

File created: 3-Jul-2022 14:07:11 {DSK}<users>kaplan>local>medley3.5>working-medley>internal>test>filepos
>TFP.;31

changes to: (FNS TFP TFP1)

previous date: 3-Jul-2022 13:32:16 {DSK}<users>kaplan>local>medley3.5>working-medley>internal>test>filepos>TFP.
;27

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ TFPCOMS
((FNS TFP TFP1 FPC FPCS)
 (FNS OLDFFILEPOS OLDFFILEPOS)
 (FILES FPTESTS)
 (ADDVARS (DIRECTORIES {WMEDLEY}<internal>test>filepos>))
 ;; Compiling also requires EXPORTS.ALL
 (DECLARE%: DOEVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
 IOCHAR))))
```

(DEFINEQ

(TFP

```
[LAMBDA (TESTNAMES TAGS FN) ; Edited 3-Jul-2022 14:06 by rmk
 (CL:UNLESS TESTNAMES (SETQ TESTNAMES ALLTESTS))
 (LET [(TESTS (FOR TN INSIDE TESTNAMES FIRST (PRINTOUT T "Testing") JOIN (PRINTOUT T " " TN)
 (CONS (MKSTRING TN)
 (COPY (EVALV TN))))
 FINALLY (TERPRI T)]
 (CL:WHEN TAGS
 (SETQ TESTS (FOR TEST IN TESTS WHEN (THEREIS TAG INSIDE TAGS SUCHTHAT (MEMB TAG TEST))
 COLLECT TEST)))
 (PRINTOUT T (LENGTH TESTS)
 " tests" T)
 (FOR TEST VAL COMMENT PRINTED IN TESTS EACHTIME (CL:WHEN (STRINGP TEST)
 (SETQ COMMENT TEST)
 (SETQ PRINTED NIL))
 WHEN [AND (LISTP TEST)
 (NOT (AND FN (CADDR TEST)
 UNLESS (EQUAL (CAR TEST)
 (SETQ VAL (TFP1 (CADR TEST)
 FN))))
 COLLECT (CL:WHEN COMMENT
 (CL:UNLESS PRINTED (PRINTOUT T COMMENT T)))
 (PRINTOUT T 5 VAL " <- " .P2 TEST T)
 (CONS VAL TEST])
```

(TFP1

```
[LAMBDA (FPARGS FN) ; Edited 3-Jul-2022 14:04 by rmk
```

```
;; FN is the search function to apply: NIL = FILEPOS, OLDFFILEPOS, FFILEPOS. OLDFFILEPOS
;; For convenience: NIL -> FILEPOS, OF -> OLDFFILEPOS, FF -> FFILEPOS, OFF -> OLDFFILEPOS.
;; OLDFFILEPOS and OLDFFILEPOS do only a byte searches.
;; FPARGS is a list of FILEPOS args. CASEARRAY=T means Transparent case array, pushes to FFILEPOS.
;; The file extension gives the format, defaulting to *DEFAULT-EXTERNALFORMAT* = :XCCS
```

```
(SETQ FN (SELECTQ FN
 ((NIL FILEPOS)
 'FILEPOS)
 ((FF FFILEPOS)
 'FFILEPOS)
 ((OF OLDFFILEPOS)
 'OLDFFILEPOS)
 ((OFF OLDFFILEPOS)
 'OLDFFILEPOS)
 (HELP "BAD FN" FN)))
 (CL:WHEN (OR (FIXP (CAR FPARGS))
 (NULL (CAR FPARGS))
 (AND (LISTP (CAR FPARGS))
 (FIXP (CAAR FPARGS))
 (FIXP (CDAR FPARGS))))
 (LISTP (CADR FPARGS)))
 (SETQ FPARGS (CADR FPARGS)))
 (LET (STREAM VAL PATTERN FILE START END SKIP TAIL CASEARRAY EXT (FORMAT *DEFAULT-EXTERNALFORMAT*))
 (SETQ PATTERN (EVAL (POP FPARGS))) ; So we can do substrings, CHARACTER etc.
 (SETQ FILE (POP FPARGS))
 (SETQ START (POP FPARGS))
 (SETQ END (POP FPARGS))
 (SETQ SKIP (POP FPARGS))
 (SETQ TAIL (POP FPARGS))
 (SETQ CASEARRAY (POP FPARGS))
 (SETQ EXT (FILENAMEFIELD.STRING FILE 'EXTENSION))
```

```
(CL:WHEN EXT
  (CL:WHEN (STRPOS "UTF8" EXT)
    (SETQ EXT "UTF-8"))
  [SETQ FORMAT (FIND-FORMAT (CL:INTERN EXT 'KEYWORD))]
  [SETQ STREAM (OPENSTREAM (OR (FINDFILE FILE T)
    FILE)
    'INPUT NIL `((FORMAT ,FORMAT]
  (SETQ CASEARRAY (IF (EQ CASEARRAY T)
    THEN (CASEARRAY)
    ELSE (EVAL CASEARRAY)))
  (SETQ VAL (APPLY* FN PATTERN STREAM START END SKIP TAIL CASEARRAY))
  (CLOSEF? STREAM)
  VAL])
```

(FPC

[LAMBDA (STR FILE START END SKIP TAIL CASEARRAY) ; Edited 29-Jun-2022 21:22 by rmk

:: Compare old and new filepos

```
(LET (OLD NEW EXT FORMAT)
  (CL:UNLESS (STREAMP FILE)
    (SETQ EXT (FILENAMEFIELD.STRING FILE 'EXTENSION))
    (SETQ FORMAT (CL:INTERN EXT 'KEYWORD))
    (CL:UNLESS (FIND-FORMAT FORMAT T)
      (SETQ FORMAT :XCCS))
    (STREAMPROP FILE 'FORMAT FORMAT))
  (SETQ OLD (OLDFILEPOS STR FILE START END SKIP TAIL CASEARRAY))
  (SETQ NEW (FILEPOS STR FILE START END SKIP TAIL CASEARRAY))
  (CLOSEF FILE)
  (CL:UNLESS (EQUAL OLD (IF (EQ TAIL 'BOTH)
    THEN (CDR NEW)
    ELSE NEW))
    (HELP (CONCAT "OLD=" (OR OLD "NIL")
      " NEW="
      (OR NEW "NIL"))))
  (LIST OLD NEW])
```

(FPCS

[LAMBDA (STR FILE START END SKIP TAIL) ; Edited 29-Jun-2022 23:56 by rmk
; Edited 28-Jun-2022 22:21 by rmk

:: Compare old and new slow filepos

```
(LET (FAST SLOW EXT FORMAT)
  (CL:UNLESS (STREAMP FILE)
    (SETQ EXT (FILENAMEFIELD.STRING FILE 'EXTENSION))
    (SETQ FORMAT (CL:INTERN EXT 'KEYWORD))
    (CL:UNLESS (FIND-FORMAT FORMAT T)
      (SETQ FORMAT :XCCS))
    (STREAMPROP FILE 'FORMAT FORMAT))
  (SETQ FAST (FILEPOS STR FILE START END SKIP TAIL))
  (SETQ SLOW (FILEPOS STR FILE START END SKIP TAIL (CASEARRAY)))
  (CLOSEF FILE)
  (CL:UNLESS (EQUAL FAST SLOW)
    (HELP (CONCAT "FAST=" (OR FAST "NIL")
      " SLOW="
      (OR SLOW "NIL"))))
  (LIST FAST SLOW])
```

)

(DEFINEQ

(OLDFILEPOS

[LAMBDA (STR FILE START END SKIP TAIL CASEARRAY) ; Edited 27-Jun-2022 23:35 by rmk
; Edited 10-Aug-2020 21:44 by rmk:
(* Pavel "12-Oct-86 15:13")

:: RMK: Added coercion from internal XCCS string to UTF8 if searching a UTF8 file

:: NB: this function now works on non-PAGEMAPPED files. It must use only IO functions that respect that.

```
(PROG ((SKIPCHAR (AND SKIP (CHCON1 SKIP)))
  [CA (fetch (ARRAYP BASE) of (COND
    [CASEARRAY (COND
      ((AND (ARRAYP CASEARRAY)
        (EQ (fetch (ARRAYP TYP) of CASEARRAY)
          \ST.BYTE))
      CASEARRAY)
      (T (CASEARRAY CASEARRAY))
      (T \TRANSPARENT]
    (STREAM (\GETSTREAM FILE 'INPUT))
    CHAR FIRSTCHAR STRBASE STRINDEX PATLEN PATINDEX ORGFILEPTR LASTINDEX STARTBYTE ENDBYTE BIGENDBYTE
    STARTSEG ENDSEG)
  (CL:WHEN (EQ :UTF-8 (\EXTERNALFORMAT STREAM))
    (SETQ STR (XTOUSTRING STR)))
  [COND
    ((LITATOM STR)
      (SETQ STRBASE (fetch (LITATOM PNAMEBASE) of STR))
```

```

    (SETQ STRINDEX 1)
    (SETQ PATLEN (fetch (LITATOM PNAMELENGTH) of STR))
    (T (OR (STRINGP STR)
           (SETQ STR (MKSTRING STR))))
       (SETQ STRBASE (fetch (STRINGP BASE) of STR))
       (SETQ STRINDEX (fetch (STRINGP OFFST) of STR))
       (SETQ PATLEN (fetch (STRINGP LENGTH) of STR)) ; calculate start addr and set file ptr.
    [SETQ STARTBYTE (COND
                     (START (COND
                              ((NOT (AND (FIXP START)
                                           (IGEQ START 0)))
                               (LISPERROR "ILLEGAL ARG" START)))
                              (SETQ ORGFILEPTR (\GETFILEPTR STREAM))
                              (\SETFILEPTR STREAM START)
                              START)
                              (T (SETQ ORGFILEPTR (\GETFILEPTR STREAM))
                                  ; calculate the character address of the character after the last
                                  ; possible match.
                              )
                     )
    [SETQ ENDBYTE (ADD1 (COND
                       ((NULL END) ; Default is end of file
                        (IDIFFERENCE (\GETEOFPTR STREAM)
                                     PATLEN))
                       ((IGEQ END 0) ; Absolute byte pointer given
                        (IMIN END (IDIFFERENCE (\GETEOFPTR STREAM)
                                               PATLEN)))
                       ((IGREATERP PATLEN (IMINUS END))
                        ; END is too far, use eof less length
                        (IDIFFERENCE (\GETEOFPTR STREAM)
                                     PATLEN))
                       (T (IDIFFERENCE (IPLUS (\GETEOFPTR STREAM)
                                               END 1)
                                       PATLEN)
                          )
                       )
    ]

```

:: use STARTBYTE and ENDBYTE instead of START and END because vm functions shouldn't change their arguments.

```

    (COND
      ((IGEQ STARTBYTE ENDBYTE) ; nothing to search
       (GO FAILED)))
    (SETQ LASTINDEX PATLEN)
    SKIPLP ; set the first character to FIRSTCHAR, handling leading skips.

    (COND
      ((EQ LASTINDEX 0) ; null case
       (GO FOUNDIT))
      ((EQ (SETQ FIRSTCHAR (\GETBASEBYTE CA (\GETBASEBYTE STRBASE STRINDEX)))
           SKIPCHAR) ; first character in pattern is skip.
       (SETQ LASTINDEX (SUB1 LASTINDEX))
       (\BIN STREAM) ; Move forward a character.
       (add STRINDEX 1)
       (add STARTBYTE 1)
       (GO SKIPLP)))
      (SETQ LASTINDEX (IPLUS LASTINDEX STRINDEX)) ; Used for end of pattern check, comparing against current
      ; INDEX

    [COND
      ((SMALLP ENDBYTE)
       (SETQ STARTSEG (SETQ ENDSEG 0)))
      (T ; The search will be in the large integers at least part of the time, so split the start and end fileptrs into hi and lo parts. The
         ; 'segment' size we choose is smaller than 2^16 so that we are still smallp near the boundary (can get around that here by
         ; decrementing everyone, but can't in FFILEPOS). Note that STARTBYTE and ENDBYTE are never actually used as file ptrs, just
         ; for counting.
         (SETQ ENDSEG (FOLDLO ENDBYTE FILEPOS.SEGMENT.SIZE))
         (SETQ BIGENDBYTE (IMOD ENDBYTE FILEPOS.SEGMENT.SIZE))
         (SETQ STARTSEG (FOLDLO STARTBYTE FILEPOS.SEGMENT.SIZE))
         (SETQ STARTBYTE (IMOD STARTBYTE FILEPOS.SEGMENT.SIZE))
         (SETQ ENDBYTE (COND
                       ((EQ STARTSEG ENDSEG)
                        BIGENDBYTE)
                       (T ; In different segments, so we'll have to search all the way to the end of this seg; hence, 'end' is
                          ; currently as big as it gets
                          FILEPOS.SEGMENT.SIZE)
                       )
         )
      ]
    FIRSTCHARLP

```

:: STARTBYTE is the possible beginning of a match. the file ptr of the file is always at STARTBYTE position when the FIRSTCHAR loop is passed.

```

    (COND
      ((EQ STARTBYTE ENDBYTE) ; end of this part of search
       (COND
         ((EQ STARTSEG ENDSEG) ; failed
          (GO FAILED)))
         (SETQ STARTBYTE 0) ; Finished this segment, roll over into new one
         ; = STARTBYTE-FILEPOS.SEGMENT.SIZE
       )
      [COND
        ((EQ (add STARTSEG 1)
             ENDSEG) ; Entering final segment, so set ENDBYTE to actual end instead
         ; of segment end
        )
      ]
    (COND

```

```

      ((EQ (SETQ ENDBYTE BIGENDBYTE)
           0)
       (GO FAILED]
      (GO FIRSTCHARLP))
      ((NEQ FIRSTCHAR (\GETBASEBYTE CA (\BIN STREAM)))
       (add STARTBYTE 1)
       (GO FIRSTCHARLP)))
      (SETQ PATINDEX STRINDEX)
MATCHLP
; At this point, STR is matched thru offset PATINDEX

(COND
  ((EQ (SETQ PATINDEX (ADD1 PATINDEX))
       LASTINDEX)
   ; matched for entire length
   (GO FOUNDIT))
  ((OR (EQ (SETQ CHAR (\GETBASEBYTE CA (\GETBASEBYTE STRBASE PATINDEX)))
           (\GETBASEBYTE CA (\BIN STREAM)))
       (EQ CHAR SKIPCHAR))
   ; Char from file matches char from STR
   (GO MATCHLP))
  (T
   ; Match failed, so we have to start again with first char
   (\SETFILEPTR STREAM (IDIFFERENCE (\GETFILEPTR STREAM)
                                     (IDIFFERENCE PATINDEX STRINDEX)))
   ;; Back up over the chars we have just read in trying to match, less one. I.e. go back to one past the previous starting point
   (add STARTBYTE 1)
   (GO FIRSTCHARLP)))
FOUNDIT
; set fileptr, adjust for beginning skips and return proper value.

(COND
  ((NOT TAIL)
   ; Fileptr wants to be at start of string
   (\SETFILEPTR STREAM (IDIFFERENCE (\GETFILEPTR STREAM)
                                     PATLEN])
   (RETURN (\GETFILEPTR STREAM))
  FAILED
; return the fileptr to its initial position.

  (\SETFILEPTR STREAM ORGFILEPTR)
  (RETURN NIL))

```

(OLDDFFILEPOS

```

[LAMBDA (PATTERN FILE START END SKIP TAIL CASEARRAY)
; Edited 10-Aug-2020 21:44 by rmk:
;; RMK: Added coercion from internal XCCS string to UTF8 if searching a UTF8 file ( Pavel "12-Oct-86 15:20")
(PROG ([STREAM (\GETSTREAM (OR FILE (INPUT)
PATBASE PATOFFSET PATLEN ORGFILEPTR STARTOFFSET ENDOFFSET BIGENDOFFSET STARTSEG ENDSEG EOF)
(COND
  (SKIP
   ; Slow case--use FILEPOS
   (GO TRYFILEPOS))
  ((NOT (fetch PAGEMAPPED of (fetch (STREAM DEVICE) of STREAM)))
   ; This is a non-page-oriented file. Use FILEPOS instead.
   ; calculate start addr and set file ptr.
   (GO TRYFILEPOS)))
(CL:WHEN (EQ :UTF8 (\EXTERNALFORMAT STREAM))
  (SETQ PATTERN (XTOUSTRING PATTERN)))
[COND
  ((LITATOM PATTERN)
   (SETQ PATBASE (fetch (LITATOM PNAMEBASE) of PATTERN))
   (SETQ PATOFFSET 1)
   (SETQ PATLEN (fetch (LITATOM PNAMELENGTH) of PATTERN)))
  (T (OR (STRINGP PATTERN)
         (SETQ PATTERN (MKSTRING PATTERN)))
       (SETQ PATBASE (fetch (STRINGP BASE) of PATTERN))
       (SETQ PATOFFSET (fetch (STRINGP OFFST) of PATTERN))
       (SETQ PATLEN (fetch (STRINGP LENGTH) of PATTERN]
(COND
  ((OR (IGREATERP PATLEN \MAX.PATTERN.SIZE)
       (ILESSP PATLEN \MIN.PATTERN.SIZE))
   (GO TRYFILEPOS)))
(SETQ ORGFILEPTR (\GETFILEPTR STREAM))
(SETQ STARTOFFSET (IPLUS (COND
  (START (COND
    ((NOT (AND (FIXP START)
               (IGEQL START 0)))
     (LISPERROR "ILLEGAL ARG" START)))
    START)
  (T ORGFILEPTR))
  (SUB1 PATLEN)))
; STARTOFFSET is the address of the character corresponding
; to the last character of PATTERN.
; calculate the character address of the character after the last
; possible match.
(SETQ EOF (\GETEOFPTR STREAM))
[SETQ ENDOFFSET (COND
  ((NULL END)
   EOF)
  (T (IMIN (IPLUS (COND
    ((ILESSP END 0)
     (IPLUS EOF END 1))
    (T END))
   PATLEN)
  EOF]

```

;; use STARTOFFSET and ENDOFFSET instead of START and END because vm functions shouldn't change their arguments.

```
(COND
  ((IGEQ STARTOFFSET ENDOFFSET) ; nothing to search
   (RETURN))
  ((ILESSP (IDIFFERENCE ENDOFFSET STARTOFFSET)
            \MIN.SEARCH.LENGTH) ; too small to make FFFILEPOS worthwhile
   (GO TRYFILEPOS)))
(\SETFILEPTR STREAM STARTOFFSET)
[RETURN (GLOBALRESOURCE (\FFDELTA1 \FFDELTA2 \FFPATCHAR)
  (PROG ((CASE (fetch (ARRAYP BASE) of (COND
    [CASEARRAY (COND
      ((AND (ARRAYP CASEARRAY)
            (EQ (fetch (ARRAYP TYP)
                      of CASEARRAY)
                \ST.BYTE))
      CASEARRAY)
      (T (CASEARRAY CASEARRAY)
          (T \TRANSPARENT))))))
  (DELTA1 (fetch (ARRAYP BASE) of \FFDELTA1))
  (DELTA2 (fetch (ARRAYP BASE) of \FFDELTA2))
  (PATCHAR (fetch (ARRAYP BASE) of \FFPATCHAR))
  (MAXPATINDEX (SUB1 PATLEN))
  CHAR CURPATINDEX LASTCHAR INC)
```

;; Use Boyer-Moore string search algorithm. Use two auxiliary tables, DELTA1 and DELTA2, to tell how far ahead to
 ;; move in the file when a partial match fails. DELTA1 contains, for each character code, the distance of that
 ;; character from the right end of the pattern, or PATLEN if the character does not occur in the pattern. DELTA2
 ;; contains, for each character position in the pattern, how far ahead to move such that the partial substring
 ;; discovered to the right of the position now matches some other substring (to the left) in the pattern. PATCHAR is
 ;; just PATTERN translated thru CASEARRAY

```
(\SETUP.FFILEPOS PATBASE PATOFFSET PATLEN PATCHAR DELTA1 DELTA2 CASE)
[COND
  ((SMALLP ENDOFFSET)
   (SETQ STARTSEG (SETQ ENDSEG 0)))
  (T ; The search will be in the large integers at least part of the time, so split the start and end fileptrs into
      ;; hi and lo parts. The 'segment' size we choose is smaller than 2^16 so that we are still small near
      ;; the boundary. Note that STARTOFFSET and ENDOFFSET are never actually used as file ptrs, just
      ;; for counting.
      (SETQ ENDSEG (FOLDLO ENDOFFSET FILEPOS.SEGMENT.SIZE))
      (SETQ BIGENDOFFSET (MOD ENDOFFSET FILEPOS.SEGMENT.SIZE))
      (SETQ STARTSEG (FOLDLO STARTOFFSET FILEPOS.SEGMENT.SIZE))
      (SETQ STARTOFFSET (MOD STARTOFFSET FILEPOS.SEGMENT.SIZE))
      (SETQ ENDOFFSET (COND
        ((EQ STARTSEG ENDSEG)
         BIGENDOFFSET)
        (T
         ;; In different segments, so we'll have to search all the way to the end of this seg; hence,
         ;; 'end' is currently as big as it gets
         FILEPOS.SEGMENT.SIZE)
        ))
      (SETQ LASTCHAR (GETBASEBYTE PATCHAR MAXPATINDEX))
      FIRSTCHARLP
      (COND
        [(IGEQ STARTOFFSET ENDOFFSET) ; End of this chunk
         (COND
           ((EQ STARTSEG ENDSEG) ; failed
            (GO FAILED))
           (T ; Finished this segment, roll over into new one
            (add STARTSEG 1)
            (SETQ STARTOFFSET (IDIFFERENCE STARTOFFSET FILEPOS.SEGMENT.SIZE))
            (COND
              ((EQ STARTSEG ENDSEG)
               (SETQ ENDOFFSET BIGENDOFFSET)))
            (GO FIRSTCHARLP)
            ((NEQ (SETQ CHAR (GETBASEBYTE CASE (\BIN STREAM)))
                 LASTCHAR)
             (add STARTOFFSET (SETQ INC (GETBASEBYTE DELTA1 CHAR)))
             (OR (EQ INC 1)
                 (\INCFILEPTR STREAM (SUB1 INC)))
             ; advance file pointer accordingly (\BIN already advanced it one)
            (GO FIRSTCHARLP)))
            (SETQ CURPATINDEX (SUB1 MAXPATINDEX))
            MATCHLP
            (COND
              ((ILESSP CURPATINDEX 0)
               (GO FOUNDIT)))
              (\DECFILEPTR STREAM 2) ; back up to read previous char
            (COND
              ((NEQ (SETQ CHAR (GETBASEBYTE CASE (\BIN STREAM)))
                   (GETBASEBYTE PATCHAR CURPATINDEX))
               ; Mismatch, advance by greater of delta1 and delta2
               (add STARTOFFSET (IDIFFERENCE (SETQ INC (IMAX (GETBASEBYTE DELTA1 CHAR)
                                                             (GETBASEBYTE DELTA2 CURPATINDEX)
                                                             )))
                   (IDIFFERENCE MAXPATINDEX CURPATINDEX)))
```

```

      (OR (EQ INC 1)
          (\INCFILEPTR STREAM (SUB1 INC)))
      (GO FIRSTCHARLP))
      (SETQ CURPATINDEX (SUB1 CURPATINDEX))
      (GO MATCHLP)
FOUNDIT
      (\INCFILEPTR STREAM (COND
          (TAIL (SUB1 PATLEN)) ; Put fileptr at end of string
          (T ; back up over the last char we looked at, i.e. the first char of
              ; string
              -1)))
      (RETURN (\GETFILEPTR STREAM))
FAILED
      (\SETFILEPTR STREAM ORGFILEPTR) ; return the fileptr to its initial position.
      (RETURN NIL]

```

```

TRYFILEPOS
  (RETURN (FILEPOS PATTERN STREAM START END SKIP TAIL CASEARRAY])
)

```

```

(FILESLoad FPTESTS)

```

```

(ADDTovar DIRECTORIES {WMEDLEY}<internal>test>filepos>)

```

:: Compiling also requires EXPORTS.ALL

```

(DECLARE%: DOEVAL@COMPILE DONTCOPY

```

```

(FILESLoad (LOADCOMP)
  IOCHAR)
)

```

FUNCTION INDEX

FPC2 FPCS2 OLDFILEPOS ...4 OLDFILEPOS2 TFP1 TFP11

VARIABLE INDEX

DIRECTORIES ...6
