Ruby - Bug #4681

Timeout.timeout doesn't actually time out?

05/13/2011 06:44 AM - meta (mathew murphy)

Status:	Rejected		
Priority:	Normal		
Assignee:	akr (Akira Tanaka)		
Target version:	1.9.3		
ruby -v:	1.9.2p180	**Backport:**	

Description

Documentation for Timeout says "A way of performing a potentially long-running operation in a thread, and terminating its execution if it hasn‘t finished by a fixed amount of time."

This doesn't actually seem to be what it does.

Example code:

```
#!/usr/bin/ruby
```

encoding: UTF-8

```
require 'open3'
require 'timeout'

puts "Time out after 2 seconds. Count them..."
result = Timeout.timeout(2) do
stdin, stdout, stderr = Open3.capture3("sleep 30")
output = stdout.read + "\n" + stderr.read
end
```

On my systems, this doesn't time out after 2 seconds. Instead, it runs for 30 seconds, and then throws an error saying it took longer than 2 seconds.

History

#1 - 05/13/2011 08:23 AM - normalperson (Eric Wong)

mathew murphy meta@pobox.com wrote:

> This doesn't actually seem to be what it does.
>
> Example code:
>
> ```
> #!/usr/bin/ruby
> ```
>
> # encoding: UTF-8
>
> ```
> require 'open3'
> require 'timeout'
>
> puts "Time out after 2 seconds. Count them..."
> result = Timeout.timeout(2) do
> stdin, stdout, stderr = Open3.capture3("sleep 30")
> output = stdout.read + "\n" + stderr.read
> end
> ```
>
> On my systems, this doesn't time out after 2 seconds. Instead, it runs
> for 30 seconds, and then throws an error saying it took longer than 2
> seconds.

It's because open3 has an ensure block where it does Thread#join and
that waits forever, so the timeout thread raised to unblock the

main thread, and then it got stuck again inside the ensure block.

```
diff --git a/lib/open3.rb b/lib/open3.rb
index b65cb19..d335f9f 100644
--- a/lib/open3.rb
+++ b/lib/open3.rb
@@ -207,6 +207,7 @@ module Open3
begin
return yield(*result)
ensure
```

- [redacted]

  ```
  parent_io.each{|io| io.close unless io.closed?}
  wait_thr.join
  end
  @@ -702,6 +703,7 @@ module Open3
  begin
  return yield(*result)
  ensure
  ```

- [redacted]

  ```
  parent_io.each{|io| io.close unless io.closed?}
  wait_thrs.each {|t| t.join }
  end
  ```

You can work around it by having an extra timeout block (ugly):

```
puts "Time out after 2 seconds. Count them..."
Timeout.timeout(2) do
result = Timeout.timeout(2) do
stdin, stdout, stderr = Open3.capture3("sleep 30")
output = stdout.read + "\n" + stderr.read
end
end
```

But IMHO, timeout is a very fragile module and shouldn't be relied on.

--
Eric Wong

#2 - 06/26/2011 06:58 PM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

- Assignee set to akr (Akira Tanaka)

#3 - 06/26/2011 07:06 PM - nahi (Hiroshi Nakamura)

- Target version set to 1.9.3

#4 - 07/10/2011 03:52 PM - kosaki (Motohiro KOSAKI)

- Status changed from Assigned to Rejected

I think Eric described the reason and workaround at commetn#1. IOW, it's spec unfortunately.

#5 - 12/08/2011 01:37 PM - meta (mathew murphy)

Just noting for the benefit of anyone else passing by that I posted a way to do popen3 with timeout at
https://gist.github.com/1032297