

CoCoALib - Bug #1015

Bruns SEGV: SparsePolyIter/DMPI problem

01 Mar 2017 13:37 - John Abbott

Status:	Closed	Start date:	01 Mar 2017
Priority:	Normal	Due date:	
Assignee:	John Abbott	% Done:	100%
Category:	Data Structures	Estimated time:	6.01 hours
Target version:	CoCoALib-0.99550 spring 2017	Spent time:	6.15 hours
Description			
Winfried Bruns reported a SEGV problem by email.			
The following code will exhibit it: F is a poly with a single term in a DMPI ring. Either monomial or the assignment triggers SEGV.			
<pre>void john(const RingElem& F) { SparsePolyRing P=owner(F); long dim=NumIndets(P); vector<long> v(dim); SparsePolyIter it=BeginIter(F); exponents(v,PP(it)); vector<RingElem> junk(1, zero(P)); junk[0] = monomial(P,coeff(it), v); }</pre>			
Related issues:			
Related to CoCoALib - Design #1019: CPP flags in installed library		Closed	05 Mar 2017

History

#1 - 01 Mar 2017 13:40 - John Abbott

- Status changed from New to In Progress
- % Done changed from 0 to 10

The problem arises under the following conditions:

- CoCoALib is configured with `--threadsafe-hack` and `--no-boost`
- compilation of CoCoALib is **with optimization**; problem does not arise if compiled without **-O2**

Compilation of test program is with the following command:

```
g++ -std=c++11 -O3 -Wno-deprecated-declarations -Wall -pedantic -funroll-loops -g -I /home/jabbott/Work/CoCoALib-0.99/include -I . CoCoATest.C /home/jabbott/Work/CoCoALib-0.99/lib/libcocoa.a -lgmpxx -lgmp -o CoCoATest
```

Working to narrow down the problem.

#2 - 01 Mar 2017 14:06 - John Abbott

Here we have a minimal failing example:

```
{
  GlobalManager CoCoAFoundations;

  SparsePolyRing R=NewPolyRing_DMPI(RingQQ(),1);
  const RingElem& F = one(R);

  SparsePolyIter it=BeginIter(F);
  cout <<   coeff(it) << endl;
  cout << "R is .." <<endl;
  cout << R <<endl;
}
```

It fails **in the last line@** whether we print `coeff(it)` or `PP(it)`.

#3 - 01 Mar 2017 14:16 - John Abbott

The problem continues to arise with the following simplified compilation command:

```
g++ -g -I /home/jabbott/Work/CoCoALib-0.99/include -I . CoCoATest.C /home/jabbott/Work/CoCoALib-0.99/lib/libcocoa.a -lgmp -o CoCoATest
```

JAA thinks it is very likely that there is a problem with `SparsePolyIter` over a `DMPI`. Tracking it down will be "fun" 8-
NOTE the problem arises with `g++` versions 5.3.1 and 6.2.1, so is almost certainly out fault :-/

#4 - 01 Mar 2017 14:43 - John Abbott

- Status changed from *In Progress* to *Resolved*

- % Done changed from 10 to 50

The problem goes away if the flag `-DCoCoA_THREADSafe_HACK` is added to the compilation line.

At the moment I do not know how I can protect a user against such an omission.

#5 - 01 Mar 2017 18:04 - John Abbott

- Assignee set to John Abbott
- % Done changed from 50 to 90

I have now added an automatic check for CoCoA_THREADSAFE_HACK.
If you compile the library with one setting and try to compile an application with a different setting then you should get a compile time error (when trying to create a GlobalManager).

I'll check in soon, and may send a trial version to Winfried.

#6 - 02 Mar 2017 15:51 - John Abbott

JAA should also check the value of ULONG2LONG... perhaps need a template approach?

#7 - 03 Mar 2017 15:12 - John Abbott

- Status changed from Resolved to Feedback

I have now added a similar check for ULONG2LONG. It's all a nasty hack, but seems to work.

Checked-in too. Now I must write the documentation :-/

#8 - 05 Mar 2017 21:01 - John Abbott

- Related to Design #1019: CPP flags in installed library added

#9 - 29 Mar 2017 18:10 - John Abbott

- Target version changed from CoCoALib-0.99560 to CoCoALib-0.99550 spring 2017

#10 - 22 Apr 2017 22:42 - John Abbott

- Status changed from Feedback to Closed
- % Done changed from 90 to 100

Regarding this as solved by issue [#1019](#). Closing.

#11 - 28 Apr 2017 09:30 - Anna Maria Bigatti

- Estimated time set to 6.01 h