

Ruby - Bug #17645

`shorten-64-to-32` errors with Clang 11.1.0 for 32-bit targets w.r.t. `time\_t` (maybe Y2038?)

02/19/2021 03:20 PM - xtkoba (Tee KOBAYASHI)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	ruby 3.1.0dev (2021-02-19 master a319c04818) [i686-linux]	Backport: 2.5: DONTNEED, 2.6: DONTNEED, 2.7: DONE, 3.0: DONE
Description		
Targeting to i686-linux-gnu or armv7a-linux-gnu, Clang/LLVM 11.1.0 fails to compile file.c:		
<pre>compiling file.c file.c:1256:44: error: implicit conversion loses integer precision: 'const __s64' (aka 'const long long') to 'time_t' (aka 'long') [-Werror,-Wshorten-64-to-32] return rb_time_nano_new(stx->stx_btime.tv_sec, stx->stx_btime.tv_nsec); ~~~~~^~~~~~ 1 error generated. make: *** [Makefile:428: file.o] Error 1</pre>		
A workaround is to explicitly cast the first argument of rb_time_nano_new, as in the attached patch. It will work until 2038.		

Associated revisions

Revision 3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f - 03/14/2021 11:10 AM - xtkoba (Tee KOBAYASHI)

Explicitly cast __s64 to time_t [Bug #17645]

A workaround of shorten-64-to-32 error where 32-bit linux.

Revision 3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f - 03/14/2021 11:10 AM - xtkoba (Tee KOBAYASHI)

Explicitly cast __s64 to time_t [Bug #17645]

A workaround of shorten-64-to-32 error where 32-bit linux.

Revision 3ac28de5 - 03/14/2021 11:10 AM - xtkoba (Tee KOBAYASHI)

Explicitly cast __s64 to time_t [Bug #17645]

A workaround of shorten-64-to-32 error where 32-bit linux.

Revision 2a6bfd22468343003463e0cbf91953a01b0dbba5 - 03/14/2021 11:18 AM - nobu (Nobuyoshi Nakada)

Properly convert time_t [Bug #17645]

Revision 2a6bfd22468343003463e0cbf91953a01b0dbba5 - 03/14/2021 11:18 AM - nobu (Nobuyoshi Nakada)

Properly convert time_t [Bug #17645]

Revision 2a6bfd22 - 03/14/2021 11:18 AM - nobu (Nobuyoshi Nakada)

Properly convert time_t [Bug #17645]

Revision ff2ea4daeb795ebb3d0afa8c57d86a4af9757e50 - 03/14/2021 12:30 PM - naruse (Yui NARUSE)

merge revision(s) 2a6bfd22468343003463e0cbf91953a01b0dbba5,3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f: [Backport #17645]

```
Properly convert time_t [Bug #17645]

---
thread_sync.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

Explicitly cast __s64 to time_t [Bug #17645]
```

```
A workaround of shorten-64-to-32 error where 32-bit linux.
---
file.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

Revision ff2ea4daeb795ebb3d0afa8c57d86a4af9757e50 - 03/14/2021 12:30 PM - naruse (Yui NARUSE)

merge revision(s) 2a6bfd22468343003463e0cbf91953a01b0dbba5,3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f: [Backport #17645]

```
Properly convert time_t [Bug #17645]
```

```
---
thread_sync.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
Explicitly cast __s64 to time_t [Bug #17645]
```

```
A workaround of shorten-64-to-32 error where 32-bit linux.
---
file.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

Revision ff2ea4da - 03/14/2021 12:30 PM - naruse (Yui NARUSE)

merge revision(s) 2a6bfd22468343003463e0cbf91953a01b0dbba5,3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f: [Backport #17645]

```
Properly convert time_t [Bug #17645]
```

```
---
thread_sync.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
Explicitly cast __s64 to time_t [Bug #17645]
```

```
A workaround of shorten-64-to-32 error where 32-bit linux.
---
file.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

Revision 9994860dcb56697fad1799128d1493adc06b23c7 - 03/20/2021 04:02 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 2a6bfd22468343003463e0cbf91953a01b0dbba5,3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f: [Backport #17645]

```
Properly convert time_t [Bug #17645]
```

```
---
thread_sync.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
Explicitly cast __s64 to time_t [Bug #17645]
```

```
A workaround of shorten-64-to-32 error where 32-bit linux.
---
file.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

Revision 9994860dcb56697fad1799128d1493adc06b23c7 - 03/20/2021 04:02 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 2a6bfd22468343003463e0cbf91953a01b0dbba5,3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f: [Backport #17645]

```
Properly convert time_t [Bug #17645]
```

```
---
thread_sync.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

```
Explicitly cast __s64 to time_t [Bug #17645]
```

```
A workaround of shorten-64-to-32 error where 32-bit linux.
---
file.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

Revision 9994860d - 03/20/2021 04:02 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 2a6bfd22468343003463e0cbf91953a01b0dbba5,3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f: [Backport #17645]

Properly convert time_t [Bug #17645]

```
---
thread_sync.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

Explicitly cast __s64 to time_t [Bug #17645]

A workaround of shorten-64-to-32 error where 32-bit linux.

```
---
file.c | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)
```

History

#1 - 03/11/2021 07:31 AM - xtkoba (Tee KOBAYASHI)

- Subject changed from `make` stops at `file.c` with Clang 11.1.0 for 32-bit GNU/Linux targets (maybe Y2038?) to `shorten-64-to-32` errors with Clang 11.1.0 for 32-bit targets w.r.t. `time\_t` (maybe Y2038?) `shorten-64-to-32` errors with Clang 11.1.0 for 32-bit targets w.r.t. `time\_t` (maybe Y2038?)

I encountered another compilation error of shorten-64-to-32 for 32-bit musl targets, where time_t is 64-bit. A workaround:

```
--- a/thread_sync.c
+++ b/thread_sync.c
@@ -560,7 +560,7 @@ rb_mutex_sleep(VALUE self, VALUE timeout

    RUBY_VM_CHECK_INTS_BLOCKING(GET_EC());
    time_t end = time(0) - beg;
-   return INT2FIX(end);
+   return INT2FIX((long)end);
}

/*
```

This case, however, does not seem to cause an Y2038 problem in the real world, as the value of end is a time difference in seconds and will almost always be less than 2³¹. We need only to suppress the compilation error.

#2 - 03/11/2021 07:32 AM - xtkoba (Tee KOBAYASHI)

- Subject changed from `shorten-64-to-32` errors with Clang 11.1.0 for 32-bit targets w.r.t. `time\_t` (maybe Y2038?) `shorten-64-to-32` errors with Clang 11.1.0 for 32-bit targets w.r.t. `time\_t` (maybe Y2038?) to `shorten-64-to-32` errors with Clang 11.1.0 for 32-bit targets w.r.t. `time\_t` (maybe Y2038?)

#3 - 03/12/2021 02:39 AM - nobu (Nobuyoshi Nakada)

Yet another solution would be to define rb_time64_t for rb_time_nano_new (and rb_time_new), but may be an overkill.

#4 - 03/14/2021 11:13 AM - xtkoba (Tee KOBAYASHI)

- Status changed from Open to Closed

Applied in changeset [git|3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f](https://github.com/ruby/ruby/commit/3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f).

Explicitly cast __s64 to time_t [Bug [#17645](#)]

A workaround of shorten-64-to-32 error where 32-bit linux.

#5 - 03/14/2021 11:23 AM - nobu (Nobuyoshi Nakada)

- Backport changed from 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN to 2.5: DONTNEED, 2.6: DONTNEED, 2.7: REQUIRED, 3.0: REQUIRED

#6 - 03/14/2021 01:37 PM - naruse (Yui NARUSE)

- Backport changed from 2.5: DONTNEED, 2.6: DONTNEED, 2.7: REQUIRED, 3.0: REQUIRED to 2.5: DONTNEED, 2.6: DONTNEED, 2.7: REQUIRED, 3.0: DONE

ruby_3_0 ff2ea4daeb795ebb3d0afa8c57d86a4af9757e50 merged revision(s)
2a6bfd22468343003463e0cbf91953a01b0dbba5,3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f.

#7 - 03/20/2021 04:19 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.5: DONTNEED, 2.6: DONTNEED, 2.7: REQUIRED, 3.0: DONE to 2.5: DONTNEED, 2.6: DONTNEED, 2.7: DONE, 3.0: DONE

ruby_2_7 9994860dcb56697fad1799128d1493adc06b23c7 merged revision(s)
2a6bfd22468343003463e0cbf91953a01b0dbba5,3ac28de5414c81a0da8c8ec969ea74db5a0f1b0f.

Files

ruby-time_t.patch	358 Bytes	02/19/2021	xtkoba (Tee KOBAYASHI)
-------------------	-----------	------------	------------------------