Subject: Re: Read Total lines in an ASCII file
Posted by Jeff Guerber on Sat, 21 Dec 2002 02:41:17 GMT
View Forum Message <> Reply to Message

On 20 Dec 2002, Big Bird wrote:

> As far as I recall, the OP just wanted to know the number of lines,
> not necessarily try to convert them into anything. The only deviation
> from the usual 10b linefeed out there on idl'ish platforms is the DOS
> [10b,13b] LF/CR, right? Or do VMS systems do yet something different?
> How do the various suggested methods hold up on VMS?

I haven't used VMS in a number of years (thank goodness!), but IIRC it had about 5 or 6 different file types, most of which could be used for ASCII files... Ahh, here's my old copy of "Programming in VAX Fortran" (1984 ed; I kept it because it makes a pretty good Fortran 77 reference): the RECORDTYPE keyword to OPEN had 6 possible values (fixed, variable, segmented, stream, stream_cr, stream_lf). As I recall, "variable", which started each record with a 2-byte (4-byte?) length count, was pretty common for text files, even more than the "stream" types. Then there was ORGANIZATION which could be sequential, relative, or indexed...

And I think I read somewhere that Macs use (used?) just CR.

```
>  If the LF and CR/LF are the only two, the only thing you'd have to do
>  is counting the number of 10b in the byte-filed:
>
>    f = read_binary('Big_honking_example_file')
>    h = histogram(f)
>    print,h[10]
>       1479054
>
>  If you're really intent on accessing the individual data items in the
>  file, you could retain the reverse indices of the histogram for a
>  handy field of pointers to each individual line that can be converted
>  into a string at will...
```

That's pretty cool, but wouldn't they point to the _ends_ of the lines? I suppose you could add 1, pop off the last value, and push a 0 onto the front of the array.

Jeff Guerber