



Tech Info Library

Pascal: Speeding up Pascal text file reading (2 of 3)

Revised: 11/30/84
Security: Everyone

Pascal: Speeding up Pascal text file reading (2 of 3)

```
=====

BEGIN
  IF BUFINDEX >= NOTNULLS THEN FILLBUFFER;
      (* If the buffer needs refilling,
        go and get another buffer. *)
  IF NOT EMPTY THEN BEGIN (* If the file is not yet empty,
      then do the following: *)
    LINELEN := SCAN (BUFSIZE, = CHR(13), BUFFER [BUFINDEX]);
      (* Set LINELEN to the number of
        characters from the current
        buffer pointer position
        (BUFINDEX) to the next carriage
        return in the buffer.*)
    IF BUFFER [BUFINDEX] = CHR (16) THEN BEGIN
      (* If the character at the buffer
        index is an ASCII DLE, then we
        have to unpack the leading
        spaces. *)
      INDENT := ORD (BUFFER [BUFINDEX + 1]) - 32;
      (* Set INDENT to the number found
        at BUFINDEX + 1, the number
        of space characters to
        insert. *)

      (*$R-*)
      LINE [0] := CHR (LINELEN + INDENT - 2);
      (*$R+*)      (* Turn off Range Checking so we can
                    manually change the string length. Set
                    the length of LINE to the number we had
                    already gotten plus the number of spaces
                    to unpack, throwing away two bytes for
                    the DLE and count bytes. Turn Range
                    Checking back on.*)
    IF INDENT > 0 THEN FILLCHAR (LINE [1], INDENT, ' ');
      (* If there are spaces, then fill in the
        appropriate number of them, starting
        with the first position in the new
        string. *)
```

```
IF LINELEN > 2 THEN MOVELEFT (BUFFER [BUFINDEX + 2],
    LINE [1 + INDENT], LINELEN - 2);
    (* If the string is more than 2
    characters long, then move the rest of
    it from the buffer into the string
    starting just after the leading spaces
    previously inserted. *)

END ELSE BEGIN
    (* No DLE character was found. That
    means straight ASCII. *)

    (*$R-*)
    LINE [0] := CHR (LINELEN);
    (*$R+*)

    (* Turn Range Checking off, set the
    length of the string to LINELEN, and
    turn Range Checking back on. *)
    IF LINELEN > 0 THEN MOVELEFT (BUFFER [BUFINDEX],
        LINE [1], LINELEN);
        (* Move the characters from the buffer
        into LINE as above. *)

    END;
    BUFINDEX := BUFINDEX + LINELEN + 1;
    (* Sets the pointer to the first
    character of the next string in the
    buffer for the next time through. *)

END;
END;
```

Here's a program that demonstrates the difference in speed between the two methods of reading strings:

```
PROGRAM QUICKREAD; (* Very fast line read routine *)
CONST BUFSIZE = 1024;
    BUFLen = 1023;
    FILENAME = 'QWERTY9.TEXT';
(* Probably not on user disk *)
VAR LINE: STRING;
    INFILE: FILE;
    TEXTFILE: TEXT;
    CH, OPTION: CHAR;
    EMPTY, HELL_FREEZES_OVER: BOOLEAN;
    ERROR: INTEGER;
    NOTNULLS: 0..BUFSIZE;
(* # of non-null chars *)
    BUFINDEX: 0..BUFSIZE;
(* Index within buffer *)
    BUFFER: PACKED ARRAY [0..BUFLen] of CHAR;
```

Apple Tech Notes

Tech Info Library Article Number:652