Ruby - Bug #14578

Forking a child process inside of a mutex crashes the ruby interpreter

03/05/2018 07:45 PM - bengovero (Ben Govero)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	2.5.0	**Backport:**	2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: DONE

Description

OS: Mac OS X 10.13.3 (High Sierra)
Affects ruby versions 2.5.0 and 2.6.0preview1

Issue **not** present in 2.4.1

Consider the following script. I contrived it as an experiment for a more complicated project. We have a resource that we want to synchronize access to, but we want to fork the process when actually using the resource. This script works in 2.4.1, but not in 2.5.0 or beyond.

```
class Synchronizer

  def initialize
    @mutex = Mutex.new
  end

  def use(&block)
    @mutex.synchronize do
      Process.fork do
        block.call
      end

      Process.wait
    end
  end

end

@s = Synchronizer.new

5.times do |i|
  Thread.new do
    @s.use do
      puts "block #{i}"
    end
  end
end

sleep 10
```

The error I get when the interpreter crashes is: [BUG] unexpected THREAD_KILLED

Is this a crazy implementation? Or a real bug?

Associated revisions

Revision 1b455428d311a7c2e562a72960a916f8be606b8f - 03/05/2018 10:58 PM - Eric Wong

thread.c: reset waitq of keeping mutexes in child

We must not maintain references to threads in the parent process
in any mutexes held by the child process.

- thread_sync.c (rb_mutex_cleanup_keeping_mutexes): new function
- thread.c (rb_thread_atfork): cleanup keeping mutexes
  [ruby-core:85940] [Bug #14578]

Fixes: r58604 (commit 3586c9e0876e784767a1c1adba9ebc2499fa0ec2)
("reduce rb_mutex_t size from 160 to 80 bytes on 64-bit")

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62668 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 1b455428 - 03/05/2018 10:58 PM - Eric Wong

thread.c: reset waitq of keeping mutexes in child

We must not maintain references to threads in the parent process
in any mutexes held by the child process.

- thread_sync.c (rb_mutex_cleanup_keeping_mutexes): new function
- thread.c (rb_thread_atfork): cleanup keeping mutexes
  [ruby-core:85940] [Bug #14578]

Fixes: r58604 (commit 3586c9e0876e784767a1c1adba9ebc2499fa0ec2)
("reduce rb_mutex_t size from 160 to 80 bytes on 64-bit")

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62668 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 59a6ed8cc10e11720cb3eaea8d4750d29cd61167 - 03/20/2018 02:08 AM - naruse (Yui NARUSE)

merge revision(s) 62668: [Backport #14578]

```
    thread.c: reset waitq of keeping mutexes in child

    We must not maintain references to threads in the parent process
    in any mutexes held by the child process.

    * thread_sync.c (rb_mutex_cleanup_keeping_mutexes): new function
    * thread.c (rb_thread_atfork): cleanup keeping mutexes
      [ruby-core:85940] [Bug #14578]

    Fixes: r58604 (commit 3586c9e0876e784767a1c1adba9ebc2499fa0ec2)
           ("reduce rb_mutex_t size from 160 to 80 bytes on 64-bit")
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_5@62852 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 59a6ed8c - 03/20/2018 02:08 AM - naruse (Yui NARUSE)

merge revision(s) 62668: [Backport #14578]

```
    thread.c: reset waitq of keeping mutexes in child

    We must not maintain references to threads in the parent process
    in any mutexes held by the child process.

    * thread_sync.c (rb_mutex_cleanup_keeping_mutexes): new function
    * thread.c (rb_thread_atfork): cleanup keeping mutexes
      [ruby-core:85940] [Bug #14578]

    Fixes: r58604 (commit 3586c9e0876e784767a1c1adba9ebc2499fa0ec2)
           ("reduce rb_mutex_t size from 160 to 80 bytes on 64-bit")
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_5@62852 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 0fd53f519fb37b1dfe37be9f53ebc7889f405114 - 12/22/2018 01:41 AM - Eric Wong

thread_sync.c (rb_mutex_t): eliminate fork_gen

The true bug fork_gen was hiding was rb_mutex_abandon_locking_mutex
failing to unconditionally clear the waitq of mutexes it was
waiting on. So we fix rb_mutex_abandon_locking_mutex, instead,
and eliminate rb_mutex_cleanup_keeping_mutexes.

This commit was tested heavily on a single-core Pentium-M which
was my most reliable reproducer of the "crash.rb" script from
[Bug #15383]

[Bug #14578] [Bug #15383]

Note: [Bug #15430] turned out to be an entirely different
problem: RLIMIT_NPROC limit was hit on the CI VMs.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@66489 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 0fd53f519fb37b1dfe37be9f53ebc7889f405114 - 12/22/2018 01:41 AM - Eric Wong

thread_sync.c (rb_mutex_t): eliminate fork_gen

The true bug fork_gen was hiding was rb_mutex_abandon_locking_mutex
failing to unconditionally clear the waitq of mutexes it was
waiting on. So we fix rb_mutex_abandon_locking_mutex, instead,
and eliminate rb_mutex_cleanup_keeping_mutexes.

This commit was tested heavily on a single-core Pentium-M which
was my most reliable reproducer of the "crash.rb" script from
[Bug #15383]

[Bug #14578] [Bug #15383]

Note: [Bug #15430] turned out to be an entirely different
problem: RLIMIT_NPROC limit was hit on the CI VMs.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@66489 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 0fd53f51 - 12/22/2018 01:41 AM - Eric Wong

thread_sync.c (rb_mutex_t): eliminate fork_gen

The true bug fork_gen was hiding was rb_mutex_abandon_locking_mutex
failing to unconditionally clear the waitq of mutexes it was
waiting on. So we fix rb_mutex_abandon_locking_mutex, instead,
and eliminate rb_mutex_cleanup_keeping_mutexes.

This commit was tested heavily on a single-core Pentium-M which
was my most reliable reproducer of the "crash.rb" script from
[Bug #15383]

[Bug #14578] [Bug #15383]

Note: [Bug #15430] turned out to be an entirely different
problem: RLIMIT_NPROC limit was hit on the CI VMs.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@66489 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 03/05/2018 09:51 PM - normalperson (Eric Wong)

ben.govero@gmail.com wrote:

> https://bugs.ruby-lang.org/issues/14578

My fault. r58604 ("reduce rb_mutex_t size from 160 to 80 bytes on 64-bit")
Hope I can fix it soon before I'm offline.

#2 - 03/05/2018 10:57 PM - normalperson (Eric Wong)

- Backport changed from 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: UNKNOWN to 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: REQUIRED

#3 - 03/05/2018 10:58 PM - Anonymous

- Status changed from Open to Closed

Applied in changeset trunk|r62668.

---

thread.c: reset waitq of keeping mutexes in child

We must not maintain references to threads in the parent process
in any mutexes held by the child process.

- thread_sync.c (rb_mutex_cleanup_keeping_mutexes): new function
- thread.c (rb_thread_atfork): cleanup keeping mutexes

> [ruby-core:85940] [Bug #14578]

Fixes: r58604 (commit 3586c9e0876e784767a1c1adba9ebc2499fa0ec2)
("reduce rb_mutex_t size from 160 to 80 bytes on 64-bit")

#4 - 03/05/2018 11:03 PM - normalperson (Eric Wong)

Eric Wong wrote:

> ben.govero@gmail.com wrote:
>
> > https://bugs.ruby-lang.org/issues/14578
>
> My fault. r58604 ("reduce rb_mutex_t size from 160 to 80 bytes on 64-bit")
> Hope I can fix it soon before I'm offline.

r62668 works for me; care to give it a shot?

#5 - 03/20/2018 02:08 AM - naruse (Yui NARUSE)

- Backport changed from 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: REQUIRED to 2.3: UNKNOWN, 2.4: UNKNOWN, 2.5: DONE

ruby_2_5 r62852 merged revision(s) 62668.

Files

ruby_2018-03-05-133827-1_bens-mac.crash	36.3 KB	03/05/2018	bengovero (Ben Govero)
ruby_2018-03-05-133827_bens-mac.crash	36.3 KB	03/05/2018	bengovero (Ben Govero)
ruby_2018-03-05-133827-2_bens-mac.crash	36.3 KB	03/05/2018	bengovero (Ben Govero)