undefined reference to `__muloti4')

Asked 1 year, 5 months ago Active 1 year, 5 months ago Viewed 303 times

3

In clang, I run into linking error if the Undefined Behavior Sanitizer (`-fsanitize=undefined`) when the program uses 128 bit integer. The linking errors complain about `__muloti4` :

```
$ cat example.c
__int128_t a;
int main (void) {
  a = a * a;
  return 0;
}

$ clang -fsanitize=undefined example.c
/tmp/example-df4873.o: In function `main':
example.c:(.text+0x4c): undefined reference to `__muloti4'
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```

(Tested on Ubuntu 17.10 with clang 4.0.1.)

With gcc it works out of the box (`gcc -fsanitize=undefined example.c`).

What does work with clang is the following call, but I neither understand it fully (`--rtlib=compiler-rt`), nor does it look like to me:

```
clang -lgcc_s -lubsan --rtlib=compiler-rt -fsanitize=undefined /tmp/example.c
```

I found it by trial and error, but it feels wrong to use clang and link against some gcc library. Also explicitly linking against `ubsan` should not be necessary according to the documentation.

Is this the correct way to get rid of the error, or is there a more robust solution?

c clang linker-errors sanitizer ubsan

edited Apr 12 '18 at 10:53

asked Apr 12 '18 at 10:00

Philipp Claßen
20.2k 15 91 149

1 Answer

2

This is a known problem (also see this). Libgcc (which clang links against by default) does not provide necessary symbols to sanitize 128-bit types so you need to ask clang to use compiler-rt runtime library instead.

answered Apr 12 '18 at 11:32