

CoCoALib - Bug #648

QBGenerator crashes

10 Nov 2014 21:22 - John Abbott

Status:	Closed	Start date:	10 Nov 2014
Priority:	High	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Maths Bugs	Estimated time:	4.00 hours
Target version:	CoCoALib-0.99536 June 2015	Spent time:	4.25 hours
<b>Description</b> I have a program which causes QBGenerator to produce a SEGV. Even just printing the QBGenerator causes a SEGV:  <pre>QBG=QBGenerator(QB=[1, x[2], x[1], x[2]^2, x[1]*x[2], x[1]^2, x[2]^3, x[1]*x[2]^2, x[1]^2* x[2]], corners=[x[1]^3, x[2]^4 Process cocoa5 segmentation fault</pre>			
<b>Related issues:</b> Related to CoCoALib - Bug #232: No test for QBGenerator <div>New24 Sep 2012</div>			

History

#1 - 10 Nov 2014 21:43 - John Abbott

Trying to find a simple program which produces the bug; first attempts failed.  
  
Also trying valgrind.

#2 - 11 Nov 2014 09:50 - John Abbott

Still not tracked down the bug :-(  
  
The bug vanishes when I use valgrind -- how can that be?  
  
The bug persists after make veryclean; make (which is Good News, I suppose).  
  
I'll try with MemPoolDebug, and perhaps also disabling MemPool altogether.

#3 - 11 Nov 2014 12:09 - John Abbott

- % Done changed from 0 to 10  
  
The bug did not show up on a (32-bit) Linux VM; nor on a 32-bit Linux netbook.  
  
The bug does not arise when compiled with --threadsafe-hack (which disables MemPool).  
So maybe it is a MemPool problem?

#4 - 11 Nov 2014 14:13 - John Abbott

- Status changed from New to Resolved  
  
- % Done changed from 10 to 60

I think I may have solved the issue: my code wrote one place beyond the end of a vector (and presumably this overwrote something inside the QBGenerator).

Here's what I did to track down the bug (fairly obvious in retrospect):

- compile with `--threadsafe-hack` which disables all MemPool allocators
- run inside valgrind, and look carefully through the output (there was a message about an invalid write of size 4, but note that valgrind lets execution continue!)

Actually locating the line which triggered the bad write was rather *ad hoc*; I had hoped for more help from valgrind here.

If you do not have valgrind, you can try relinking with `debug_new.o` (it is inside `src/AlgebraicCore`). An easy way to do this is to make your executable again; copy the compilation line which make prints out, and append `src/AlgebraicCore/debug_new.o` to the end of the line; then execute the resulting command.

Marking as closed.

**#5 - 17 Nov 2014 11:38 - John Abbott**

- *Status changed from Resolved to Closed*

- *% Done changed from 60 to 100*