



Bug 1657544 - clang++ linking fails with undefined reference to __muloti4

Keywords:

Status: CLOSED ERRATA

Alias: None

Product: Fedora

Component: clang  

Version: 29

Hardware: Unspecified

OS: Unspecified

Priority: unspecified

Severity: unspecified

Target Milestone: ---

Assignee: Tom Stellard

QA Contact: Fedora Extras Quality Assurance

Docs Contact:

URL:

Whiteboard:

Depends On:

Blocks:

TreeView+ [depends on](#) / [blocked](#)

Reported: 2018-12-09 20:38 UTC by Georg Sauthoff

Modified: 2019-02-18 02:04 UTC ([History](#))

CC List: 5 users ([show](#))

Fixed In Version: clang-7.0.1-2.fc29

Doc Type: If docs needed, set a value

Doc Text:

Clone Of:

Environment:

Last Closed: 2019-02-18 02:04:04 UTC

Dependent Products:

Attachments		(Terms of Use)
minimal reproducing C++ example (281 bytes, text/x-csrc) 2018-12-09 20:38 UTC, Georg Sauthoff	no flags	Details
Add an attachment (proposed patch, testcase, etc.)		

Links

System	ID	Priority	Status	Summary	Last Updated
LLVM	34920	None	None	None	2018-12-18 19:06:03 UTC

Georg Sauthoff 2018-12-09 20:38:42 UTC

[Description](#)

Created [attachment 1512902 \[details\]](#)
minimal reproducing C++ example

Description of problem:

When compiling a program that calls `std::from_chars()` clang++

cannot resolve the symbol __muloti4 during linking.

Version-Release number of selected component (if applicable):

clang-7.0.0-2.fc29.x86_64

How reproducible:

always

Steps to Reproduce:

1. get attached minimal example
2. clang++ -Wall tochars_clang_fail.cc -o tochars
3. ./tochars 123

Actual results:

```
/usr/bin/ld: /tmp/tochars_clang_fail-e603f8.o: in function
`bool std::__detail::__raise_and_add<unsigned long>(unsigned
long&, int, unsigned char)':
tochars_clang_fail.cc:
(.text._ZNSt8__detail15__raise_and_addImEEbRT_ih[_ZNSt8__detai
l15__raise_and_addImEEbRT_ih]+0x4c): undefined reference to
`__muloti4'
clang-7: error: linker command failed with exit code 1 (use -v
to see invocation
```

Expected results:

successful compilation and as output:
123

Additional info:

Compiling with GCC works as expected:

```
g++ -Wall tochars_clang_fail.cc -o tochars
```

Linking with libc++ also does work:

```
clang++ -Wall -stdlib=libc++ tochars_clang_fail.cc -o tochars
```

Tom Stellard 2018-12-18 06:35:31 UTC

[Comment 1](#)

Possibly related upstream commit:

<https://reviews.llvm.org/rL320902>

Tom Stellard 2018-12-18 06:46:31 UTC

[Comment 2](#)

Simplified test case:

```
#include <stddef.h>

void mul(size_t a, int b) {
    size_t res;
    __builtin_mul_overflow(a, b, &res);
}
```

Tom Stellard 2018-12-21 03:48:56 UTC

[Comment 3](#)

Fixed in rawhide: clang-7.0.1-2.fc30. I will backport this back to f29 too.

Tom Stellard 2019-02-12 20:31:41 UTC

[Comment 4](#)

Submitted PR: <https://src.fedoraproject.org/rpms/clang/pull-request/33>

Fedora Update System 2019-02-14 16:09:51 UTC

[Comment 5](#)

clang-7.0.1-2.fc29 has been submitted as an update to Fedora 29. <https://bodhi.fedoraproject.org/updates/FEDORA-2019-5065cb8af8>

Fedora Update System 2019-02-15 02:57:23 UTC

[Comment 6](#)

clang-7.0.1-2.fc29 has been pushed to the Fedora 29 testing repository. If problems still persist, please make note of it in this bug report.

See https://fedoraproject.org/wiki/QA:Updates_Testing for instructions on how to install test updates.

You can provide feedback for this update here:

<https://bodhi.fedoraproject.org/updates/FEDORA-2019-5065cb8af8>

Fedora Update System 2019-02-18 02:04:04 UTC

[Comment 7](#)

clang-7.0.1-2.fc29 has been pushed to the Fedora 29 stable repository. If problems still persist, please make note of it in this bug report.

Note

You need to [log in](#) before you can comment on or make changes to this bug.

