Subject: Re: Write to a certain line of a file
Posted by natha on Wed, 18 Jan 2012 20:05:53 GMT
View Forum Message <> Reply to Message

If the file is an ascii file you can read the file, modify the content
and then rewrite everything.

```
content=STRARR(FILE_LINES(file))

OPENR, lun, file, /GET_LUN
READU, lun, content
FREE_LUN, lun

;; then u modify the content
content[i]='new content'

;; rewrite everything
OPENW, lun, file, /GET_LUN
PRINTF, lun, content
FRRE_LUN, lun
```

Cheers,
nara

Subject: Re: Write to a certain line of a file
Posted by Michael Galloy on Wed, 18 Jan 2012 20:08:22 GMT
View Forum Message <> Reply to Message

On 1/18/12 12:49 PM, Zhang Bo wrote:
> Is there a way open a file and go to the nth line and change the
> content of this line, left the rest of file intact.

Read the file into a string array, change the nth line, and then write
the file back to the file.

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL, A Guide to Learning IDL: http://modernidl.idldev.com
Research Mathematician
Tech-X Corporation

Subject: Re: Write to a certain line of a file
Posted by Matt Francis on Thu, 19 Jan 2012 01:16:17 GMT

View Forum Message <> Reply to Message

This all sounds very inefficient, in terms of the I/O overheads for a
single change. Is there really no other way of doing this in IDL?

Subject: Re: Write to a certain line of a file
Posted by Brian Wolven on Thu, 19 Jan 2012 02:02:11 GMT
View Forum Message <> Reply to Message

You *could* write C code using fseek, then compile it as a sharable object and call it from IDL.
This is not likely to be much of an improvement, however. ;)

Subject: Re: Write to a certain line of a file
Posted by Brian Daniel on Thu, 19 Jan 2012 12:18:55 GMT
View Forum Message <> Reply to Message

I haven't done this, but POINT_LUN may do what you want.  You'd need
to know the number of bytes per line.  You will still have to read the
entire line, change the bit you want to change, move the LUN back to
the beginning of the line, and rewrite it.

```
OpenR, LUN, filename, /Get_Lun
Point_Lun, LUN, line_number * bytes_per_line
tmp=''
ReadF, LUN, tmp
; change the line as you see fit
new_line = tmp + 'changed line!'
Point_Lun, LUN, line_number * bytes_per_line
PrintF, LUN, new_line
Free_Lun, LUN
```

Code is UNTESTED!

Subject: Re: Write to a certain line of a file
Posted by Mark Piper on Thu, 19 Jan 2012 16:43:47 GMT
View Forum Message <> Reply to Message

On 1/19/2012 5:18 AM, Brian J. Daniel wrote:

> I haven't done this, but POINT_LUN may do what you want.  You'd need
> to know the number of bytes per line.  You will still have to read the
> entire line, change the bit you want to change, move the LUN back to
> the beginning of the line, and rewrite it.
>
> OpenR, LUN, filename, /Get_Lun

```
> Point_Lun, LUN, line_number * bytes_per_line
> tmp=''
> ReadF, LUN, tmp
> ; change the line as you see fit
> new_line = tmp + 'changed line!'
> Point_Lun, LUN, line_number * bytes_per_line
> PrintF, LUN, new_line
> Free_Lun, LUN
>
> Code is UNTESTED!
```

To add to Brian's idea, if you have a text file, use SKIP_LUN with the
LINES keyword set. A crude example:

```
pro replace_line_ex
   compile_opt idl2

   f0 = 'ascii.txt'
   f1 = 'ascii-1.txt'
   file_copy, file_which(f0), f1, /overwrite

   nskip = 7
   line = ''
   replacement = 'My dog has fleas.'

   openu, u, f1, /get_lun
   skip_lun, u, nskip, /lines
   point_lun, -u, mark
   readf, u, line
   nchars = strlen(line)
   point_lun, u, mark
   printf, u, replacement, format='(a-' + strtrim(nchars,2) + ')'
   free_lun, u
end
```

Diff 'ascii.txt' in the example/data directory of the IDL distro with
'ascii-1.txt' in your working directory.

mp

## Subject: Re: Write to a certain line of a file
## Posted by Michael Galloy on Thu, 19 Jan 2012 17:49:23 GMT

View Forum Message <> Reply to Message

On 1/19/12 9:43 AM, Mark Piper wrote:
> On 1/19/2012 5:18 AM, Brian J. Daniel wrote:
>> I haven't done this, but POINT_LUN may do what you want. You'd need

```
>>  to know the number of bytes per line. You will still have to read the
>>  entire line, change the bit you want to change, move the LUN back to
>>  the beginning of the line, and rewrite it.
>>
>>  OpenR, LUN, filename, /Get_Lun
>>  Point_Lun, LUN, line_number * bytes_per_line
>>  tmp=''
>>  ReadF, LUN, tmp
>>  ; change the line as you see fit
>>  new_line = tmp + 'changed line!'
>>  Point_Lun, LUN, line_number * bytes_per_line
>>  PrintF, LUN, new_line
>>  Free_Lun, LUN
>>
>>  Code is UNTESTED!
>
>  To add to Brian's idea, if you have a text file, use SKIP_LUN with the
>  LINES keyword set. A crude example:
>
>  pro replace_line_ex
>  compile_opt idl2
>
>  f0 = 'ascii.txt'
>  f1 = 'ascii-1.txt'
>  file_copy, file_which(f0), f1, /overwrite
>
>  nskip = 7
>  line = ''
>  replacement = 'My dog has fleas.'
>
>  openu, u, f1, /get_lun
>  skip_lun, u, nskip, /lines
>  point_lun, -u, mark
>  readf, u, line
>  nchars = strlen(line)
>  point_lun, u, mark
>  printf, u, replacement, format='(a-' + strtrim(nchars,2) + ')'
>  free_lun, u
>  end
>
>  Diff 'ascii.txt' in the example/data directory of the IDL distro with
>  'ascii-1.txt' in your working directory.
```

You are essentially treating the file as a binary file where you have more options, but also more responsibility for keeping things straight.

Sure, this would work in the case where the new output has the exact same length in bytes as the old output, but if not everything is going

to get screwed up. You could rewrite the file after the point of
insertion to fix this, but how much is it worth it to get around writing
this file?

Mike
--
Michael Galloy
www.michaelgalloy.com
Modern IDL, A Guide to Learning IDL: http://modernidl.idldev.com
Research Mathematician
Tech-X Corporation

## Subject: Re: Write to a certain line of a file
## Posted by David Fanning on Thu, 19 Jan 2012 19:02:34 GMT

View Forum Message <> Reply to Message

Michael Galloy writes:

> You are essentially treating the file as a binary file where you have
> more options, but also more responsibility for keeping things straight.
>
> Sure, this would work in the case where the new output has the exact
> same length in bytes as the old output, but if not everything is going
> to get screwed up. You could rewrite the file after the point of
> insertion to fix this, but how much is it worth it to get around writing
> this file?

I was curious how long it would take to re-write a 10000 line ASCII file
after replacing a line in it. Answer: 0.014 seconds!

Here is the code I used.

```
 ;------------------------------------------------------------ ----
; Create a random set of strings.
LENGTH=10000
len = Round(Scale_Vector(Randomu(-3L, LENGTH), 10, 60))
file = StrArr(LENGTH)
FOR j=0,N_Elements(len)-1 DO BEGIN
   str = Byte(Scale_Vector(RandomU(seed, len[j]), 97, 122))
   space = Fix(randomu(seed, len[j]/5 > 1) * len[j] > 2)
   str[space] = 32B
   file[j] = String(str)
ENDFOR

; Write the random strings to a file.
OpenW, lun, 'test_ascii_file.txt', /Get_Lun
```

```
FOR j=0, LENGTH-1 DO PrintF, lun, file[j]
Free_Lun, lun

; Replace line 50 with 'The dog ate my homework!'
start = Systime(1)
lines = File_Lines('test_ascii_file.txt')
OpenR, lun, 'test_ascii_file.txt', /Get_Lun
strings = StrArr(lines)
strings[49] = 'The dog ate my homework!'
Free_Lun, lun

OpenW, lun, 'test_ascii_file_1.txt'
FOR j=0,N_Elements(strings)-1 DO PrintF, lun, strings[j]
Free_Lun, lun
Print, 'Time to rewrite file: ', systime(1) - start

END
;---------------------------------------------------
```

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

## Subject: Re: Write to a certain line of a file
Posted by Mark Piper on Fri, 20 Jan 2012 16:21:24 GMT

View Forum Message <> Reply to Message

On 1/19/2012 10:49 AM, Michael Galloy wrote:
>
> You are essentially treating the file as a binary file where you have
> more options, but also more responsibility for keeping things straight.
>
> Sure, this would work in the case where the new output has the exact
> same length in bytes as the old output, but if not everything is going
> to get screwed up. You could rewrite the file after the point of
> insertion to fix this, but how much is it worth it to get around writing
> this file?
>
>
> Mike

Ah, nuts! Yeah, I understand. I guess it's moot anyway because, as David has shown, it's trivial to read/replace/rewrite a text file.

mp