Ruby - Bug #18912

Build failure with Xcode 14 and macOS 13 (Ventura) Beta

07/14/2022 05:40 AM - hsbt (Hiroshi SHIBATA)

Status:	Closed		
Priority:	Normal		
Assignee:	hsbt (Hiroshi SHIBATA)		
Target version:			
ruby -v:		**Backport:**	2.7: DONE, 3.0: DONE, 3.1: DONE

Description

TL;DR: We fixed this issue at Ruby 2.7-3.1 and master branch. But the stable versions are not released yet.

- Option 1: We strongly recommend to not upgrade Xcode 14 and its toolchains for macOS 12(Monterey) users. If you already update Xcode 14, you remove them with sudo rm -rf /Library/Developer/CommandLineTools and install Xcode 13.x or toolchains from https://developer.apple.com/download/all
- Option 2: macOS 13 (Ventura) couldn't install Xcode13. We should add --without=+,bigdecimal --enable-shared to the configure option.

Today, I tried to build ruby master with macOS 13 (Ventura) Beta. It breaks the build status caused by Xcode 14 beta changes.

1. Build failed without --enable-shared.

I build ruby master without --enable-shared option. I got the following error.

```
(snip)
linking shared-object -test-/arith_seq/extract.bundle
Undefined symbols for architecture arm64:
  "_rb_arithmetic_sequence_extract", referenced from:
      _arith_seq_s_extract in extract.o
  "_rb_ary_new_capa", referenced from:
      _arith_seq_s_extract in extract.o
  "_rb_ary_store", referenced from:
      _arith_seq_s_extract in extract.o
  "_rb_define_singleton_method", referenced from:
      _Init_extract in extract.o
  "_rb_path2class", referenced from:
      _Init_extract in extract.o
ld: symbol(s) not found for architecture arm64
clang: error: linker command failed with exit code 1 (use -v to see invocation)
```

@katei (Yuta Saito) says this error affects with -undefined dynamic_lookup flags.

1. Build error with bigdecimal

With --enabled-shared resolved the first error. But I got the another build failure.

```
compiling bigdecimal.c
In file included from bigdecimal.c:13:
In file included from ./bigdecimal.h:14:
./missing.h:127:1: error: static declaration of 'rb_rational_num' follows non-static declaration
rb_rational_num(VALUE rat)
^
../../../include/ruby/internal/intern/rational.h:128:7: note: previous declaration is here
VALUE rb_rational_num(VALUE rat);
      ^
In file included from bigdecimal.c:13:
In file included from ./bigdecimal.h:14:
(snip)
```

It's affected with static inline declaration in missing.h of bigdecimal.

1. test failure with mjit

I could build with --with-out-ext=+,bigdecimal --enable-share option. But I also got the test failure with mjit.

```
[215/402] TestMJIT#test_lambda_longjmp = 0.19 s
192) Failure:
TestMJIT#test_lambda_longjmp [/Users/hsbt/Documents/github.com/ruby/ruby/test/ruby/test_mjit.rb:10
45]:
Expected 1 times of JIT success, but succeeded 0 times.

script:
"""

fib = lambda do |x|
  return x if x == 0 || x == 1
  fib.call(x-1) + fib.call(x-2)
end
print fib.call(5)

"""


stderr:
"""
Undefined symbols for architecture arm64:
  "_mjit_call_p", referenced from:
      __mjit0 in _ruby_mjit_p39885u0-643ab5.o
      _vm_sendish in _ruby_mjit_p39885u0-643ab5.o
```

I already shared this to @k0kubun (Takashi Kokubun) .

macOS 13 beta is still development status. I will track this until the official release date.

Related issues:

Related to Ruby - Bug #19005: Ruby interpreter compiled XCode 14 cannot build...	**Closed**
Related to Ruby - Feature #19422: Make `--enabled-shared` mandatory on macOS	**Assigned**
Related to Ruby - Bug #20631: Build failure with Xcode 16 beta and macOS 15 (...	**Closed**

History

#1 - 07/19/2022 03:44 PM - alanwu (Alan Wu)

Can you build with bigdecimal included and post
cat ext/bigdecimal/mkmf.log from the build directory?
I ran into similar errors when building on illumos and
it was due to bigdecimal not configuring properly.

#2 - 07/28/2022 08:32 AM - hsbt (Hiroshi SHIBATA)

@alanwu (Alan Wu) full log is here https://gist.github.com/hsbt/293250f86a2731b4f76ceea4d903bd64

#3 - 07/28/2022 09:04 AM - katei (Yuta Saito)

I've made a patch to address the linking failure issue by replacing -undefined dynamic_lookup with -bundle_loader.
https://github.com/ruby/ruby/pull/6193

The patch has been tested with Xcode 14 beta2 toolchain on macOS 12.4 Monterey, but not tested on the actual Ventura yet.

#4 - 07/28/2022 09:43 AM - hsbt (Hiroshi SHIBATA)

@katei (Yuta Saito) Thanks! I will test your patch when it's complete.

FYI: Xcode 14 beta 3 with macOS 13.0 public beta is still same results.

#5 - 07/28/2022 12:01 PM - alanwu (Alan Wu)

@hsbt (Hiroshi SHIBATA) Thanks for the logs. Some symbols are
indeed not found due to link errors. For example,
rb_complex_real() is provided but not found.

The linker error contradicts itself:

```
have_func: checking for rb_complex_real() in ruby.h... -------------------- no
...
ld: warning: ignoring file ../../libruby.3.2-static.a, building for macOS-arm64 but attempting to link with fi
le built for macOS-arm64
```

So the error message could be better, at least.
Maybe the linker is not printing out the full
architecture stamp that it's comparing. You could
try checking otool -hv libruby.3.2-static.a and
comparing that with otool -hv bigdecimal.o.
The cpusubtype column might be different.
Also, maybe bigdecimal and libruby are built with
different toolchains due the way the XCode beta is
setup? I should find some time to install the beta...

#6 - 08/01/2022 04:24 AM - hsbt (Hiroshi SHIBATA)

I tried https://github.com/ruby/ruby/pull/6193 .

The first try is failed. Because my environment mixed the homebrew binutils built by macOS 12.0(monterey).

I could build ruby after removing GNU binutils from PATH. and it could build bigdecimal and pass with tests related MJIT.

#7 - 08/01/2022 06:23 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Open to Assigned

- Assignee set to hsbt (Hiroshi SHIBATA)

#8 - 08/05/2022 03:40 AM - hsbt (Hiroshi SHIBATA)

After merging https://github.com/ruby/ruby/pull/6193. We could build Ruby.

But We have still failing tests:

```
TestProcess#test_daemon_nocloseobjc[10525]: +[NSPlaceholderMutableString initialize] may have been in progress
 in another thread when fork() was called.
objc[10525]: +[NSPlaceholderMutableString initialize] may have been in progress in another thread when fork()
was called. We cannot safely call it or ignore it in the fork() child process. Crashing instead. Set a breakpo
int on objc_initializeAfterForkError to debug.
 = 0.00 s
  1) Failure:
TestProcess#test_daemon_noclose [/Users/hsbt/Documents/github.com/ruby/ruby/test/ruby/test_process.rb:1846]:
<"ok\n" + "/\n"> expected but was
<"">.
```

and spec/ruby/core/process/daemon_spec.rb is also failed.

#9 - 08/23/2022 10:09 AM - hsbt (Hiroshi SHIBATA)

TestProcess#test_daemon_noclose passed with export OBJC_DISABLE_INITIALIZE_FORK_SAFETY=YES from
https://github.com/rails/rails/issues/38560

#10 - 09/15/2022 11:24 PM - hsbt (Hiroshi SHIBATA)

- Related to Bug #19005: Ruby interpreter compiled XCode 14 cannot build some native gems on macOS added

#11 - 09/17/2022 07:47 AM - hsbt (Hiroshi SHIBATA)

- Subject changed from Build failure with macOS 13 (Ventura) Beta to Build failure with Xcode 14 and macOS 13 (Ventura) Beta

Xcode 14 has been released. The all stable versions of Ruby couldn't build with Xcode 14.

We fixed this issue at Ruby 2.7-3.1 and master branch. But the stable versions are not released yet.

I strongly recommend to not upgrade Xcode 14 and its toolchains for macOS users. If you already update Xcode 14, you remove them with

```
sudo rm -rf /Library/Developer/CommandLineTools
```

and install Xcode 13.x or toolchains from https://developer.apple.com/download/all

#12 - 09/17/2022 11:29 AM - katei (Yuta Saito)

I've been doing some digging around the TestProcess#test_daemon_noclose test failure.
Minimum reproducible code is here

```
Process.daemon(false, true)
Dir.pwd
```

And got some facts:

- On macOS, you should not use Objective-C classes for the first time in a process forked from a multithreaded process without exec
  - Because the +initialize mechanism called the first time you use an Objective-C class is not async signal safe.
  - Sources: detailed blog post, comments in objc4 source code
- CRuby on macOS is always multithreaded due to the timer thread.
- Kernel.fork and Process.daemon can create "forked process without exec".
  - In other words, using Objective-C API after forked process may cause deadlock.
- CoreFoundation API used by Dir.pwd now depends on Objective-C classes from Ventura. (NEW in Ventura)

So, I guess this means that the potential deadlock was everywhere even before Ventura, and it was not caught in test suites. However it was finally revealed, thanks(?) to the CoreFoundation change introduced in Ventura.

IMHO this unsafe-ness is a problem that can only be solved on the user program side, not CRuby. So we can leave this problem to user responsibility and update the test case to be Objective-C friendly...
(I know this is not ideal solution, so let me know if you have a better solution)

#13 - 09/18/2022 02:47 AM - nobu (Nobuyoshi Nakada)

- Backport changed from 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN to 2.7: REQUIRED, 3.0: REQUIRED, 3.1: REQUIRED

Thank you, then is it ok if CFString has been initialized first?

```
diff --git a/file.c b/file.c
index de7ed5e33ba..b91b8c346cd 100644
--- a/file.c
+++ b/file.c
@@ -268,6 +268,20 @@ rb_str_encode_ospath(VALUE path)

 #ifdef __APPLE__
 # define NORMALIZE_UTF8PATH 1
+__attribute__((constructor))
+static void
+finish_async_unsafe_CFString_class_initialization(void)
+{
+    const UInt8 empty[1] = {0};
+    CFStringRef s = CFStringCreateWithBytesNoCopy(kCFAllocatorDefault,
+                                                  empty, 0,
+                                                  kCFStringEncodingUTF8, FALSE,
+                                                  kCFAllocatorNull);
+    CFMutableStringRef m = CFStringCreateMutableCopy(kCFAllocatorDefault, 0, s);
+    CFRelease(m);
+    CFRelease(s);
+}
+
 static VALUE
 rb_str_append_normalized_ospath(VALUE str, const char *ptr, long len)
 {
```

#14 - 09/19/2022 03:32 AM - nobu (Nobuyoshi Nakada)

According to the experiments by @katei (Yuta Saito), the followings are ok.

- IO.popen

  ```
  p IO.popen("-") { |f|
    if f
      f.gets
    else
      puts Dir.pwd
    end
  }
  ```

- Dir.pwd in forked process

  ```
  fork {p Dir.pwd}
  ```

- Dir.pwd in doubly forked process

  ```
  fork {fork {p Dir.pwd}}
  ```

Only the code using Process.daemon in #note-12 does crash.

#15 - 09/23/2022 06:53 PM - katei (Yuta Saito)

From my further debugging, I found that CoreFoundation internally creates NSTaggedPointerString for small strings since macOS 13.

This C code can check a string is represented by tagged pointer in CoreFoundation.

```
// $ clang -g main.c -framework CoreFoundation
// $ ./a.out # on macOS ~12
// CFStringCreateWithBytesNoCopy("/") =     0x600003d94780 (tagged = NO)
// $ ./a.out # on macOS 13
// CFStringCreateWithBytesNoCopy("/") = 0xab98bd5bc57e5fed (tagged = YES)

#include <CoreFoundation/CFString.h>
#include <stdio.h>

// Ref: https://github.com/apple-oss-distributions/objc4/blob/8701d5672d3fd3cd817aeb84db1077aafe1a1604/runtime
/objc-internal.h#L446-L482
#if (TARGET_OS_OSX || TARGET_OS_MACCATALYST) && __x86_64__
# define _OBJC_TAG_MASK 1ULL
#else
# define _OBJC_TAG_MASK 0x8000000000000000ULL
#endif

#define _objc_isTaggedPointer(x) (((uintptr_t)(x)&_OBJC_TAG_MASK) != 0)

int main(void) {
  long len = 1;
  const char *ptr = "/";
  CFStringRef s = CFStringCreateWithBytesNoCopy(
      kCFAllocatorDefault, (const UInt8 *)ptr, len, kCFStringEncodingUTF8,
      FALSE, kCFAllocatorNull);
  printf("CFStringCreateWithBytesNoCopy(\"%s\") = %18p (tagged = %s)\n", ptr, s,
         _objc_isTaggedPointer(s) ? "YES" : "NO");
  return 0;
}
```

In the above reproducible case in #note-12,

1. Process.daemon(false, true) performs chdir("/")
2. Dir.pwd calls CFStringCreateWithBytesNoCopy API with "/"
3. CFStringCreateWithBytesNoCopy returns NSTaggedPointerString because "/" is small enough to fit in the tagged string.
4. CFStringCreateMutableCopy calls NSMutableString's methods based on the tagged string.

So a reduced minimum repro can be:

```
pid = fork { p File.realpath "/" }
Process.waitpid(pid)
```

To avoid Objective-C runtime crash, we need to warm up Objective-C classes internally used in CFString family API by passing small string before fork().

Here is a patch to address it https://github.com/ruby/ruby/pull/6426

#16 - 09/26/2022 02:40 AM - hsbt (Hiroshi SHIBATA)

@katei (Yuta Saito) Thanks for your investigation. I confirmed to pass the failing tests with https://github.com/ruby/ruby/pull/6426 on macOS Ventura beta 8.

#17 - 09/26/2022 08:16 AM - hsbt (Hiroshi SHIBATA)

- Status changed from Assigned to Closed

I created changesets for Ventura and Xcode14 with https://bugs.ruby-lang.org/issues/19005

- https://github.com/ruby/ruby/pull/6440
- https://github.com/ruby/ruby/pull/6441
- https://github.com/ruby/ruby/pull/6442

#18 - 10/01/2022 09:12 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.7: REQUIRED, 3.0: REQUIRED, 3.1: REQUIRED to 2.7: REQUIRED, 3.0: REQUIRED, 3.1: DONE

merged https://github.com/ruby/ruby/pull/6440

#19 - 10/12/2022 09:21 AM - hsbt (Hiroshi SHIBATA)

- Backport changed from 2.7: REQUIRED, 3.0: REQUIRED, 3.1: DONE to 2.7: DONE, 3.0: DONE, 3.1: DONE

https://github.com/ruby/ruby/pull/6297 and https://github.com/ruby/ruby/pull/6298 have been merged.

#20 - 10/12/2022 09:46 PM - hsbt (Hiroshi SHIBATA)

- Description updated

#21 - 11/11/2022 09:21 PM - hsbt (Hiroshi SHIBATA)

- Description updated

#22 - 11/12/2022 04:24 AM - mrkn (Kenta Murata)

I tried to investigate bigdecimal's build problem on my MacBook Air. But, unfortunately, I couldn't. I used Xcode 14.0 on Ventura. I first tried with the following configure options: --enable-shared --with-openssl-dir=$(brew --prefix openssl@1.1) --with-libyaml-dir=$(brew --prefix libyaml). Then I tried just rbenv install 3.1.2. In both cases, these build succeeded.

Could you please give me mkmf.log in ext/bigdecimal directory?

#23 - 02/09/2023 07:44 AM - naruse (Yui NARUSE)

- Related to Feature #19422: Make `--enabled-shared` mandatory on macOS added

#24 - 07/23/2024 02:21 AM - hsbt (Hiroshi SHIBATA)

- Related to Bug #20631: Build failure with Xcode 16 beta and macOS 15 (Sequoia) Beta added