Pascal: Speeding up Pascal text file reading (1 of 3)

Although the READLN statement reads Pascal text files--for example:

```
READLN (FILEID, STRINGVARIABLE);
```

--this operation can be made much faster by using the routines contained in the following program. Three procedures do the work. Their operation is explained, line-by-line, below:

```
PROCEDURE FILLBUFFER;
                        (* Fills the working buffer with data
                           from the .TEXT file.  *)
   BEGIN
   EMPTY := BLOCKREAD (INFILE,BUFFER,2) = 0;
                        (* Reads 2 blocks of the file into
                           BUFFER and leaves the variable EMPTY
                           equal to zero if the end of file
                           marker is not yet reached.*)
   IF NOT EMPTY THEN BEGIN
                        (* If there is still unprocessed data
                           in the buffer, do this:*)
   NOTNULLS := BUFSIZE +
               SCAN (- BUFSIZE, <> CHR(0), BUFFER [1023]);
                        (* The length of a Pascal .TEXT file
                           should always be in multiples of 2
                           blocks.  Since strings (lines) do not
                           span blocks, each block is likely to
                           contain nulls (ASCII ZERO) at the end.
                           This line returns the number of real
                           characters in the file, and discards
                           the null ones.  *)
   BUFINDEX := 0;
                        (* The working index into the buffer is
                           reset to zero after refilling the
                           buffer.  *)
   END;
END;

PROCEDURE OPENFILE (FNAME:  STRING);
                        (* Opens the file using the name passed
                           by calling procedure.*)
BEGIN
   IF ((POS('.text',FNAME) = 0) AND
```

```
                              (POS('.TEXT',FNAME) = 0)) THEN
                              FNAME := CONCAT (FNAME,'.TEXT');
                           (* Adds the .TEXT suffix if it's not
                              already there.  *)

   RESET (INFILE, FNAME);
                           (* Actually opens the referenced file.  *)

   FILLBUFFER; FILLBUFFER;
                           (* The first call to FILLBUFFER skips
                              over the 2 blocks of header
                              information on .TEXT files.  The
                              second call actually fills the buffer
                              with information which will be used.*)
END;

PROCEDURE READFILE (VAR LINE:  STRING);

(* Reads from the file and returns a string at a time in the variable LINE. A
   word about the Pascal .TEXT file format: Lines are stored as ASCII
   characters terminated with carriage returns.  If a line contains any leading
   spaces, and most Pascal source files contain some, these spaces are "packed"
   into two bytes.  The first byte is an ASCII DLE (decimal 16) signifying that
   the line is packed.  The second byte is a count of spaces to be expanded.
   The Editor unpacks these lines automatically, as does a READLN from a file.
   We do that operation ourselves in this procedure.  The increase in speed is
   because we are using highly specialized routine, whereas the READLN
   intrinsic is very general in nature, accepting strings, integers and reals
   from the keyboard as well as from files.  Note that this format is optimized
   for Pascal source files and it wastes two bytes for each and every line that
   does not contain leading spaces. *)

VAR INDENT, LINELEN:  INTEGER;  (* INDENT is the number of
                                   space characters to add.
                                   LINELEN is the length of the
                                   new string to be formed.  *)
Apple Tech Notes


Keywords:  <None>

=================================================================

This information is from the Apple Technical Information Library.

19960215 11:05:19.00
```