Ruby master - Bug #13876

Tempfile's finalizer can be interrupted by a Timeout exception which can cause the process to hang

09/06/2017 09:15 PM - jrafanie (Joe Rafaniello)

Status:	Open		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.4.1p111 (2017-03-22 revision 58053) [x86_64-darwin15]	**Backport:**	2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

Ruby hangs if a Timeout is raised when a Tempfile's finalizer is run.

This also happens on ruby 2.3.4 on OSX. I could only recreate this on linux once but not reliably. Maybe the finalizer runs at a different and unpredictable time on Linux.

See the following script:

```
require 'tempfile'
require 'timeout'
Tempfile.new("x")

# this empty method is needed for some reason, maybe scope/GC related
def stuff
  ""
end

2.times do |i|
  begin
    Timeout.timeout(1e-9) do
      loop do
        begin
          "result"
        ensure
        end
      end
    end
  rescue Timeout::Error
  end
end
```

When run, I get the following output and it hangs ~4 out of 5 times on OSX:

```
$ ruby -d test.rb
Exception `LoadError' at /Users/joerafaniello/.rubies/ruby-2.4.1/lib/ruby/2.4.0/rubygems.rb:1345 -
 cannot load such file -- rubygems/defaults/operating_system
Exception `LoadError' at /Users/joerafaniello/.rubies/ruby-2.4.1/lib/ruby/2.4.0/rubygems.rb:1354 -
 cannot load such file -- rubygems/defaults/ruby
removing /var/folders/fq/blrz820d3qz7nm7vj8mbtfs40000gq/T/x20170906-72981-1ttdgpr...
<HANGS>
```

When it doesn't hang (~1 in 5 tries), it looks like this (the exception seems to be raised after the file cleanup is done):

```
$ ruby -d test.rb
Exception `LoadError' at /Users/joerafaniello/.rubies/ruby-2.4.1/lib/ruby/2.4.0/rubygems.rb:1345 -
 cannot load such file -- rubygems/defaults/operating_system
Exception `LoadError' at /Users/joerafaniello/.rubies/ruby-2.4.1/lib/ruby/2.4.0/rubygems.rb:1354 -
 cannot load such file -- rubygems/defaults/ruby
removing /var/folders/fq/blrz820d3qz7nm7vj8mbtfs40000gq/T/x20170906-72963-164cw8j...
done
Exception `Timeout::Error' at /Users/joerafaniello/.rubies/ruby-2.4.1/lib/ruby/2.4.0/timeout.rb:11
4 - execution expired
```

```
Exception `Timeout::Error' at /Users/joerafaniello/.rubies/ruby-2.4.1/lib/ruby/2.4.0/timeout.rb:11
4 - execution expired
```

When it hangs, it seems to be this code:
https://github.com/ruby/ruby/blob/0f25c6d7d59ff9744d1ca5dc908cc12c8ce79e25/lib/tempfile.rb#L253-L261

History

#1 - 09/12/2017 01:27 AM - kernigh (George Koehler)

The bug happens when an async error (from Thread#raise) interrupts a finalizer. Ruby seems to rescue errors from finalizers, so the error never reaches the main code. In the script by Joe Rafaniello, the Timeout::Error disappears, so the infinite loop never times out, and the program hangs.

Because of this bug, we can't rely on Timeout in any program that uses ObjectSpace::define_finalizer. We might get unlucky, as the garbage collector might decide to run a finalizer when the Timeout::Error happens. Then the error disappears and our code never times out.

This script reproduces the bug more reliably:

```
GC.disable
fzer = proc do |id|
  puts "Before sleep #{id}"
  sleep 0.2
  puts "After sleep #{id}"
end
2.times do
  o = Object.new
  ObjectSpace.define_finalizer(o, fzer)
end

class MyError < RuntimeError; end
begin
  main_th = Thread.current
  Thread.new do
    sleep 0.1
    main_th.raise(MyError)
  end
  GC.start
  puts "After GC"
  sleep
rescue MyError
  puts "Rescued #$!"
end
```

The GC can fail to destroy the object. The C compiler that compiled Ruby might leave an unused VALUE on the stack, so the GC marks the object and doesn't destroy it. To work around this, I make 2 objects in a loop, and hope that the GC can destroy at least one of them.

To reproduce the bug, I use GC.disable and GC.start to simulate some bad luck and run the finalizer at a bad time. My main thread gets interrupted with MyError after 0.1 seconds, but the finalizer takes 0.2 seconds to run. So MyError is almost certain to interrupt the finalizer. The bug happens as MyError never reaches my main code. I expect my main thread to rescue MyError after 0.1 seconds, but it never happens, so my main thread sleeps forever.

```
$ ruby -v
ruby 2.5.0dev (2017-09-11 trunk 59840) [x86_64-openbsd6.1]
$ ruby -d my-error-13876.rb
Exception `LoadError' at /home/kernigh/prefix/lib/ruby/2.5.0/rubygems.rb:1346 - cannot load such file -- rubyg
ems/defaults/operating_system
Exception `LoadError' at /home/kernigh/prefix/lib/ruby/2.5.0/rubygems.rb:1355 - cannot load such file -- rubyg
ems/defaults/ruby
Exception `Gem::MissingSpecError' at /home/kernigh/prefix/lib/ruby/2.5.0/rubygems/dependency.rb:308 - Gem::Mis
singSpecError
Before sleep 13867240187460
Exception `MyError' at my-error-13876.rb:4 - MyError
After GC
<HANGS>
```

MyError must have interrupted my finalizer, because it never printed the message After sleep 13867240187460.