

Ruby master - Bug #15798

Fix threads not waking up on SIGINT when using UBF_TIMER_PTHREAD

04/26/2019 11:46 PM - jeremyevans0 (Jeremy Evans)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.7.0dev (2019-04-26 trunk e804fcb42c) [x86_64-openbsd]	Backport: 2.4: DONTNEED, 2.5: DONTNEED, 2.6: DONE
Description <p>When using UBF_TIMER_PTHREAD, the timer_thread_fn function will not signal the main thread with SIGVTALRM in cases where timer_thread is armed before consume_communication_pipe is called. This is because consume_communication_pipe will unarm the timer.</p> <p>Fix this by checking the return value of consume_communication_pipe. If it returns TRUE and the timer_thread is disarmed, then signal the main thread with SIGVTALRM.</p> <p>This fixes TestThread#test_thread_timer_and_interrupt. It also fixes the use of Ctrl+C/SIGINT in irb on OpenBSD, and I'm guessing other platforms that use UBF_TIMER_PTHREAD. Previously, you could hit Ctrl+C multiple times and it would do nothing until another key was pressed (even Backspace).</p> <p>If this is determined to be the correct fix, this should be backported to 2.6, but it is not a problem in earlier versions.</p>	

Associated revisions

Revision 1ef39d8d - 05/25/2019 12:50 AM - jeremyevans (Jeremy Evans)

Fix process not waking up on signals on OpenBSD

When using UBF_TIMER_PTHREAD (the UBF handler on OpenBSD), the timer_thread_fn function will not signal the main thread with SIGVTALRM in cases where timer_thread is armed before consume_communication_pipe is called. This is because consume_communication_pipe will unarm the timer.

Fix this by checking the return value of consume_communication_pipe. If it returns TRUE and the timer_thread is disarmed, then signal the main thread with SIGVTALRM.

On OpenBSD, this fixes TestThread#test_thread_timer_and_interrupt, and fixes hangs in TestProcess#test_execopts_redirect_open_fifo_interrupt_raise and TestProcess#test_execopts_redirect_open_fifo_interrupt_print. It also fixes the use of Ctrl+C/SIGINT in irb on OpenBSD. It does not cause any test failures on Linux when UBF_TIMER_PTHREAD is forced as the UBF handler.

Fixes [Bug #15798]

Revision 194a5525 - 08/03/2019 12:27 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 1ef39d8d099f145222b9352423af16a2bab6e05b: [Backport #15798]

Fix process not waking up on signals on OpenBSD

When using UBF_TIMER_PTHREAD (the UBF handler on OpenBSD), the timer_thread_fn function will not signal the main thread with SIGVTALRM in cases where timer_thread is armed before consume_communication_pipe is called. This is because consume_communication_pipe will unarm the timer.

Fix this by checking the return value of consume_communication_pipe.

If it returns TRUE and the timer_pthread is disarmed, then signal the main thread with SIGVTALRM.

On OpenBSD, this fixes TestThread#test_thread_timer_and_interrupt, and fixes hangs in TestProcess#test_execopts_redirect_open_fifo_interrupt_raise and TestProcess#test_execopts_redirect_open_fifo_interrupt_print. It also fixes the use of Ctrl+C/SIGINT in irb on OpenBSD. It does not cause any test failures on Linux when UBF_TIMER_PTHREAD is forced as the UBF handler.

Fixes [Bug #15798]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_6@67725 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 67725 - 08/03/2019 12:27 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 1ef39d8d099f145222b9352423af16a2bab6e05b: [Backport #15798]

Fix process not waking up on signals on OpenBSD

When using UBF_TIMER_PTHREAD (the UBF handler on OpenBSD), the timer_pthread_fn function will not signal the main thread with SIGVTALRM in cases where timer_pthread is armed before consume_communication_pipe is called. This is because consume_communication_pipe will unarm the timer.

Fix this by checking the return value of consume_communication_pipe. If it returns TRUE and the timer_pthread is disarmed, then signal the main thread with SIGVTALRM.

On OpenBSD, this fixes TestThread#test_thread_timer_and_interrupt, and fixes hangs in TestProcess#test_execopts_redirect_open_fifo_interrupt_raise and TestProcess#test_execopts_redirect_open_fifo_interrupt_print. It also fixes the use of Ctrl+C/SIGINT in irb on OpenBSD. It does not cause any test failures on Linux when UBF_TIMER_PTHREAD is forced as the UBF handler.

Fixes [Bug #15798]

History

#1 - 05/16/2019 03:54 PM - jeremyevans0 (Jeremy Evans)

I've tested on Linux when forcing UBF_TIMER_PTHREAD. While the patch does not appear to be necessary on Linux, there are no regressions in the tests when using it, so it should be safe. It does fix the annoying current behavior on OpenBSD, so I hope a committer can review and commit it.

#2 - 05/25/2019 12:51 AM - jeremyevans (Jeremy Evans)

- Status changed from Open to Closed

Applied in changeset [git|1ef39d8d099f145222b9352423af16a2bab6e05b](https://github.com/ruby/ruby/commit/1ef39d8d099f145222b9352423af16a2bab6e05b).

Fix process not waking up on signals on OpenBSD

When using UBF_TIMER_PTHREAD (the UBF handler on OpenBSD), the timer_pthread_fn function will not signal the main thread with SIGVTALRM in cases where timer_pthread is armed before consume_communication_pipe is called. This is because consume_communication_pipe will unarm the timer.

Fix this by checking the return value of consume_communication_pipe. If it returns TRUE and the timer_pthread is disarmed, then signal the main thread with SIGVTALRM.

On OpenBSD, this fixes TestThread#test_thread_timer_and_interrupt, and fixes hangs in TestProcess#test_execopts_redirect_open_fifo_interrupt_raise and TestProcess#test_execopts_redirect_open_fifo_interrupt_print. It also fixes the use of Ctrl+C/SIGINT in irb on OpenBSD. It does not cause any test failures on Linux when UBF_TIMER_PTHREAD is forced as the UBF handler.

Fixes [Bug #15798]

#3 - 08/03/2019 12:27 PM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.4: DONTNEED, 2.5: DONTNEED, 2.6: REQUIRED to 2.4: DONTNEED, 2.5: DONTNEED, 2.6: DONE

ruby_2_6_r67725 merged revision(s) 1ef39d8d099f145222b9352423af16a2bab6e05b.

Files

fix-pthread-timer.patch	2.43 KB	04/26/2019	jeremyevans0 (Jeremy Evans)
-------------------------	---------	------------	-----------------------------