUFFCHAR-OTHER-BOUT))
                      (REPLACE F2 OF STREAM
                         WITH (SETQ OTHER-STREAM
                                    (|create| STREAM
                                            BINABLE _ NIL
```

```
                                    BOUTABLE _ NIL
                                    DEVICE _ FDEV
                                    FULLFILENAME _ NAME
                                    STRMBOUTFN _ (FUNCTION \\BUFFERED.BOUT)
                                    USERCLOSEABLE _ T
                                    USERVISIBLE _ T
                                    EOLCONVENTION _ LF.EOLC)))
                      (SETQ ACCESS-VALUE 1))
                (BOTH (|replace| (STREAM STRMBINFN) |of| STREAM |with| (FUNCTION \\BUFFERED.BIN))
                      (|replace| (STREAM STRMBOUTFN) |of| STREAM |with| (FUNCTION \\BUFFCHAR-OTHER-BOUT))
                      (REPLACE F2 OF STREAM
                         WITH (SETQ OTHER-STREAM
                                    (|create| STREAM
                                           BINABLE _ NIL
                                           BOUTABLE _ NIL
                                           DEVICE _ FDEV
                                           FULLFILENAME _ NAME
                                           STRMBOUTFN _ (FUNCTION \\BUFFERED.BOUT)
                                           USERCLOSEABLE _ T
                                           USERVISIBLE _ T
                                           EOLCONVENTION _ LF.EOLC)))
                            (SETQ ACCESS-VALUE 2))
                (APPEND (\\ILLEGAL.ARG ACCESS))
                (\\ILLEGAL.ARG ACCESS))
          (COND
             ((SETQ IODESCRIPTOR (SUBRCALL CHAR-OPENFILE UNIX-NAME ACCESS-VALUE ERRNO))
              ;; Open happened, so put things together.
              (|replace| (STREAM F1) |of| STREAM |with| IODESCRIPTOR)
              (COND
                 (OTHER-STREAM (|replace| (STREAM F1) |of| OTHER-STREAM |with| IODESCRIPTOR))))
             (T (\\CHAR-ERROR ERRNO NAME)))
          STREAM)))
```

(**TCP-FORCEOUTPUT**

```
(LAMBDA (STREAM WAIT)                                    ; Edited 15-Dec-89 16:09 by jds
```

;;; Generic buffer refiller for Buffered character streams (e.g. TCP streams on Sun, or the Lisp side of a shell CHAT, eventually)

```
(PROG (ERRCODE (OTHER-STREAM (|fetch| F2 |of| STREAM))
               (ERRNO (\\CREATECELL \\FIXP))
               BUFFER)
      (COND
         ((NULL (|fetch| CPPTR |of| OTHER-STREAM))
          ;; No buffer allocated yet; create one.
          (REPLACE CPPTR OF OTHER-STREAM WITH (NCREATE 'VMEMPAGEP))
          (REPLACE CBUFSIZE OF OTHER-STREAM WITH 512)
          (REPLACE CBUFMAXSIZE OF OTHER-STREAM WITH 512)
          (REPLACE COFFSET OF OTHER-STREAM WITH 0)
          T)
         ((ZEROP (|fetch| COFFSET |of| OTHER-STREAM))
          T)
         ((SETQ ERRCODE (\\CHAR-BOUTS OTHER-STREAM (|fetch| CPPTR |of| OTHER-STREAM)
                                0
                                (|fetch| COFFSET |of| OTHER-STREAM)
                                NIL))
          ;; WRITE HAPPENED.
          (|replace| CBUFSIZE |of| OTHER-STREAM |with| 512)
          (|replace| CBUFMAXSIZE |of| OTHER-STREAM |with| 512)
          (|replace| COFFSET |of| OTHER-STREAM |with| 0)
          T)))))
```

(**TCP-GETNEXTBUFFER**

```
(LAMBDA (STREAM WHATFOR NOERRORFLG EOF-TEST)             ; Edited 20-Feb-90 12:43 by jds
```

;;; Generic buffer refiller for Buffered character streams (e.g. TCP streams on Sun, or the Lisp side of a shell CHAT, eventually).

```
(PROG (ERRCODE (ERRNO (\\CREATECELL \\FIXP))
               BUFFER)
  READ-LOOP
      (RETURN (SELECTQ WHATFOR
                   (READ ;; READING; GET A FRESH BUFFER FULL OF UN-READ CHARACTERS.
                         (SETQ BUFFER (OR (FETCH (STREAM CPPTR) OF STREAM)
                                          (NCREATE 'VMEMPAGEP)))
                         (|replace| CPPTR |of| STREAM |with| BUFFER)
                         (COND
                            ((ZEROP (SETQ ERRCODE (TCP 6 (|fetch| (STREAM F1) |of| STREAM)
                                                       BUFFER 512)))
                             (AND (NULL NOERRORFLG)
                                  (\\EOF.ACTION STREAM))
                             NIL)
                            ((EQ ERRCODE T)
```

```
                                (AND EOF-TEST (RETURN T))
                                (BLOCK)
                                (GO READ-LOOP))
                               (ERRCODE
                                      ;; Read succeeded, and ERRCODE has # of chars read.
                                      (|replace| CPPTR |of| STREAM |with| BUFFER)
                                      (|replace| COFFSET |of| STREAM |with| 0)
                                      (|replace| CBUFSIZE |of| STREAM |with| ERRCODE)
                                      (|replace| CBUFMAXSIZE |of| STREAM |with| ERRCODE)
                                      T)
                               ((NULL NOERRORFLG)
                                (\\CHAR-ERROR ERRNO STREAM))))
                        (WRITE (COND
                                ((NULL (FETCH CPPTR OF STREAM))
                                 ;; No buffer allocated yet; create one.
                                 (REPLACE CPPTR OF STREAM WITH (NCREATE 'VMEMPAGEP))
                                 (REPLACE CBUFSIZE OF STREAM WITH 512)
                                 (REPLACE CBUFMAXSIZE OF STREAM WITH 512)
                                 (REPLACE COFFSET OF STREAM WITH 0)
                                 T)
                                ((ZEROP (FETCH COFFSET OF STREAM))
                                 T)
                                ((SETQ ERRCODE (\\CHAR-BOUTS STREAM (FETCH CPPTR OF STREAM)
                                                             0
                                                             (FETCH COFFSET OF STREAM)
                                                             NOERRORFLG))
                                 ;; WRITE HAPPENED.
                                 (REPLACE CBUFSIZE OF STREAM WITH 512)
                                 (REPLACE CBUFMAXSIZE OF STREAM WITH 512)
                                 (REPLACE COFFSET OF STREAM WITH 0)
                                 T)))
                        (SHOULDNT))))))
```

(**TCP-EOFP**

```
  (LAMBDA (STREAM)                                          ; Edited 20-Feb-90 12:42 by jds
    ;; T if there will be no more data on the stream
    (AND (OR (NOT (|fetch| (STREAM CPPTR) |of| STREAM))
             (IEQP (FETCH (STREAM COFFSET) OF STREAM)
                   (FETCH (STREAM CBUFSIZE) OF STREAM)))
         (NOT (\\TCP-GETNEXTBUFFER STREAM 'READ T T)))))
```

(**TCP-CLOSEFILE**

```
  (LAMBDA (STREAM)                                          ; Edited 18-Dec-89 11:17 by jds
    ;; Close a TCP connection or listening-socket cleanly.
    (TCP 3 (|fetch| (STREAM F1) |of| STREAM))
    STREAM))
```

(**TCP-EVENTFN**

```
  (LAMBDA (FDEV EVENT)                                      ; Edited 30-Jan-90 13:56 by jds
    (SELECTQ EVENT
         ((BEFORELOGOUT BEFORESYSOUT BEFOREMAKESYS BEFORESAVEVM)
                                                            ;; Clean up existing connections, and remember any LISTENS in
                                                            ;; progress

             )
         ((AFTERLOGOUT AFTERSYSOUT AFTERMAKESYS AFTERSAVEVM)
                                                            ;; Try to reopen streams that had been open, and re-establish any
                                                            ;; LISTENs in progress when we exited.

             )
         NIL)))
```

(**TCP.BUFFERED.BOUTS**

```
  (LAMBDA (STREAM SBASE OFFSET NBYTES)
    (\\BUFFERED.BOUTS (FETCH F2 OF STREAM)
           SBASE OFFSET NBYTES)))
```

)

(**TCP-DEV-INIT**)

;; User-level TCP operations

(DEFINEQ

(**TCP**

```
  (LAMBDA (A B C D E F G H I J K L M)                       ; Edited  4-Apr-90 17:29 by jds
    ;; Generic TCP-operation hider function.  Hides the fact of TCP ops being SUBRCALLs.
```

```
    ;; Returns whatever result the TCP operation returns.
    (SUBRCALL TCP A B C D E F G H I J K L M)))
```

(OPENTCPSTREAM

```
(LAMBDA (HOST PORT)                                    ; Edited  3-May-90 11:38 by jds
  (LET ((ERRNO (CREATECELL \\FIXP))
        IODESCRIPTOR ACCESS-VALUE STREAM OTHER-STREAM)
       (SETQ STREAM
        (|create| STREAM
               BINABLE _ T
               BOUTABLE _ T
               DEVICE _ \\TCP-FDEV
               FULLFILENAME _ HOST
               USERCLOSEABLE _ T
               USERVISIBLE _ T
               EOLCONVENTION _ LF.EOLC))
       (|replace| (STREAM ACCESS) |of| STREAM |with| 'BOTH)
       (|replace| (STREAM STRMBINFN) |of| STREAM |with| (FUNCTION \\BUFFERED.BIN))
       (|replace| (STREAM STRMBOUTFN) |of| STREAM |with| (FUNCTION \\BUFFCHAR-OTHER-BOUT))
       (|replace| F2 |of| STREAM
          |with| (SETQ OTHER-STREAM
                   (|create| STREAM
                          BINABLE _ T
                          BOUTABLE _ T
                          DEVICE _ \\TCP-FDEV
                          FULLFILENAME _ HOST
                          STRMBOUTFN _ (FUNCTION \\BUFFERED.BOUT)
                          USERCLOSEABLE _ NIL
                          USERVISIBLE _ NIL
                          EOLCONVENTION _ LF.EOLC)))
       (COND
          ((SETQ IODESCRIPTOR (TCP 4 HOST PORT))
           ;; Open happened, so put things together.
           (|replace| (STREAM F1) |of| STREAM |with| IODESCRIPTOR)
           (|replace| (STREAM F1) |of| OTHER-STREAM |with| IODESCRIPTOR))
          (T (\\CHAR-ERROR ERRNO HOST)))
       STREAM)))
```

(TCP-ACCEPT

```
(LAMBDA (WAITING-SOCKET)                               ; Edited 22-May-90 10:18 by jhb
  (LET ((ERRNO (CREATECELL \\FIXP))
        IODESCRIPTOR ACCESS-VALUE STREAM OTHER-STREAM SOCKET)
       (|while| (OR (NOT SOCKET)
                    (< SOCKET 0))
          |do| (BLOCK)
               (SETQ SOCKET (TCP 8 WAITING-SOCKET)))
       (PRINTOUT *TRACE-OUTPUT* "SOCKET ACCEPTED " SOCKET T)
       (SETQ STREAM (|create| STREAM
                          BINABLE _ T
                          BOUTABLE _ T
                          DEVICE _ \\TCP-FDEV
                          FULLFILENAME _ (GETHOSTFROMSOCKET SOCKET)
                          USERCLOSEABLE _ T
                          USERVISIBLE _ T
                          EOLCONVENTION _ LF.EOLC))
       (|replace| (STREAM ACCESS) |of| STREAM |with| 'BOTH)
       (|replace| (STREAM STRMBINFN) |of| STREAM |with| (FUNCTION \\BUFFERED.BIN))
       (|replace| (STREAM STRMBOUTFN) |of| STREAM |with| (FUNCTION \\BUFFCHAR-OTHER-BOUT))
       (|replace| F2 |of| STREAM
          |with| (SETQ OTHER-STREAM
                   (|create| STREAM
                          BINABLE _ T
                          BOUTABLE _ T
                          DEVICE _ \\TCP-FDEV
                          FULLFILENAME _ HOST
                          STRMBOUTFN _ (FUNCTION \\BUFFERED.BOUT)
                          USERCLOSEABLE _ NIL
                          USERVISIBLE _ NIL
                          EOLCONVENTION _ LF.EOLC)))
       (COND
          (SOCKET
                   ;; Open happened, so put things together.
                   (|replace| (STREAM F1) |of| STREAM |with| SOCKET)
                   (|replace| (STREAM F1) |of| OTHER-STREAM |with| SOCKET))
          (T (\\CHAR-ERROR ERRNO HOST)))
       STREAM)))
```

(TCP-LISTEN

```
(LAMBDA (SOCKET-NUMBER ACCEPT-FUNCTION ACCEPT-DATA)    ; Edited 22-May-90 10:54 by jhb
  (LET ((SOCKET (TCP 7 SOCKET-NUMBER)))
       (SETQ \\MAIKO.IO-INTERRUPT-VECTOR (CONS (LIST (LLSH 1 SOCKET)
```

```
                                                    ACCEPT-FUNCTION SOCKET ACCEPT-DATA)
                                                \\MAIKO.IO-INTERRUPT-VECTOR))
        SOCKET)))
```

(TCP-CLOSE

```
  (LAMBDA (DESCRIPTOR-NUMBER)                                   ; Edited  4-Apr-90 14:51 by jds
    (LET ((ACCEPTOR (ASSOC (LLSH 1 DESCRIPTOR-NUMBER)
                           \\MAIKO.IO-INTERRUPT-VECTOR)))
         (TCP 3 DESCRIPTOR-NUMBER)                              ; Close the TCP connection
         (DREMOVE ACCEPTOR \\MAIKO.IO-INTERRUPT-VECTOR)         ; REmove any acceptor.
         DESCRIPTOR-NUMBER)))

)

(DEFINEQ
```

(UDP-LISTEN

```
  (LAMBDA (SOCKET-NUMBER ACCEPT-FUNCTION ACCEPT-INFO)           ; Edited  4-Apr-90 15:49 by jds
    ;; Listen on a particular UDP socket for incoming packet traffic.  Also has the effect of opening the socket for outgoing traffic.
    (LET ((SOCKET (TCP 128 SOCKET-NUMBER)))
         (SETQ \\MAIKO.IO-INTERRUPT-VECTOR (CONS (LIST (LLSH 1 SOCKET)
                                                       ACCEPT-FUNCTION SOCKET ACCEPT-INFO)
                                                 \\MAIKO.IO-INTERRUPT-VECTOR))
         SOCKET)))
```

(UDP-SEND

```
  (LAMBDA (SOCKET BUFFER LEN ADDR PORT)
    (TCP 130 SOCKET ADDR PORT BUFFER LEN)))
```

(UDP-RECV

```
  (LAMBDA (SOCKET)                                              ; Edited  3-May-90 11:40 by jds
    ;; Xall recvfrom() to get an incoming packet on a UDP socket.
    ;; Returns 4 results:
    ;;   The 1500-byte buffer containing the packet
    ;;   The length of the incoming packet
    ;;   The address of the guy who sent it
    ;;   The port to answer him on (or where he sent it from)
    (LET ((BUFFER (NCREATE 'VMEMPAGEP))
          LEN
          (ADDR (\\CREATECELL \\FIXP))
          (PORT (\\CREATECELL \\FIXP)))
         (SETQ LEN (TCP 131 SOCKET BUFFER 512 ADDR PORT))
         (CL:VALUES BUFFER LEN ADDR PORT))))

)

(DEFINEQ
```

(GETHOSTFROMNAME

```
  (LAMBDA (NAME)                                                ; Edited  1-Feb-90 11:26 by jds
    ;; Given a host name, return the IP address for that host.  If the host isn't found, return NIL.
    (TCP 0 NAME)))
```

(GETHOSTFROMADDR

```
  (LAMBDA (ADDR)                                                ; Edited  6-Apr-90 20:23 by jds
    ;; Given a host's IP address, return the string name of the host, or NIL if it can't be found.
    (LET* ((BUF (\\ALLOCBLOCK 100 NIL))
           (LEN (TCP 66 ADDR BUF)))
          (COND
             ((ZEROP LEN)
              NIL)
             (T (\\GETBASESTRING BUF 0 LEN))))))
```

(GETHOSTFROMSOCKET

```
  (LAMBDA (SOCKET)                                              ; Edited  1-Feb-90 11:30 by jds
    ;; Given the socket FD of a TCP connection, return the NAME of the remote host, or NIL if it can't be found.
    (LET* ((BUF (\\ALLOCBLOCK 100 NIL))
           (LEN (TCP 65 SOCKET BUF)))
          (COND
             ((ZEROP LEN)
              NIL)
             (T (CONCATLIST (FOR I FROM 0 TO (SUB1 LEN) COLLECT (\\GETBASEBYTE BUF I))))))))
```

(GETHOSTNAME

```
  (LAMBDA NIL                                                    ; Edited  6-Apr-90 20:25 by jds
    ;; Given a host's IP address, return the string name of the host, or NIL if it can't be found.
    (LET* ((BUF (\\ALLOCBLOCK 100 NIL))
           (LEN (TCP 67 BUF)))
          (COND
             ((ZEROP LEN)
              NIL)
             (T (\\GETBASESTRING BUF 0 LEN))))))

)

(DECLARE\: EVAL@LOAD DONTCOPY

;; Debugging functions &c

(RPAQ BUFFER (\\ALLOCBLOCK 100))

(DEFINEQ

(TCPRECV
  (LAMBDA (PORT)
    (LET ((LEN (TCP 6 PORT BUFFER 100)))
         (|for| I |from| 0 |to| (SUB1 LEN) |do| (PRIN1 (CHARACTER (\\GETBASEBYTE BUFFER I))))
         (TERPRI))))

(TCPSEND
  (LAMBDA (PORT BASE LEN)                                        ; Edited 15-Dec-89 15:13 by jds
    (TCP 5 PORT BASE OFFSET LEN)))

(SEEBUFFER
  (LAMBDA (BUF)
    (|for| I |from| 0 |to| 11 |do| (PRIN1 (CHARACTER (\\GETBASEBYTE BUF I))))))

(FOON
  (LAMBDA (INFO)                                                 ; Edited  4-Apr-90 17:35 by jds
    (LET ((RES (CL:MULTIPLE-VALUE-LIST (UDP-RECV (CADDR INFO)))))
         (AND (CADR RES)
              (SETQ RESULT RES)))))

)
)

(PUTPROPS TCPOPS COPYRIGHT ("Savoir, Inc." 1989 1990))
```

FUNCTION INDEX

VARIABLE INDEX