

Ruby - Bug #5768

TestRequire#test_race_exception

12/17/2011 10:48 AM - naruse (Yui NARUSE)

Status:Closed Priority:Normal Assignee:kosaki (Motohiro KOSAKI) Target version:2.0.0 ruby -v:-	Backport:
Description require diff --git a/test/ruby/test_require.rb b/test/ruby/test_require.rb index 9186a6f..262a5ef 100644 --- a/test/ruby/test_require.rb +++ b/test/ruby/test_require.rb @@ -352,7 +352,7 @@ class TestRequire < Test::Unit::TestCase TestRequire.scratch << :pre Thread.pass until t2 = TestRequire.scratch[1] Thread.pass until t2.stop? -open(FILES, "w") { f f.puts "TestRequire.scratch << :post" +open(FILES, "w") { f f.puts "TestRequire.scratch << :post"; f.puts "t1,t2=TestRequire.scratch[1, 2];if Thread.current == t2; Thread.pass until t1.stopped?; end" raise "con1" EOS tmp.close @@ -364,6 +364,7 @@ raise "con1" t2_res = nil t1 = Thread.new do begin require(path) rescue RuntimeError @@ -389,8 +390,8 @@ raise "con1" assert_nothing_raised(ThreadError, bug5754) {t1.join} assert_nothing_raised(ThreadError, bug5754) {t2.join} • assert_equal(true, (t1_res ^ t2_res), bug5754) • assert_equal([:pre, t2, :post, :t2, :t1], scratch, bug5754) • assert_equal(true, (t1_res ^ t2_res), bug5754 + " t1:#{t1_res} t2:#{t2_res}") • assert_equal([:pre, t1, t2, :post, :t2, :t1], scratch, bug5754) ensure tmp.close(true) if tmp end	
Related issues: Related to Ruby - Bug #5754: Double require bug in 1.9.3 Closed12/13/2011	

Associated revisions

Revision 82ab1e18 - 12/30/2011 10:25 PM - naruse (Yui NARUSE)

- thread.c (rb_barrier_waiting): save the number of waiting threads in RBASIC()->flags. [ruby-dev:45002] [Bug #5768]

- thread.c (rb_barrier_destroy): ditto.

git-svn-id: [svn+ssh://ci.ruby-lang.org/ruby/trunk@34163](https://ci.ruby-lang.org/ruby/trunk@34163) b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 12/19/2011 10:58 AM - nagachika (Tomoyuki Chikanaga)

```
bug5754 .rb t2 Thread.current scratch
```

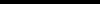
```
diff --git a/test/ruby/test_require.rb b/test/ruby/test_require.rb
index 9186a6f..02f7efe 100644
--- a/test/ruby/test_require.rb
+++ b/test/ruby/test_require.rb
@@ -349,10 +349,11 @@ class TestRequire < Test::Unit::TestCase
  tmp = Tempfile.new(%w"bug5754 .rb")
  path = tmp.path
  tmp.print <<-EOS
+Thread.pass until TestRequire.scratch[1]
  TestRequire.scratch << :pre
-Thread.pass until t2 = TestRequire.scratch[1]
+ t2 = TestRequire.scratch[0,2].reject{|t| t == Thread.current}[0]
  Thread.pass until t2.stop?
- open(FILE, "w") {|f| f.puts "TestRequire.scratch << :post"}
+ open(FILE, "w") {|f| f.puts "TestRequire.scratch << :post"; f.puts "t1,t2=TestRequire.scratch[0, 2];if Thread.current == t2; Thread.pass until
  t1.stopped?; end"}
  raise "con1"
EOS
tmp.close
@@ -364,6 +365,8 @@ raise "con1"
  t2 res = nil
```

```
t1 = Thread.new do
```

- [REDACTED]
- [REDACTED]
begin
 require(path)
rescue RuntimeError

```
@@ -376,6 +379,7 @@ raise "con1"
end
```

```
t2 = Thread.new do
```

-  Thread.pass until scratch[0]
begin
 scratch << t2

```
@@ -389,8 +393,8 @@ raise "con1"
assert_nothing_raised(ThreadError, bug5754) {t1.join}
assert_nothing_raised(ThreadError, bug5754) {t2.join}
```

- `assert_equal(true, (t1_res ^ t2_res), bug5754)`
- `assert_equal([:pre, t2, :post, t2, :t1], scratch, bug5754)`
- `assert_equal(true, (t1_res ^ t2_res), bug5754 + " t1:#{t1_res} t2:#{t2_res}")`
- `assert_equal([t1, t2, :pre, :post, t2, :t1], scratch, bug5754)`
`ensure`
`tmp.close(true) if tmp`
`end`

```
test race exception ThreadError (require (require))
```

1. Failure:

```
<#<ThreadError: Attempt to unlock a mutex which is locked by another thread>>.
```

```
diff --git a/thread.c b/thread.c
index ced10c2..c46653b 100644
--- a/thread.c
+++ b/thread.c
@@ -3409,7 +3409,6 @@ lock_func(rb_thread_t *th, rb_mutex_t *mutex, int timeout_ms)
int interrupted = 0;
int err = 0;
```

- mutex->cond_waiting++;
for (;;) {
if (!mutex->th) {
mutex->th = th;
@@ -3438,7 +3437,6 @@ lock_func(rb_thread_t *th, rb_mutex_t *mutex, int timeout_ms)
err = 0;
}
}

- mutex->cond_waiting--;

return interrupted;
}
@@ -3493,10 +3491,12 @@ rb_mutex_lock(VALUE self)
if (vm_living_thread_num(th->vm) == th->vm->sleeper) {
timeout_ms = 100;
}

- ```
GVL_UNLOCK_BEGIN();
interrupted = lock_func(th, mutex, timeout_ms);
native_mutex_unlock(&mutex->lock);
GVL_UNLOCK_END();
```

- ```
reset_unblock_function(th, &oldubf);
```

```
@@ -3727,9 +3727,11 @@ rb_barrier_release(VALUE self)
{
VALUE mutex = GetBarrierPtr(self);
rb_mutex_t *m;
```

- rb_mutex_unlock(mutex);
- int waiting;
GetMutexPtr(mutex, m);
- return m->cond_waiting > 0 ? Qtrue : Qfalse;
- waiting = m->cond_waiting;
- rb_mutex_unlock(mutex);
- return waiting > 0 ? Qtrue : Qfalse;
- }

```
/*
@@ -3740,10 +3742,12 @@ rb_barrier_destroy(VALUE self)
{
VALUE mutex = GetBarrierPtr(self);
rb_mutex_t *m;
```

- int waiting;
DATA_PTR(self) = 0;
- rb_mutex_unlock(mutex);
GetMutexPtr(mutex, m);
- return m->cond_waiting > 0 ? Qtrue : Qfalse;
- waiting = m->cond_waiting;
- rb_mutex_unlock(mutex);
- return waiting > 0 ? Qtrue : Qfalse;
- }

int

#3 - 12/20/2011 02:10 PM - kosaki (Motohiro KOSAKI)

- Category set to core
- Assignee changed from nobu (Nobuyoshi Nakada) to kosaki (Motohiro KOSAKI)
- Target version set to 2.0.0

lock_func rb_mutex_lock

#4 - 12/20/2011 03:59 PM - nobu (Nobuyoshi Nakada)

- ruby -v changed from ruby 2.0.0dev (2011-12-16 trunk 34058) [x86_64-freebsd9.0] to -

(11/12/20 13:43), Yui NARUSE wrote:

```
lock rb_barrier_release/rb_barrier_destroy rb_mutex_lock
rb_mutex_unlock mutex->cond_waiting
GVL_UNLOCK_BEGIN()
```

```
rb_mutex_lock GVL
rb_barrier_release/rb_barrier_destroy mutex
Barrier
```

```
diff --git a/test/ruby/test_require.rb b/test/ruby/test_require.rb
index 9186a6f..f1d8d12 100644
--- a/test/ruby/test_require.rb
+++ b/test/ruby/test_require.rb
@@ -350,9 +350,18 @@ class TestRequire < Test::Unit::TestCase
  path = tmp.path
  tmp.print <<-EOS
  TestRequire.scratch << :pre
  -Thread.pass until t2 = TestRequire.scratch[1]
  +TestRequire.scratch << Thread.current
  +Thread.pass until t2 = TestRequire.scratch[2]
  Thread.pass until t2.stop?
  -open(FILES, "w") {|f| f.puts "TestRequire.scratch << :post"}
  +open(FILES, "w") do |f|
    • f.puts "t1, t2 = TestRequire.scratch[1, 2]"
    • f.puts "if Thread.current == t2"
    • f.puts " TestRequire.scratch << :post"
    • f.puts " until t1.stop?"
    • f.puts " Thread.pass"
    • f.puts " end"
    • f.puts "end"
    +end
    raise "con1"
```

```
t2 require
t1, t2
```

```
--
--- Bug
--- Bug
  2 2
```

#5 - 12/20/2011 04:55 PM - naruse (Yui NARUSE)

Nobuyoshi Nakada wrote:

```
rb_barrier_release/rb_barrier_destroy mutex
Barrier
```

```
t1 t2.stop? == true require
```

```

00000000 GVL_UNLOCK_BEGIN 0000 mutex->cond_waiting++ 0000000000000000
mutex->cond_waiting 00000000 00000000000000000000000000000000
0000000000000000000000000000000000000000000000000000000000000000
00000000 load.c 0000 load unlock 0000 rb barrier destroy/rb barrier destroy 0000 st delete 0000000000000000

```

```
diff --git a/test/ruby/test_require.rb b/test/ruby/test_require.rb
index 9186a6f..f1d8d12 100644
--- a/test/ruby/test_require.rb
+++ b/test/ruby/test_require.rb
@@ -350,9 +350,18 @@ class TestRequire < Test::Unit::TestCase
   path = tmp.path
   tmp.print <<-EOS
   TestRequire.scratch << :pre
   -Thread.pass until t2 = TestRequire.scratch[1]
   +TestRequire.scratch << Thread.current
   +Thread.pass until t2 = TestRequire.scratch[2]
   Thread.pass until t2.stop?
   -open(FILES, "w") {|f| f.puts "TestRequire.scratch << :post"}
   +open(FILES, "w") do |f|
```

- f.puts "t1, t2 = TestRequire.scratch[1, 2]"
- f.puts "if Thread.current == t2"
- f.puts " TestRequire.scratch << :post"
- f.puts " until t1.stop?"
- f.puts " Thread.pass"
- f.puts " end"
- f.puts "end"
- +end
- raise "con1"

```

000000t2000000000000000000require000
00000000t1,t200000000000000000000000
000

```

```
00000000 :post 00000000000000000000000000000000
```

#6 - 01/05/2012 10:58 AM - naruse (Yui NARUSE)

[illegible]

#7 - 11/26/2012 08:05 PM - kosaki (Motohiro KOSAKI)

- Status changed from Assigned to Closed

[illegible]