Subject: G4-related problem
Posted by on Mon, 11 Aug 2003 12:47:31 GMT

View Forum Message <> Reply to Message

Hi.

I recently discovered a very nasty problem on my G4 (2*867, Mac OS X 10.2.6) running IDL 5.6.

The following code should not produce any output, if everything works. Unfortunately, this is not the case on my G4, where the variable assignment "data1=data0" produces messy data in a randomly fashion. So it works most of the time, but once in a while, the data in "data1" will be corrupted. Typically, there are some zeros showing up in the array and some other garbage.

Here is the code in question:

```
data0=fltarr(10000)+1.
for ii=0L, 200000 do begin & $
   data1=data0 & $
   if min(where(data1 eq 0)) gt 0 then begin & $
     print, ii & ii=200001L & endif & $
endfor
```

This problem did not show up on other machines I have tested (Compaq Alpha, a Windows PC or my iMac G3 running the same OS and IDL versions).

So this can either be a G4-specific bug in IDL 5.6, a G4-specific bug in Mac OS X 10.2.6, or a hardware problem on my G4.

Since I do not have a second G4 available, I would like to ask anyone with a G4 to test the code snippet given above. You can simply paste the code to the command line in IDL.

Thank you,
Rudi

PS: What I already tested:
- setting TPOOL_NTHREADS=1 and VECTOR_ENABLE=0
- using different RAM
- replacing the RAM modules
- testing the RAM with the hardware test CD

Subject: Re: G4-related problem

Posted by Kenneth P. Bowman on Mon, 11 Aug 2003 15:40:24 GMT

View Forum Message <> Reply to Message

In article <bh8393$2as$1@surz18.uni-marburg.de>,
Rudi Schafer <rudi.schaefer@physik.uni-marburg.de> wrote:

> Hi.
>
> I recently discovered a very nasty problem on my G4 (2*867, Mac OS X
> 10.2.6) running IDL 5.6.

There is a known problem with IDL running on Mac OS X 10.2. It appears as memory corruption when working with large arrays.

When I get to the office I'll try to post some additional information.

RSI claims that the problem disappears under OS X 10.3, suggesting that it is an OS (library?) problem, but they have not pin-pointed the bug. They have suggested that it has something to do with XDarwin (X windows emulator based on XFree86), but I see the problem even when my IDL codes do no graphics and XDarwin is not running. I run eXodus on my own machine and still see the problem there.

We currently are not doing any real calculations on our Macs. This is a serious problem for us, but fortunately we do have some other architectures to crunch numbers on.

Ken Bowman

Subject: Re: G4-related problem

Posted by MKatz843 on Mon, 11 Aug 2003 18:38:17 GMT

View Forum Message <> Reply to Message

Not only can I reproduce this problem on my G4, running IDL 5.6 in OS X 10.2.6, I can add some additional information that's quite distressing.

I modified your code to give a little more output:

```
data0=fltarr(10000)+1.
for ii=0L, 10000L do begin $
  data1=data0
    ;--- look for 0s
  w0 = where(data1 EQ 0., count0)
  if min(w0) GT 0 then begin
```

```
    print, 'ii, count0 = ', ii, count0
    print, '       w0 = ', w0, FORMAT='(a,99i9)'
    print, ' data1(w0) = ', data1(w0), FORMAT='(a,99f9.4)'
    print, ' '
  endif
    ;--- look for not (1 or 0)
  w1 = where((data1 NE 1.) AND (data1 NE 0.), count1)
  if min(w1) GT 0 then begin
    print, 'ii, count1 = ', ii, count1
    print, '       w1 = ', w1, FORMAT='(a,99i9)'
    print, ' data1(w1) = ', data1(w1), FORMAT='(a,99f9.4)'
    print, ' '
  endif
end
```

On a G4 733, the output was different every time the program was run, but there is a striking predictability to the problem.

Here's the output of one test run

```
ii, count =        9187          8
       w0 =     4076     4077     4078     4079     4080     4081
4082     4083
data1(w0) =    0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
0.0000   0.0000

ii, count =       11494          8
       w0 =     8172     8173     8174     8175     8176     8177
8178     8179
data1(w0) =    0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
0.0000   0.0000

ii, count =       13739          8
       w0 =     4076     4077     4078     4079     4080     4081
4082     4083
data1(w0) =    0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
0.0000   0.0000

ii, count =       18309          8
       w0 =     2028     2029     2030     2031     2032     2033
2034     2035
data1(w0) =    0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
0.0000   0.0000

ii, count =       25220          8
       w0 =     6124     6125     6126     6127     6128     6129
6130     6131
data1(w0) =    0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
```

```
0.0000   0.0000

ii, count =       60714          8
      w0 =     8172     8173     8174     8175     8176     8177
8178     8179
data1(w0) =    0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
0.0000   0.0000

ii, count =       66317          8
      w0 =     1004     1005     1006     1007     1008     1009
1010     1011
data1(w0) =    0.0000   0.0000   0.0000   0.0000   0.0000   0.0000
0.0000   0.0000
```

I noticed that there were usually strings of 8 zeros in a row (yet
sometimes just 2),
and that they corresponded most frequently to array index values 1004,
2028, 4076, 6124, 8172.
Those values happen to be 2^n - 20 where n=10,11,12, and 13. 6124 =
2^12 + 2^11 - 20.

The case where the numbers are neither zero nor one did not come up.

As designed, the test should produce no output whatsoever. On a P4
Linux machine I tested, there was no output.

This is terrible. Now what? I think the next step is to contact RSI
and see if they can duplicate the problem internally.

M. Katz

Rudi Schï¿½fer <rudi.schaefer@physik.uni-marburg.de> wrote in message
news:<bh8393$2as$1@surz18.uni-marburg.de>...

> I recently discovered a very nasty problem on my G4 (2*867, Mac OS X
> 10.2.6) running IDL 5.6.
>
> The following code should not produce any output, if everything works.
> Unfortunately, this is not the case on my G4, where the variable
> assignment "data1=data0" produces messy data in a randomly fashion. So
> it works most of the time, but once in a while, the data in "data1" will
> be corrupted. Typically, there are some zeros showing up in the array
> and some other garbage.
>
> Here is the code in question:
>

```
> data0=fltarr(10000)+1.
> for ii=0L, 200000 do begin & $
>    data1=data0 & $
>    if min(where(data1 eq 0)) gt 0 then begin & $
>      print, ii & ii=200001L & endif & $
> endfor
```

>
> This problem did not show up on other machines I have tested (Compaq
> Alpha, a Windows PC or my iMac G3 running the same OS and IDL versions).
>
> So this can either be a G4-specific bug in IDL 5.6, a G4-specific bug in
>   Mac OS X 10.2.6, or a hardware problem on my G4.
>
> Since I do not have a second G4 available, I would like to ask anyone
> with a G4 to test the code snippet given above. You can simply paste the
>   code to the command line in IDL.
>
> Thank you,
> Rudi

Subject: Re: G4-related problem
Posted by MKatz843 on Mon, 11 Aug 2003 19:24:09 GMT

View Forum Message <> Reply to Message

A little more information.

The problem also occurs with double-precision and long data types,
although the array-indices of the problem points are different in the
double case. Float matches long so I wonder if it's a similarity in
4-byte data types.

M. Katz

Subject: Re: G4-related problem
Posted by Jean Koclas on Mon, 11 Aug 2003 22:02:02 GMT

View Forum Message <> Reply to Message

My lowly iBook G3 600Mhz OSX 10.2.6 with Apple X11 beta 3 passes the
test...

Could the problem be related to dual processors? You get me worried, as
I am thinking of upgrading to a G4 portable.

--
Jean Koclas

Ecole Polytechnique de Montreal

In article <bh8393$2as$1@surz18.uni-marburg.de>,
Rudi Schafer <rudi.schaefer@physik.uni-marburg.de> wrote:

> Hi.
>
> I recently discovered a very nasty problem on my G4 (2*867, Mac OS X 10.2.6) running IDL 5.6.
>
> The following code should not produce any output, if everything works. Unfortunately, this is not the case on my G4, where the variable assignment "data1=data0" produces messy data in a randomly fashion. So it works most of the time, but once in a while, the data in "data1" will be corrupted. Typically, there are some zeros showing up in the array and some other garbage.
>
> Here is the code in question:
>
> ```
> data0=fltarr(10000)+1.
> for ii=0L, 200000 do begin & $
>    data1=data0 & $
>    if min(where(data1 eq 0)) gt 0 then begin & $
>      print, ii & ii=200001L & endif & $
> endfor
> ```
>
> This problem did not show up on other machines I have tested (Compaq Alpha, a Windows PC or my iMac G3 running the same OS and IDL versions).
>
> So this can either be a G4-specific bug in IDL 5.6, a G4-specific bug in Mac OS X 10.2.6, or a hardware problem on my G4.
>
> Since I do not have a second G4 available, I would like to ask anyone with a G4 to test the code snippet given above. You can simply paste the code to the command line in IDL.
>
> Thank you,
> Rudi
>
> PS: What I already tested:
> - setting TPOOL_NTHREADS=1 and VECTOR_ENABLE=0
> - using different RAM
> - replacing the RAM modules

> - testing the RAM with the hardware test CD

Subject: Re: G4-related problem
Posted by Karl Schultz on Fri, 15 Aug 2003 17:50:27 GMT
View Forum Message <> Reply to Message

news:bh8393$2as$1@surz18.uni-marburg.de...
> Hi.
>
> I recently discovered a very nasty problem on my G4 (2*867, Mac OS X
> 10.2.6) running IDL 5.6.
>
> The following code should not produce any output, if everything works.
> Unfortunately, this is not the case on my G4, where the variable
> assignment "data1=data0" produces messy data in a randomly fashion. So
> it works most of the time, but once in a while, the data in "data1" will
> be corrupted. Typically, there are some zeros showing up in the array
> and some other garbage.
>
> Here is the code in question:
>
```
> data0=fltarr(10000)+1.
> for ii=0L, 200000 do begin & $
>    data1=data0 & $
>    if min(where(data1 eq 0)) gt 0 then begin & $
>      print, ii & ii=200001L & endif & $
> endfor
```
>
>
>
> This problem did not show up on other machines I have tested (Compaq
> Alpha, a Windows PC or my iMac G3 running the same OS and IDL versions).
>
> So this can either be a G4-specific bug in IDL 5.6, a G4-specific bug in
>   Mac OS X 10.2.6, or a hardware problem on my G4.

Aside from the Altivec code, there is no G4-specific code in IDL. Since the
Altivec is not used for simple copy operations and you disabled it, that
can't be the problem. (But good thinking on disabling it)

> Since I do not have a second G4 available, I would like to ask anyone
> with a G4 to test the code snippet given above. You can simply paste the
>   code to the command line in IDL.

> PS: What I already tested:

> - setting TPOOL_NTHREADS=1 and VECTOR_ENABLE=0
> - using different RAM
> - replacing the RAM modules
> - testing the RAM with the hardware test CD

I've spent a bunch of time on this and I think I can help a little here.

- The problem seems to only occur on OS X 10.2. I don't know if the story changes depending on the version of 10.2 (e.g., 10.2.3). The problem does not seem to occur on 10.1 and the developer's preview of 10.3. This last it em is good news.
- I've only seen or heard the problem reported on G4.
- The flavor of the X11 server involved does not matter. The IDL program needs to be connected to *any* X server to demonstrate the problem. That is, the issue is all on the client side.

Given those things, the problem is on OS X 10.2 G4's while IDL has an X connection open.

On some systems (OS X is one), IDL uses a 1-second interval timer proc that runs whenever there is an X connection. Its job is to keep your graphics windows reasonably up to date while IDL is performing some long operation, between interpreter instructions. Disabling this timer operation keeps the problem from occuring, but at the (minor?) expense of unresponsive windows during these operations.

So, possible solutions are:
1) Use 10.1 or wait for 10.3.
2) Issue (the undocumented) DEVICE, /NOTIMER to turn off the timer proc. You'll have to judge if this is ok for your app or not.
3) Make sure your program does not do any graphics. Note that if you have some DEVICE command in your startup file to configure your windows, this still makes a connection, even if you don't open any graphic or UI windows.

If you aren't sure if your machine suffers from the problem, I'd suggest running the program above.

Hope this helps,
Karl

Subject: Re: G4-related problem
Posted by Karl Schultz on Wed, 03 Sep 2003 16:26:27 GMT

View Forum Message <> Reply to Message

Update:

We heard back from Apple and they were able to find the problem via a bug report I filed.

They said that the problem does exist on 10.2 and also in 10.3. In the 10.3 case, another issue hid the actual bug, which led us to believe that the problem didn't exist in 10.3. In any case, the problem will not appear in future OS X releases. The fix has been rolled into the current OS X 10.3 builds. It is unclear if there is going to be another 10.2.x release and if this fix will be in it. (If there is another 10.2.x (10.2.7 ?), look for a mention of this in the "fixed issues" list when you install the update).

Adding together what I observed and what they told us, I believe that the cause of the problem was Altivec state not being handled correctly when the OS X kernel interrupts the main thread to run the timer signal handler thread. OS X uses the Altivec to implement the bcopy function, and so any changes to Altivec state that may occur in the signal handler would cause an interrupted bcopy call to malfunction.

Anyway, the problem is understood and the workarounds I suggested below are still valid.

Karl

"Karl Schultz" <kschultz_no_spam@rsinc.com> wrote in message
news:vjq75m7imf90a2@corp.supernews.com...
>

```
>  news:bh8393$2as$1@surz18.uni-marburg.de...
>>  Hi.
>>
>>  I recently discovered a very nasty problem on my G4 (2*867, Mac OS X
>>  10.2.6) running IDL 5.6.
>>
>>  The following code should not produce any output, if everything works.
>>  Unfortunately, this is not the case on my G4, where the variable
>>  assignment "data1=data0" produces messy data in a randomly fashion. So
>>  it works most of the time, but once in a while, the data in "data1" will
>>  be corrupted. Typically, there are some zeros showing up in the array
>>  and some other garbage.
>>
>>  Here is the code in question:
>>
>>  data0=fltarr(10000)+1.
>>  for ii=0L, 200000 do begin & $
>>     data1=data0 & $
>>     if min(where(data1 eq 0)) gt 0 then begin & $
>>       print, ii & ii=200001L & endif & $
>>  endfor
>>
```

>>
>>
>> This problem did not show up on other machines I have tested (Compaq
>> Alpha, a Windows PC or my iMac G3 running the same OS and IDL versions).
>>
>> So this can either be a G4-specific bug in IDL 5.6, a G4-specific bug in
>>   Mac OS X 10.2.6, or a hardware problem on my G4.
>
> Aside from the Altivec code, there is no G4-specific code in IDL. Since
the
> Altivec is not used for simple copy operations and you disabled it, that
> can't be the problem. (But good thinking on disabling it)
>
>> Since I do not have a second G4 available, I would like to ask anyone
>> with a G4 to test the code snippet given above. You can simply paste the
>>   code to the command line in IDL.
>
>
>> PS: What I already tested:
>> - setting TPOOL_NTHREADS=1 and VECTOR_ENABLE=0
>> - using different RAM
>> - replacing the RAM modules
>> - testing the RAM with the hardware test CD
>
> I've spent a bunch of time on this and I think I can help a little here.
>
> - The problem seems to only occur on OS X 10.2. I don't know if the story
> changes depending on the version of 10.2 (e.g., 10.2.3). The problem does
> not seem to occur on 10.1 and the developer's preview of 10.3. This last
it
> em is good news.
> - I've only seen or heard the problem reported on G4.
> - The flavor of the X11 server involved does not matter. The IDL program
> needs to be connected to *any* X server to demonstrate the problem. That
> is, the issue is all on the client side.
>
> Given those things, the problem is on OS X 10.2 G4's while IDL has an X
> connection open.
>
> On some systems (OS X is one), IDL uses a 1-second interval timer proc
that
> runs whenever there is an X connection. Its job is to keep your graphics
> windows reasonably up to date while IDL is performing some long operation,
> between interpreter instructions. Disabling this timer operation keeps
the
> problem from occuring, but at the (minor?) expense of unresponsive windows
> during these operations.
>

> So, possible solutions are:
> 1) Use 10.1 or wait for 10.3.
> 2) Issue (the undocumented) DEVICE, /NOTIMER to turn off the timer proc.
> You'll have to judge if this is ok for your app or not.
> 3) Make sure your program does not do any graphics. Note that if you have
> some DEVICE command in your startup file to configure your windows, this
> still makes a connection, even if you don't open any graphic or UI
windows.
>
> If you aren't sure if your machine suffers from the problem, I'd suggest
> running the program above.
>
> Hope this helps,
> Karl
>
>
>

Subject: Re: G4-related problem
Posted by K. Bowman on Wed, 03 Sep 2003 16:45:14 GMT

View Forum Message <> Reply to Message

In article <vlc5b17e31l6f8@corp.supernews.com>,
"Karl Schultz" <kschultz_no_spam@rsinc.com> wrote:

Karl,

Thanks for persisting with this and helping track down the problem. If you get any more info from Apple about whether the fix will be included in 10.2.? (if there is one), could you please post it? The info Apple releases about updates is scanty and unlikely to include this level of technical detail.

Thanks, Ken Bowman

> Update:
>
> We heard back from Apple and they were able to find the problem via a bug
> report I filed.
>
> They said that the problem does exist on 10.2 and also in 10.3. In the 10.3
> case, another issue hid the actual bug, which led us to believe that the
> problem didn't exist in 10.3. In any case, the problem will not appear in
> future OS X releases. The fix has been rolled into the current OS X 10.3
> builds. It is unclear if there is going to be another 10.2.x release and if

> this fix will be in it. (If there is another 10.2.x (10.2.7 ?), look for a
> mention of this in the "fixed issues" list when you install the update).
>
> Adding together what I observed and what they told us, I believe that the
> cause of the problem was Altivec state not being handled correctly when the
> OS X kernel interrupts the main thread to run the timer signal handler
> thread. OS X uses the Altivec to implement the bcopy function, and so any
> changes to Altivec state that may occur in the signal handler would cause an
> interrupted bcopy call to malfunction.
>
> Anyway, the problem is understood and the workarounds I suggested below are
> still valid.
>
> Karl

Subject: Re: G4-related problem
Posted by DPJ on Sat, 04 Oct 2003 13:01:15 GMT

View Forum Message <> Reply to Message

Hi all,

I am a very new IDL user (novice) and I think I am running into this G4 problem that people have been discussing. I am very disappointed since the very first thing I tried to do with IDL and it seemed to result in utter nonsense. I am working on a dual processor G4 running OSX 10.2.(6?)

Here is what I noticed.

I have an ascii file with two columns of tab delimited floating point data (x and y). I have no trouble reading in this data and making a plot of it. This seems to be true whether there is 5000 lines of data or 50,000 lines of data (I have not tried to rigorously test out if it would fail at some point).

However, if I try to read a file with 10 columns of data (x1 y1 x2 y2...), the following happens. The first two columns read in fine but the following columns do not. This seems to be true no matter how many lines of data there are.

They all read in fine up to some point and then there is what appears to be random integer data for a while, and then good data again. It appears as though columns 3-9 always have good data up to item 256. That is, data.x3[255] is good while data.x3[256] is bad. Column 10 appears to "go bad" at item 128.

I am really feeling screwed. Going back to OSX 10.1 is not an option. I don't think my program is doing any graphics before/while this data input is

going on (although it does plot the data after it has been read). I will
try the workaround that Karl suggested to turn off the timer and hopefully
that will work. If anyone has any additional suggestions, please let me
know. (Please remember that I am a novice if you respond.) This is
important because I have a ton of these data files to analyze.

David

"Karl Schultz" <kschultz_no_spam@rsinc.com> wrote in message
news:vlc5b17e31l6f8@corp.supernews.com...
> Update:
>
> We heard back from Apple and they were able to find the problem via a bug
> report I filed.
>
> They said that the problem does exist on 10.2 and also in 10.3. In the
10.3
> case, another issue hid the actual bug, which led us to believe that the
> problem didn't exist in 10.3. In any case, the problem will not appear in
> future OS X releases. The fix has been rolled into the current OS X 10.3
> builds. It is unclear if there is going to be another 10.2.x release and
if
> this fix will be in it. (If there is another 10.2.x (10.2.7 ?), look for
a
> mention of this in the "fixed issues" list when you install the update).
>
> Adding together what I observed and what they told us, I believe that the
> cause of the problem was Altivec state not being handled correctly when
the
> OS X kernel interrupts the main thread to run the timer signal handler
> thread. OS X uses the Altivec to implement the bcopy function, and so any
> changes to Altivec state that may occur in the signal handler would cause
an
> interrupted bcopy call to malfunction.
>
> Anyway, the problem is understood and the workarounds I suggested below
are
> still valid.
>
> Karl
>
> "Karl Schultz" <kschultz_no_spam@rsinc.com> wrote in message
> news:vjq75m7imf90a2@corp.supernews.com...
>>

>> news:bh8393$2as$1@surz18.uni-marburg.de...
>>> Hi.
>>>

>>> I recently discovered a very nasty problem on my G4 (2*867, Mac OS X
>>> 10.2.6) running IDL 5.6.
>>>
>>> The following code should not produce any output, if everything works.
>>> Unfortunately, this is not the case on my G4, where the variable
>>> assignment "data1=data0" produces messy data in a randomly fashion. So
>>> it works most of the time, but once in a while, the data in "data1"
will
>>> be corrupted. Typically, there are some zeros showing up in the array
>>> and some other garbage.
>>>
>>> Here is the code in question:
>>>
>>> data0=fltarr(10000)+1.
>>> for ii=0L, 200000 do begin & $
>>>   data1=data0 & $
>>>   if min(where(data1 eq 0)) gt 0 then begin & $
>>>   print, ii & ii=200001L & endif & $
>>> endfor
>>>
>>>
>>>
>>> This problem did not show up on other machines I have tested (Compaq
>>> Alpha, a Windows PC or my iMac G3 running the same OS and IDL
versions).
>>>
>>> So this can either be a G4-specific bug in IDL 5.6, a G4-specific bug
in
>>>   Mac OS X 10.2.6, or a hardware problem on my G4.
>>
>> Aside from the Altivec code, there is no G4-specific code in IDL. Since
> the
>> Altivec is not used for simple copy operations and you disabled it, that
>> can't be the problem. (But good thinking on disabling it)
>>
>>> Since I do not have a second G4 available, I would like to ask anyone
>>> with a G4 to test the code snippet given above. You can simply paste
the
>>>   code to the command line in IDL.
>>
>>
>>> PS: What I already tested:
>>> - setting TPOOL_NTHREADS=1 and VECTOR_ENABLE=0
>>> - using different RAM
>>> - replacing the RAM modules
>>> - testing the RAM with the hardware test CD
>>
>> I've spent a bunch of time on this and I think I can help a little here.

>>
>> - The problem seems to only occur on OS X 10.2. I don't know if the
story
>> changes depending on the version of 10.2 (e.g., 10.2.3). The problem
does
>> not seem to occur on 10.1 and the developer's preview of 10.3. This
last
> it
>> em is good news.
>> - I've only seen or heard the problem reported on G4.
>> - The flavor of the X11 server involved does not matter. The IDL
program
>> needs to be connected to *any* X server to demonstrate the problem.
That
>> is, the issue is all on the client side.
>>
>> Given those things, the problem is on OS X 10.2 G4's while IDL has an X
>> connection open.
>>
>> On some systems (OS X is one), IDL uses a 1-second interval timer proc
> that
>> runs whenever there is an X connection. Its job is to keep your
graphics
>> windows reasonably up to date while IDL is performing some long
operation,
>> between interpreter instructions. Disabling this timer operation keeps
> the
>> problem from occuring, but at the (minor?) expense of unresponsive
windows
>> during these operations.
>>
>> So, possible solutions are:
>> 1) Use 10.1 or wait for 10.3.
>> 2) Issue (the undocumented) DEVICE, /NOTIMER to turn off the timer proc.
>> You'll have to judge if this is ok for your app or not.
>> 3) Make sure your program does not do any graphics. Note that if you
have
>> some DEVICE command in your startup file to configure your windows, this
>> still makes a connection, even if you don't open any graphic or UI
> windows.
>>
>> If you aren't sure if your machine suffers from the problem, I'd suggest
>> running the program above.
>>
>> Hope this helps,
>> Karl
>>
>>

>>
>
>

Subject: Re: G4-related problem
Posted by Kenneth P. Bowman on Sat, 04 Oct 2003 16:25:28 GMT
View Forum Message <> Reply to Message

One possibility is that your ASCII file is corrupted. Another is a coding error. Can you post your code (minimum example that demostrates the problem)?

Ken Bowman

In article <vwzfb.27799$3S.8589@newsread2.news.atl.earthlink.net>,
"DPJ" <dpjackson@yahoo.com> wrote:

> I have an ascii file with two columns of tab delimited floating point data
> (x and y). I have no trouble reading in this data and making a plot of it.
> This seems to be true whether there is 5000 lines of data or 50,000 lines of
> data (I have not tried to rigorously test out if it would fail at some
> point).
>
> However, if I try to read a file with 10 columns of data (x1 y1 x2 y2...),
> the following happens. The first two columns read in fine but the following
> columns do not. This seems to be true no matter how many lines of data
> there are.
>
> They all read in fine up to some point and then there is what appears to be
> random integer data for a while, and then good data again. It appears as
> though columns 3-9 always have good data up to item 256. That is,
> data.x3[255] is good while data.x3[256] is bad. Column 10 appears to "go
> bad" at item 128.
>
> I am really feeling screwed. Going back to OSX 10.1 is not an option. I
> don't think my program is doing any graphics before/while this data input is
> going on (although it does plot the data after it has been read). I will
> try the workaround that Karl suggested to turn off the timer and hopefully
> that will work. If anyone has any additional suggestions, please let me
> know. (Please remember that I am a novice if you respond.) This is
> important because I have a ton of these data files to analyze.
>
> David
>

Subject: Re: G4-related problem
Posted by DPJ on Sun, 05 Oct 2003 14:34:18 GMT
View Forum Message <> Reply to Message

Hi Ken,

Thanks for the response. I will post the code I am using (it's only a few
lines long) and also include a data file tomorrow (Monday) when I am back at
work. I will start a new thread. I've checked the ascii file and it is
fine. It certainly *could* be a coding error (I am pretty new to this) but
the code is soooo simple (read in data and look at it), I would be surprised
if this was the problem.

David

"Kenneth P. Bowman" <kpb@null.com> wrote in message
news:kpb-4530AA.11252804102003@corp.supernews.com...
> One possibility is that your ASCII file is corrupted. Another is a
> coding error. Can you post your code (minimum example that demostrates
> the problem)?
>
> Ken Bowman
>
>
> In article <vwzfb.27799$3S.8589@newsread2.news.atl.earthlink.net>,
>   "DPJ" <dpjackson@yahoo.com> wrote:
>
>> I have an ascii file with two columns of tab delimited floating point
data
>> (x and y). I have no trouble reading in this data and making a plot of
it.
>> This seems to be true whether there is 5000 lines of data or 50,000
lines of
>> data (I have not tried to rigorously test out if it would fail at some
>> point).
>>
>> However, if I try to read a file with 10 columns of data (x1 y1 x2
y2...),
>> the following happens. The first two columns read in fine but the
following
>> columns do not. This seems to be true no matter how many lines of data
>> there are.
>>
>> They all read in fine up to some point and then there is what appears to
be
>> random integer data for a while, and then good data again. It appears
as
>> though columns 3-9 always have good data up to item 256. That is,
>> data.x3[255] is good while data.x3[256] is bad. Column 10 appears to

"go
>> bad" at item 128.
>>
>> I am really feeling screwed. Going back to OSX 10.1 is not an option.
I
>> don't think my program is doing any graphics before/while this data
input is
>> going on (although it does plot the data after it has been read). I
will
>> try the workaround that Karl suggested to turn off the timer and
hopefully
>> that will work. If anyone has any additional suggestions, please let me
>> know. (Please remember that I am a novice if you respond.) This is
>> important because I have a ton of these data files to analyze.
>>
>> David
>>

Subject: Re: G4-related problem
Posted by James Kuyper on Sun, 05 Oct 2003 16:34:24 GMT
View Forum Message <> Reply to Message

DPJ wrote:
>
> Hi Ken,
>
> Thanks for the response. I will post the code I am using (it's only a few
> lines long) and also include a data file tomorrow (Monday) when I am back at
> work. I will start a new thread. I've checked the ascii file and it is
> fine. It certainly *could* be a coding error (I am pretty new to this) but
> the code is soooo simple (read in data and look at it), I would be surprised
> if this was the problem.
>
> David

Post your data file too; it may look fine to you, but have a problem for
some subtle reason you're unaware of.

Subject: Re: G4-related problem
Posted by Mark Hadfield on Sun, 05 Oct 2003 21:21:23 GMT
View Forum Message <> Reply to Message

DPJ wrote:

> Thanks for the response. I will post the code I am using (it's only a few

> lines long) and also include a data file tomorrow (Monday) when I am back at
> work. I will start a new thread. I've checked the ascii file and it is
> fine. It certainly *could* be a coding error (I am pretty new to this) but
> the code is soooo simple (read in data and look at it), I would be surprised
> if this was the problem.

Life is full of surprises :-)

--
Mark Hadfield   "Ka puwaha te tai nei, Hoea tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)