Ruby - Bug #4169

Forked processes locked, won't exit

12/18/2010 03:52 PM - igrigorik (Ilya Grigorik)

Status:	Closed		
Priority:	Normal		
Assignee:	kosaki (Motohiro KOSAKI)		
Target version:	1.9.2		
ruby -v:	ruby 1.9.2p0 (2010-08-18 revision 29036) [i686-linux]	**Backport:**	

Description

=begin
Unfortunately can't pinpoint the exact problem, but this issue has been brought up by a number of people under 1.9.1 and 1.9.2 when running "god":
https://github.com/mojombo/god/issues#issue/6 - see full thread

Example traces of stuck process: https://gist.github.com/9ccfda78c52356e74291

Any other data I could provide to help identify the problem?
=end

Associated revisions

Revision 97da02b2 - 12/20/2010 03:16 PM - kosaki (Motohiro KOSAKI)

- thread.c (thread_cleanup_func): Don't touch native threading
  resource at fork. Sadly this is purely bandaid. We need to
  implement proper fix later. [Bug #4169] [ruby-core:33767]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@30272 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 4ca870d0 - 07/03/2011 12:23 PM - yugui (Yuki Sonoda)

merges r30272 from trunk into ruby_1_9_2.

```
    * thread.c (thread_cleanup_func): Don't touch native threading
      resource at fork. Sadly this is purely bandaid. We need to
      implement proper fix later. [Bug #4169] [ruby-core:33767]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_2@32375 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 12/19/2010 11:01 PM - kosaki (Motohiro KOSAKI)

- Category set to core

- Status changed from Open to Assigned

- Assignee set to kosaki (Motohiro KOSAKI)

- Priority changed from Normal to 5

=begin
I hope fix this issue ASAP really. Do you have a stable reproduce way?
=end

#2 - 12/19/2010 11:13 PM - kosaki (Motohiro KOSAKI)

=begin
The following reporter provided backtrace indicate this is ruby's bug.
I don't think this is god specific issue. And I double rb_f_fork() is broken now.

```
(gdb) bt
#0  0x00896f5d in pthread_cond_destroy@@GLIBC_2.3.2 () from /lib/libpthread.so.0
#1  0x0817087f in native_thread_destroy (key=152643360, val=0, current_th=152549360) at thread_pthread.c:183
#2  thread_cleanup_func (key=152643360, val=0, current_th=152549360) at thread.c:398
```

#3 terminate_atfork_i (key=152643360, val=0, current_th=152549360) at thread.c:2773
#4 0x080fedd6 in st_foreach (table=0x8c32080, func=0x8170800 <terminate_atfork_i>, arg=152549360) at st.c:778
#5 0x08173688 in rb_thread_atfork_internal (atfork=0x8170800 <terminate_atfork_i>) at thread.c:2754
#6 0x08173740 in rb_thread_atfork () at thread.c:2781
#7 0x080bf768 in rb_f_fork (obj=154401340) at process.c:2623
#8 0x08157fed in vm_call_cfunc (th=0x917b7f0, cfp=0xb72eedcc, num=0, blockptr=0xb72eede1, flag=8, id=8096, me=0x8c24178, recv=154401340) at vm_insnhelper.c:401
#9 vm_call_method (th=0x917b7f0, cfp=0xb72eedcc, num=0, blockptr=0xb72eede1, flag=8, id=8096, me=0x8c24178, recv=154401340) at vm_insnhelper.c:523
#10 0x0815be7d in vm_exec_core (th=0x917b7f0, initial=) at insns.def:1006
#11 0x08162cec in vm_exec (th=0x917b7f0) at vm.c:1145
#12 0x0816822b in invoke_block_from_c (th=0x917b7f0, block=0xb72eef98, self=154401620, argc=0, argv=0x0, blockptr=0x0, cref=0x0) at vm.c:557
#13 0x08168888 in vm_yield () at vm.c:587
#14 rb_yield_0 () at vm_eval.c:731
#15 loop_i () at vm_eval.c:789
#16 0x0805b5b2 in rb_rescue2 (b_proc=0x8168840 <loop_i>, data1=0, r_proc=0, data2=0) at eval.c:646
#17 0x08154814 in rb_f_loop (self=154401620) at vm_eval.c:817
#18 0x08157fed in vm_call_cfunc (th=0x917b7f0, cfp=0xb72eef84, num=0, blockptr=0xb72eef99, flag=8, id=2856, me=0x8beec68, recv=154401620) at vm_insnhelper.c:401
#19 vm_call_method (th=0x917b7f0, cfp=0xb72eef84, num=0, blockptr=0xb72eef99, flag=8, id=2856, me=0x8beec68, recv=154401620) at vm_insnhelper.c:523
#20 0x0815be7d in vm_exec_core (th=0x917b7f0, initial=) at insns.def:1006
#21 0x08162cec in vm_exec (th=0x917b7f0) at vm.c:1145
#22 0x0816822b in invoke_block_from_c (th=0x917b7f0, block=0x9280c08, self=154401620, argc=0, argv=0x933fa80, blockptr=0x0, cref=0x0) at vm.c:557
#23 0x081685c6 in rb_vm_invoke_proc (th=0x917b7f0, proc=0x9280c08, self=154401620, argc=0, argv=0x933fa80, blockptr=0x0) at vm.c:603
#24 0x081767ce in thread_start_func_2 (th=0x917b7f0, stack_start=) at thread.c:441
#25 0x08176917 in thread_start_func_1 (th_ptr=0x917b7f0) at thread_pthread.c:369
#26 0x008932ea in start_thread () from /lib/libpthread.so.0
#27 0xb726e480 in ?? ()
#28 0xb726e480 in ?? ()
#29 0xb726e480 in ?? ()
#30 0xb726e480 in ?? ()
#31 0x00000000 in ?? ()
=end

#3 - 12/20/2010 12:57 AM - kosaki (Motohiro KOSAKI)

- File 0001-reset-thread-lock-state-at-fork.patch added

=begin
Hi Ilya,

Can you please try the attached patch?
=end

#4 - 12/21/2010 12:17 AM - kosaki (Motohiro KOSAKI)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

=begin
This issue was solved with changeset r30272.
Ilya, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

=end

Files

0001-reset-thread-lock-state-at-fork.patch	943 Bytes	12/20/2010	kosaki (Motohiro KOSAKI)