
Subject: Re: Lots of files

Posted by [Paul Van Delst\[1\]](#) on Fri, 16 Mar 2007 20:07:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Fanning wrote:

> Lasse Clausen writes:

>

>> I would like to split the one file into 31. In fact, I would like to
>> open all 32 files, loop through the big file and put the data
>> according to the index into the small file. However, IDL only lets me
>> open 28 files at a time, right?

>> IDL Help for Get_lun says: The file unit number obtained is in the
>> range 100 to 128.

>> So I end up opening and closing the according small file during each
>> loop which works great, however it is excruciatingly slow due to all
>> the waiting for the hard disk.

>>

>> Ah, and the number of stations varies, sometimes its 21, sometimes 29,
>> most of the time 30, I never know what it is going to be. Therefore, I
>> can only use the point_lun procedure to skip from data to data
>> belonging to one station if I parse the stations first. Which I could
>> do, but maybe one of you has a better idea? Any thoughts?

>

> I would use the pool of LUNs from 1 to 99, rather than the

> pool IDL access with GET_LUN from 100 to 128.

>

> OPENW, 1, ...

> OPENW, 2, ...

> ...

> OPENW, 99, ...

I know you didn't intend to suggest hardwiring 99 different fileid's :o) so maybe what you meant was something more like

```
nfiles = 31
```

```
fid_child = LINDGEN(nfiles)+1
```

```
fid_parent = nfiles+1
```

```
OPEN, fid_parent, fname_parent, .....
```

```
FOR i=0,nfiles-1 DO OPEN, fid_child[i], fname_child[i], ...
```

and then redistribute the parent file data into the child files as required.

?

cheersouzo,

paulv

--

Paul van Delst Ride lots.
CIMSS @ NOAA/NCEP/EMC

Eddy Merckx

Subject: Re: Lots of files

Posted by [David Fanning](#) on Fri, 16 Mar 2007 20:53:51 GMT

[View Forum Message](#) <> [Reply to Message](#)

Lasse Clausen writes:

- > I would like to split the one file into 31. In fact, I would like to
- > open all 32 files, loop through the big file and put the data
- > according to the index into the small file. However, IDL only lets me
- > open 28 files at a time, right?
- > IDL Help for Get_lun says: The file unit number obtained is in the
- > range 100 to 128.
- > So I end up opening and closing the according small file during each
- > loop which works great, however it is excruciatingly slow due to all
- > the waiting for the hard disk.
- >
- > Ah, and the number of stations varies, sometimes its 21, sometimes 29,
- > most of the time 30, I never know what it is going to be. Therefore, I
- > can only use the point_lun procedure to skip from data to data
- > belonging to one station if I parse the stations first. Which I could
- > do, but maybe one of you has a better idea? Any thoughts?

I would use the pool of LUNs from 1 to 99, rather than the
pool IDL access with GET_LUN from 100 to 128.

```
OPENW, 1, ...  
OPENW, 2, ...  
...  
OPENW, 99, ...
```

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Lots of files

Posted by [David Fanning](#) on Fri, 16 Mar 2007 21:23:37 GMT

[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

> I know you didn't intend to suggest hardwiring 99 different fileid's :o)

With Cut and Paste it's not so bad. Of course, you
spend the next five hours fixing typos, but... :-)

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Lots of files

Posted by [lasse](#) on Sat, 17 Mar 2007 17:32:14 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 16 Mar, 21:23, David Fanning <n...@dfanning.com> wrote:

> Paul van Delst writes:

>> I know you didn't intend to suggest hardwiring 99 different fileid's :o)

>

> With Cut and Paste it's not so bad. Of course, you

> spend the next five hours fixing typos, but... :-)

>

> Cheers,

>

> David

>

> --

> David Fanning, Ph.D.

> Fanning Software Consulting, Inc.

> Coyote's Guide to IDL Programming:<http://www.dfanning.com/>

> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Well thanks, that works, however it did not bring the speed boost I had hoped for. So I had another thought: Actually, all data is one line, not in one line per station as I said earlier. But I know that each data set is 1440 characters long, so here is the outline of my code, after I opened all the files:

```

info = file_info(input_filename)
lines = info.size/1440L

for i=0L, lines-1L do begin
  point_lun, fin, i*1440L
  readf, fin, line, format='(A1440)'
  ; extracting station name
  hstat = strlowercase(strmid(line, 12, 3))
  ; find correct file unit
  tmp = where(stats eq hstat)
  printf, tmp[0]+1, line
endfor

```

I chose the above solution because my favoured one:

```

while not.eof(fin)) do begin
  readf, fin, line, format='(A1440)'
  hstat = strlowercase(strmid(line, 12, 3))
  tmp = where(stats eq hstat)
  printf, tmp[0]+1, line
endwhile

```

does not seem to work in the way I expected, i.e. read 1440 byte, parse station, write data, read next 1440 byte... until end of file. Rather, it reads the first 1440 bytes and then hits the end of the file (while loop is executed once). So that is why I wondered what the readf command with the above format code actually does. Since it hits the end of file border after the first read command, I suspect it actually reads in all data, and then extracts the first 1440 bytes from that. Which would explain why the solution I am running now (with the for loop) is so slow: about 20 seconds for 3000 lines (4MB file). On some chunky Sun server, mind you. Any more ideas?

Cheers
Lasse

Subject: Re: Lots of files
 Posted by [Foldy Lajos](#) on Sat, 17 Mar 2007 20:42:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Sat, 17 Mar 2007, Lasse Clausen wrote:

```

> On 16 Mar, 21:23, David Fanning <n...@dfanning.com> wrote:
>> Paul van Delst writes:
>>> I know you didn't intend to suggest hardwiring 99 different fileid's :o)
>>
>> With Cut and Paste it's not so bad. Of course, you

```

```

>> spend the next five hours fixing typos, but... :-)
>>
>> Cheers,
>>
>> David
>>
>> --
>> David Fanning, Ph.D.
>> Fanning Software Consulting, Inc.
>> Coyote's Guide to IDL Programming:http://www.dfanning.com/
>> Sepore ma de ni thui. ("Perhaps thou speakest truth.")
>
> Well thanks, that works, however it did not bring the speed boost I
> had hoped for. So I had another thought: Actually, all data is one
> line, not in one line per station as I said earlier. But I know that
> each data set is 1440 characters long, so here is the outline of my
> code, after I opened all the files:
>
> info = file_info(input_filename)
> lines = info.size/1440L
>
> for i=0L, lines-1L do begin
>   point_lun, fin, i*1440L
>   readf, fin, line, format='(A1440)'
>   ; extracting station name
>   hstat = strlowercase(strmid(line, 12, 3))
>   ; find correct file unit
>   tmp = where(stats eq hstat)
>   printf, tmp[0]+1, line
> endfor
>
> I chose the above solution because my favoured one:
>
> while not eof(fin) do begin
>   readf, fin, line, format='(A1440)'
>   hstat = strlowercase(strmid(line, 12, 3))
>   tmp = where(stats eq hstat)
>   printf, tmp[0]+1, line
> endwhile
>
> does not seem to work in the way I expected, i.e. read 1440 byte,
> parse station, write data, read next 1440 byte... until end of file.
> Rather, it reads the first 1440 bytes and then hits the end of the
> file (while loop is executed once). So that is why I wondered what the
> readf command with the above format code actually does. Since it hits
> the end of file border after the first read command, I suspect it
> actually reads in all data, and then extracts the first 1440 bytes
> from that. Which would explain why the solution I am running now (with

```

> the for loop) is so slow: about 20 seconds for 3000 lines (4MB file).
> On some chunky Sun server, mind you. Any more ideas?
>
> Cheers
> Lasse
>
>

what about something like this:

```
finarr=assoc(fin, bytarr(1440))

for nrec=0l,lines-1l do begin
    line=finarr[nrec]
    hstat = strlowercase(line[12:14])
    tmp = where(stats eq hstat)
    writeu, tmp[0]+1, line
endfor
```

regards,
lajos

Subject: Re: Lots of files
Posted by [David Fanning](#) on Sun, 18 Mar 2007 03:59:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

Lasse Clausen writes:

> Well thanks, that works, however it did not bring the speed boost I
> had hoped for. So I had another thought: Actually, all data is one
> line, not in one line per station as I said earlier. But I know that
> each data set is 1440 characters long, so here is the outline of my
> code, after I opened all the files:
>
> info = file_info(input_filename)
> lines = info.size/1440L
>
> for i=0L, lines-1L do begin
> point_lun, fin, i*1440L
> readf, fin, line, format='(A1440)'
> ; extracting station name
> hstat = strlowercase(strmid(line, 12, 3))
> ; find correct file unit
> tmp = where(stats eq hstat)

```
>   printf, tmp[0]+1, line
> endfor
```

Well lots of string processing and WHERE's going on here, which I think is what is slowing things down. How about something like this:

```
theLines = Assoc(lun, BytArr(1440))
maxYear = Max(stats)
for l=0L, lines-1L do begin
    aLine = theLines[l]
    ; extracting station name
    hstat = String(aLine[12:15])
    ; find correct file unit
    printf, (maxYear-hstat)+1, String(aLine)
endfor
```

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Sepore ma de ni thui. ("Perhaps thou speakest truth.")
