

GROMACS - Bug #2100

link error with clang

01/17/2017 08:29 PM - Szilárd Páll

Status:	Closed		
Priority:	Normal		
Assignee:	Mark Abraham		
Category:	core library		
Target version:	2018		
Affected version - extra info:		Difficulty:	uncategorized
Affected version:	git master		

Description

The issue has been described and characterized here:

https://mailman-1.sys.kth.se/pipermail/gromacs_org_gmx-developers/2016-October/009351.html

Summary: linking fails reproducibly with the error below whenever gcc 4.8 is used. This happens by default on machines with at least gcc (Ubuntu 4.8.4-2ubuntu1~14.04.3) 4.8.4 (default on Ubuntu 14.04).

Manually passing a recent enough gcc toolchain to clang (as a compiler flag, linker flag is not enough!) avoids the issue. Manual testing has also shown that neither gcc 4.8.4 nor 4.8.5 (both vanilla, compiled from source) work.

```
[...]
/opt/tcbsys/clang/3.9.0/bin/clang++-3.9 -march=core-avx2 -std=c++11 -Wdeprecated -Wextra -W
no-missing-field-initializers -Wpointer-arith -Wall -Wno-unused-function -g --gcc-toolchain=/
opt/tcbsys/gcc/5.2 CMakeFiles/gmx.dir/gmx.cpp.o CMakeFiles/gmx.dir/legacymodules.cpp.o CMakeFiles/
mdrun_objlib.dir/mdrun/md.cpp.o CMakeFiles/mdrun_objlib.dir/mdrun/mdrun.cpp.o CMakeFiles/mdrun_obj
lib.dir/mdrun/membed.cpp.o CMakeFiles/mdrun_objlib.dir/mdrun/repl_ex.cpp.o CMakeFiles/mdrun_objlib
.dir/mdrun/resource-division.cpp.o CMakeFiles/mdrun_objlib.dir/mdrun/runner.cpp.o CMakeFiles/view_
objlib.dir/view/view.cpp.o -o ../../bin/gmx ../../lib/libgromacs.so.3.0.0 -fopenmp=libomp -lm -Wl
,-rpath,"$ORIGIN/../../lib"
CMakeFiles/mdrun_objlib.dir/mdrun/md.cpp.o: In function `gmx::do_md(_IO_FILE*, t_commrec*, gmx::MD
Logger const&, int, t_filenm const*, gmx_output_env_t const*, int, int, gmx_vsite_t*, gmx_constr*,
int, t_inputrec*, gmx_mtop_t*, t_fcdata*, t_state*, energyhistory_t*, t_mdatoms*, t_nrnbs*, gmx_wa
llcycle*, gmx_edsam*, t_forcerc*, int, int, int, gmx_membed_t*, float, float, int, unsigned long,
gmx_walltime_accounting*)':
/data/pszilard/gromacs-master/src/programs/mdrun/md.cpp:420: undefined reference to `ekinstate_t::
~ekinstate_t()'
CMakeFiles/mdrun_objlib.dir/mdrun/md.cpp.o: In function `~t_state':
/data/pszilard/gromacs-master/src/gromacs/mdtypes/state.h:195: undefined reference to `ekinstate_t
::~~ekinstate_t()'
clang-3.9: error: linker command failed with exit code 1 (use -v to see invocation)
```

Repro'd with:

- clang 3.7-4.0-dev

- gcc 4.8.x

- by default will fail on Ubuntu 14.04 (and won't fail on either 12.04 or 16.04 as gcc 4.8 is not picked).

History

#1 - 01/17/2017 08:30 PM - Szilárd Páll

Note that Aleksei has found [this report](#) which seems similar in spirit, but I failed to eliminate the error it triggers by passing a gcc 5 toolchain.

#2 - 01/18/2017 10:36 PM - Roland Schulz

Not sure you already tried this. But by removing the brace initializer in GROMACS the problem goes away. So indeed does sound related.

#3 - 01/18/2017 11:23 PM - Roland Schulz

Cause: <https://gerrit.gromacs.org/#/c/6194/10/src/programs/mdrun/md.cpp@419>

#4 - 01/18/2017 11:59 PM - Mark Abraham

Roland Schulz wrote:

Cause: <https://gerrit.gromacs.org/#/c/6194/10/src/programs/mdrun/md.cpp@419>

If so, resolved by <https://gerrit.gromacs.org/#/c/6333/12>

It seems clear that the brace initialization isn't as reliable as we'd hope, but isn't clear why. Since that's anyway a brittle construct to use, we should in future not attempt to replace `snew` with such initialization, but instead write a proper constructor.

#5 - 01/19/2017 12:02 AM - Mark Abraham

Mark Abraham wrote:

Roland Schulz wrote:

Cause: <https://gerrit.gromacs.org/#/c/6194/10/src/programs/mdrun/md.cpp@419>

If so, resolved by <https://gerrit.gromacs.org/#/c/6333/12>

It seems clear that the brace initialization isn't as reliable as we'd hope, but isn't clear why. Since that's anyway a brittle construct to use, we should in future not attempt to replace `snew` with such initialization, but instead write a proper constructor.

More discussion found at <https://gerrit.gromacs.org/#/c/6279/5/src/gromacs/mdtypes/energyhistory.h@a56>

#6 - 01/19/2017 03:31 PM - Szilárd Páll

Roland Schulz wrote:

Not sure you already tried this. But by removing the brace initializer in GROMACS the problem goes away. So indeed does sound related.

Yeah, I did try just forgot to get back. Still not sure why does it depend on the gcc toolchain in our case, but the same does not help in the repro case on the LLVM bug report. Any idea why is that?

#7 - 01/20/2017 11:26 AM - Mark Abraham

Szilárd Páll wrote:

Roland Schulz wrote:

Not sure you already tried this. But by removing the brace initializer in GROMACS the problem goes away. So indeed does sound related.

Yeah, I did try just forgot to get back. Still not sure why does it depend on the gcc toolchain in our case, but the same does not help in the repro case on the LLVM bug report. Any idea why is that?

It's a linking error for a destructor that the compiler presumably omitted to generate. I presume the implementation of operator `new` is in the standard library, and might somehow generate calls for a destructor that then can't be linked. Obviously you'd hope the generation and the requirements were coupled appropriately, but using multiple versions of infrastructure from another project can be brittle. Presumably there's an aspect of gcc 4.8 that's relevant for the GROMACS issue, and a more long-lived issue for LLVM.

#8 - 01/23/2017 11:52 AM - Mark Abraham

- Category set to *core library*
- Status changed from *New* to *Feedback wanted*
- Assignee set to *Mark Abraham*
- Target version set to *2018*

<https://gerrit.gromacs.org/#/c/6333/12> is now submitted on master branch, so I believe this has been resolved

#9 - 01/23/2017 12:12 PM - Aleksei lupinov

Can confirm, ebca651 solves the issue on gcc 4.8.4 and clang 3.7 as per Szilard's report.

#10 - 01/23/2017 04:39 PM - Mark Abraham

- Status changed from *Feedback wanted* to *Closed*

OK. Please re-open if there's further issues, etc.