Subject: Re: idl_idlbridge weirdness on unix systems
Posted by David Fanning on Tue, 28 Feb 2012 17:59:00 GMT
View Forum Message <> Reply to Message

Yngvar Larsen writes:

> Anyone else with similar experience?

No problems on my Windows 7 64-bit machine
in either IDL 7.1.2 or IDL 8.1.

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: idl_idlbridge weirdness on unix systems
Posted by Yngvar Larsen on Tue, 28 Feb 2012 18:05:44 GMT
View Forum Message <> Reply to Message

On Feb 28, 6:59 pm, David Fanning <n...@idlcoyote.com> wrote:
> Yngvar Larsen writes:
>> Anyone else with similar experience?
>
> No problems on my Windows 7 64-bit machine
> in either IDL 7.1.2 or IDL 8.1.

OK, thanks. Then it is definitely a unix-only problem as I suspected.

--
Yngvar

Subject: Re: idl_idlbridge weirdness on unix systems
Posted by Yngvar Larsen on Tue, 28 Feb 2012 19:08:44 GMT
View Forum Message <> Reply to Message

On Feb 28, 6:49 pm, Yngvar Larsen <larsen.yng...@gmail.com> wrote:
> I encountered a strange problem today while working with IDL child
> processes instantiated using the IDL_IDLBridge class. Specifically, I

> want to be able to use a lot of subprocesses on a 4 CPU x 12-kernel
> server, so I tried using 48 IDL_IDLBridge objects to test the system.
> But no cigar. IDL hangs apparently at random. I narrowed down the
> problem to the following.
>
> The simple code below just instantiates an array of objects, and then
> tries to destroy them one by one. What happens (for me) is that when
> it encounters index ii=16, the entire IDL process hangs.
>
> N = 17
> b = objarr(N)
> for ii=0, N-1 do b[ii]=obj_new('idl_idlbridge')
>
> for ii=0, N-1 do begin
> status = b[ii]->status() ; Just to make sure the object is idle
> (status = 0)
> print, ii, status
> obj_destroy, b[ii]
> endfor
>
> Now comes the weird part; the following almost identical code, where I
> just destroy the objects in the opposite order, works fine always!
>
> N = 17
> b = objarr(N)
> for ii=0, N-1 do b[ii]=obj_new('idl_idlbridge')
>
> for ii=N-1,0,-1 do begin
> status = b[ii]->status() ; Just to make sure the object is idle
> (status = 0)
> print, ii, status
> obj_destroy, b[ii]
> endfor

Errata: it fails for ii=15, not 16.

Just to add to the weirdness: I tested this also for larger N. It
turns out it does not matter which order the objects b[0:14] are
destroyed.

But for any N>15, you must kill the objects in descending order:

for ii=N-1,15,-1 do obj_destroy, b[ii]

You can also kill any b[ii] for ii<15 at anytime, as long as the high
index objects are destroyed in descending order, e.g. this is ok:

N = 25

```
b = objarr(N)
for ii=0, N-1 do b[ii]=obj_new('idl_idlbridge')

p =  [5,24,12,7,23,14,3,22,21,4,8,9,2,20,10,19,18,11,17,1,6,16,0, 13,15]
for ii=0,N-1 do obj_destroy, b[p[ii]]
```

My solution is to wait for all child processes to finish, and then destroying the idl_idlbridge objects in descending index order. Destroying the objects whenever they finish their respective part of the full computation will not work. (Unless the child processes by chance finish in descending order by array index for indices larger than 14...)

Over and out. Thank you, Exelis. (I want half a day of my life back.)

--
Yngvar

Subject: Re: idl_idlbridge weirdness on unix systems
Posted by David Fanning on Tue, 28 Feb 2012 19:23:52 GMT
View Forum Message <> Reply to Message

Yngvar Larsen writes:

> Over and out. Thank you, Exelis. (I want half a day of my life back.)

Yeah, I don't think you get it. You are expected to fix their software AND pay for the privilege of doing so! Start saving your maintenance dollars now, and pray they get around to fixing this. I'm pretty sure it won't qualify for the "sexy" list, but it will probably be put in front of any problems having to do with traditional graphics commands! ;-)

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: idl_idlbridge weirdness on unix systems
Posted by Mark Piper on Tue, 28 Feb 2012 21:34:49 GMT
View Forum Message <> Reply to Message

On 2/28/2012 10:49 AM, Yngvar Larsen wrote:
> I encountered a strange problem today while working with IDL child
> processes instantiated using the IDL_IDLBridge class. Specifically, I
> want to be able to use a lot of subprocesses on a 4 CPU x 12-kernel
> server, so I tried using 48 IDL_IDLBridge objects to test the system.
> But no cigar. IDL hangs apparently at random. I narrowed down the
> problem to the following.
>
> The simple code below just instantiates an array of objects, and then
> tries to destroy them one by one. What happens (for me) is that when
> it encounters index ii=16, the entire IDL process hangs.
>
> N = 17
> b = objarr(N)
> for ii=0, N-1 do b[ii]=obj_new('idl_idlbridge')
>
> for ii=0, N-1 do begin
> status = b[ii]->status() ; Just to make sure the object is idle
> (status = 0)
> print, ii, status
> obj_destroy, b[ii]
> endfor
>
> Now comes the weird part; the following almost identical code, where I
> just destroy the objects in the opposite order, works fine always!
>
> N = 17
> b = objarr(N)
> for ii=0, N-1 do b[ii]=obj_new('idl_idlbridge')
>
> for ii=N-1,0,-1 do begin
> status = b[ii]->status() ; Just to make sure the object is idle
> (status = 0)
> print, ii, status
> obj_destroy, b[ii]
> endfor
>
> I have tried this with identical results on the following systems:
> IDL 7.0.8, Linux 2.6.20 (old Fedora system)
> IDL 8.1, Linux 2.6.32 (new Red Hat system)
> IDL 8.1 Linux 2.6.32 (Ubuntu 10.04 LTS)
> IDL 7.1.1, Mac OS X 10.6.8
>
> However, I tested it on the following system with no problems:
> IDL 6.4, Windows XP

>
> Anyone else with similar experience?
>
> --
> Yngvar

Hi Yngvar,

There's currently an open CR on this (64611, for your reference); I'll add you to the report. Tech Support has also identified a possible workaround. Please contact them if you're interested.

mp

Subject: Re: idl_idlbridge weirdness on unix systems
Posted by David Fanning on Tue, 28 Feb 2012 21:39:21 GMT
View Forum Message <> Reply to Message

Mark Piper writes:

> There's currently an open CR on this (64611, for your reference); I'll
> add you to the report. Tech Support has also identified a possible
> workaround. Please contact them if you're interested.

Mark maybe you can solve a mystery for us. You say, "for your reference", but most of us have not found any way to look up, say 64611, on the Exceles web page. Is there a way to do this? Can we find out what bugs have been reported, etc.? Thanks!

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: idl_idlbridge weirdness on unix systems
Posted by Yngvar Larsen on Wed, 29 Feb 2012 17:03:53 GMT
View Forum Message <> Reply to Message

On Feb 28, 10:34 pm, Mark Piper <mpi...@ittvis.com> wrote:
> On 2/28/2012 10:49 AM, Yngvar Larsen wrote:
>
>
>
>
>
>
>
>
>
>
>> I encountered a strange problem today while working with IDL child
>> processes instantiated using the IDL_IDLBridge class. Specifically, I
>> want to be able to use a lot of subprocesses on a 4 CPU x 12-kernel
>> server, so I tried using 48 IDL_IDLBridge objects to test the system.
>> But no cigar. IDL hangs apparently at random. I narrowed down the
>> problem to the following.
>
>> The simple code below just instantiates an array of objects, and then
>> tries to destroy them one by one. What happens (for me) is that when
>> it encounters index ii=16, the entire IDL process hangs.
>
>> N = 17
>> b = objarr(N)
>> for ii=0, N-1 do b[ii]=obj_new('idl_idlbridge')
>
>> for ii=0, N-1 do begin
>> status = b[ii]->status() ; Just to make sure the object is idle
>> (status = 0)
>> print, ii, status
>> obj_destroy, b[ii]
>> endfor
>
>> Now comes the weird part; the following almost identical code, where I
>> just destroy the objects in the opposite order, works fine always!
>
>> N = 17
>> b = objarr(N)
>> for ii=0, N-1 do b[ii]=obj_new('idl_idlbridge')
>
>> for ii=N-1,0,-1 do begin
>> status = b[ii]->status() ; Just to make sure the object is idle
>> (status = 0)
>> print, ii, status
>> obj_destroy, b[ii]
>> endfor
>
>> I have tried this with identical results on the following systems:

>> IDL 7.0.8, Linux 2.6.20 (old Fedora system)
>> IDL 8.1, Linux 2.6.32 (new Red Hat system)
>> IDL 8.1 Linux 2.6.32 (Ubuntu 10.04 LTS)
>> IDL 7.1.1, Mac OS X 10.6.8
>
>> However, I tested it on the following system with no problems:
>> IDL 6.4, Windows XP
>
>> Anyone else with similar experience?
>
>> --
>> Yngvar
>
> Hi Yngvar,
>
> There's currently an open CR on this (64611, for your reference); I'll
> add you to the report. Tech Support has also identified a possible
> workaround. Please contact them if you're interested.
>
> mp

Thanks.

My own workaround (destroying the objects in the opposite order of how they were created) seems to work reliably on my system, so I'm fine for the time being. If I run into more problems, I'll contact tech support before wasting more time.

Two questions:
1) Like David, I would like to know where to look up/track known issues (if such a place exists). Could be a time saver sometimes...

2) Is there a standard way to submit bugs/feature requests? I have one rather important issue that is part bug, part feature request: Decent TIFF/BigTIFF-support. In order to stay relevant for the remote sensing community, this is a must. A lot of satellite data are delivered in (Geo)TIFF format, and the typical size of some satellite products is approaching or surpassing the 4-GB limit for the TIFF format.

- Bug: READ_TIFF (at least on unix) only supports TIFF files up to 2 GB even though the TIFF standard is defined up to 4 GB. I suspect a signed 32-bit integer is used internally instead of an unsigned 32-bit integer to represent the file offsets (?), wasting one bit.

- Feature request: BigTIFF support. My research group (in an earth observation department) at work already had to implement a rudimentary BigTIFF-reader for some very important data.

--
Yngvar