Ruby master - Bug #16455

coroutine ucontext uses deprecated POSIX getcontext/swapcontext/makecontext, absent in musl and uclibc

12/26/2019 08:03 PM - luizluca (Luiz Angelo Daros de Luca)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Closed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td>ioquatix (Samuel Williams)</td>
</tr>
<tr>
<td>Target version:</td>
<td>3.0</td>
</tr>
<tr>
<td>ruby -v:</td>
<td>ruby 2.7.0p0 (2019-12-25 revision 647ee6f091)</td>
</tr>
<tr>
<td>Backport:</td>
<td>2.5: UNKNOWN, 2.6: UNKNOWN</td>
</tr>
</tbody>
</table>

Description
Hello,

While building ruby 2.7.0 for mips with musl, it fails:

```bash
linking miniruby
  cont.c:(.text+0x36c): undefined reference to `getcontext'
  cont.c:(.text+0xde6): undefined reference to `swapcontext'
/home/luizluca/prog-local/openwrt/trunk/staging_dir/toolchain-mips_24kc_gcc-8.3.0_musl/lib/gcc/mips-openwrt-linux-musl/8.3.0/../../../../mips-openwrt-linux-musl/bin/ld: cont.o: in function `fiber_switch':
  cont.c:(.text+0x16b8): undefined reference to `makecontext'
collect2: error: ld returned 1 exit status
make[3]: *** [Makefile:271: miniruby] Error 1
```

It seems that now ruby either uses native ASM code or ucontext for coroutine. Both are unavailable in my crossbuild. It was building and running with 2.6.x:

```bash
configure:25935: checking native coroutine implementation for mips-linux-gnu
configure:25976: result: no
```

With 2.7.0:

```bash
configure:26536: checking native coroutine implementation for mips-linux-gnu
configure:26601: result: ucontext
```

I tried to disable it with --without-coroutine, --with-out-coroutine and --with-coroutine=no, but it is innocuous or try to use a "no" implementation.

It looks like this introduced the change [https://github.com/ruby/ruby/commit/7291fef55c90b9ab6b3c22018b16972861b98c9d](https://github.com/ruby/ruby/commit/7291fef55c90b9ab6b3c22018b16972861b98c9d)

And musl might never fix it: [https://wiki.musl-libc.org/open-issues.html](https://wiki.musl-libc.org/open-issues.html)

History

#1 - 12/26/2019 10:28 PM - mame (Yusuke Endoh)
- Assignee set to ioquatix (Samuel Williams)

#2 - 12/27/2019 04:19 AM - nobu (Nobuyoshi Nakada)

luizluca (Luiz Angelo Daros de Luca) wrote:

- I tried to disable it with --without-coroutine, --with-out-coroutine and --with-coroutine=no, but it is innocuous or try to use a "no" implementation.
Does —-with-coroutine=copy work?

#3 - 12/30/2019 10:22 PM - luizluca (Luiz Angelo Daros de Luca)

nobu (Nobuyoshi Nakada) wrote:

luizluca (Luiz Angelo Daros de Luca) wrote:

I tried to disable it with --without-coroutine, --without-coroutine and --without-coroutine=no, but it is innocuous or try to use a "no" implementation.

Does —-with-coroutine=copy work?

Yes. And it also affects uclibc.

autoconf should test for getcontext/swapcontext/makecontext at link time as musl do have them declared (but not implemented). uclibc fails even before, while building as it does not even declare them.

For those cases, copy might work.

#4 - 12/31/2019 01:09 AM - luizluca (Luiz Angelo Daros de Luca)

- Subject changed from coroutine ucontext uses deprecated POSIX getcontext/swapcontext/makecontext, absent in musl to coroutine ucontext uses deprecated POSIX getcontext/swapcontext/makecontext, absent in musl and uclibc

nobu (Nobuyoshi Nakada) wrote:

luizluca (Luiz Angelo Daros de Luca) wrote:

I tried to disable it with --without-coroutine, --without-coroutine and --without-coroutine=no, but it is innocuous or try to use a "no" implementation.

Does —-with-coroutine=copy work?

I created a patch to fix the build with musl/uclibc:

--- a/configure.ac  
+++ b/configure.ac
@@ -2344,7 +2344,10 @@ AS_CASE([$rb_cv_coroutine], [yes''], [
     rb_cv_coroutine=copy ,
     [""], [ 
     rb_cv_coroutine=ucontext +
     AC_CHECK_FUNCS([getcontext swapcontext makecontext],+ 
     [rb_cv_coroutine=ucontext],+ 
     [rb_cv_coroutine=copy; break]+ 
     ])

     AC_MSG_RESULT({$rb_cv_coroutine})

It is ugly as it outputs AC_CHECK_FUNCS tests in the middle of checking coroutine implementations.

#5 - 12/31/2019 03:06 AM - luizluca (Luiz Angelo Daros de Luca)

There is also a missing include for copy to work with musl:

+++ a/coroutine/copy/Context.h
++++ b/coroutine/copy/Context.h
@@ -13,6 +13,7 @@
 #include <alloca.h>
+ #include <sys/types.h>
++ #include <sys/types.h>
+ + #define COROUTINE __attribute__((noreturn)) void
+

#6 - 01/17/2020 08:07 PM - ioquatix (Samuel Williams)

Copy implementation is the right one. But what architecture are you compiling for? because copy coroutine is so slow.

03/19/2022
#7 - 04/11/2020 02:34 AM - ioquatix (Samuel Williams)
- Target version set to 36
- Status changed from Open to Closed

It is fixed: https://github.com/ruby/ruby/pull/2995

#8 - 04/23/2020 08:07 AM - ncopa (Natanael Copa)
This is still an issue for alpine s390x. See #16809

#9 - 09/22/2020 11:08 PM - puchuu (Andrew Aladjev)
ioquatix (Samuel Williams) wrote in #note-7:

It is fixed: https://github.com/ruby/ruby/pull/2995

Hello, you have provided good fix but it is related to armv7 only. For example: aarch64-gentoo-linux-musl and aarch64_be-gentoo-linux-musl are valid CHOSTS, but first one works fine, second one fails. First one is using "rb_cv_coroutine=arm64", second one is using default "rb_cv_coroutine=ucontext". Same thing will be true for s390x, other arches: it will try to use default broken (for musl) "rb_cv_coroutine=ucontext".

luizluca (Luiz Angelo Daros de Luca) have provided right fix - it works perfect. https://github.com/ruby/ruby/pull/3567

#10 - 09/23/2020 11:12 PM - ioquatix (Samuel Williams)
Understood. The PR looks acceptable and the right fix.

My advice is to avoid the copy coroutine implementation if at all possible, so we should try to detect as many native implementation paths as possible.

#11 - 09/29/2020 03:37 AM - hsbt (Hiroshi SHIBATA)
- Target version changed from 36 to 3.0

#12 - 10/01/2020 10:19 PM - puchuu (Andrew Aladjev)
- File aarch64_be.patch added

I've checked that current "aarch64" coroutine works fine for "aarch64_be" (at least for qemu). So I am going to propose to update "aarch64" wildcard.

#13 - 02/06/2021 06:28 AM - luizluca (Luiz Angelo Daros de Luca)
I still the get same issue with 3.0.0.

Ruby cannot blindly try to use ucontext without checking for getcontext swapcontext makecontext presence. https://github.com/ruby/ruby/pull/3567 does work. It only needs to be refreshed, removing the #include.

Files

<table>
<thead>
<tr>
<th>Name</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>aarch64_be.patch</td>
<td>480 Bytes</td>
<td>10/01/2020</td>
<td>puchuu (Andrew Aladjev)</td>
</tr>
</tbody>
</table>