Ruby trunk - Bug #13856

MinGW / mswin intermittent failure in test/socket/test_socket.rb

09/01/2017 02:42 AM - MSP-Greg (Greg L)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 2.5.0dev (2017-09-11 trunk 59829) [x64-mingw32]	**Backport:**	2.2: UNKNOWN, 2.3: REQUIRED, 2.4: DONE

Description

Recently, there have been two Appveyor fails, first and second.

Both listed the following as the error:

```
running file: C:/projects/ruby/test/socket/test_socket.rb

Some worker was crashed. It seems ruby interpreter's bug
or, a bug of test/unit/parallel.rb. try again without -j
option.
```

I have a similar intermittent error as a silent segfault in MinGW builds. I've attached the patch file I use, it patches the TestSocket#test_closed_read method, and simply adds sock.autoclose = false after sock is created.

I haven't done much Ruby socket coding, so I've never looked into the issue, but the test passes and is stable with the patch. For all I know, the patch may 'end-around' the whole point of the test.

I don't know if this helps identify the real issue or not. If not, feel free to close.

Associated revisions

Revision 60055 - 09/28/2017 01:43 PM - shirosaki

io.c: fix segfault with closing socket on Windows

- io.c (fptr_finalize_flush): add an argument to keep GVL.
- io.c (fptr_finalize): adjust for above change.
- io.c (io_close_fptr): closing without GVL causes another exception while raising exception in another thread. This causes segfault on Windows. Keep GVL while closing when another thread raises. [Bug #13856] [ruby-core:82602]

Revision 62690 - 03/07/2018 02:05 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 60055: [Backport #13856]

```
io.c: fix segfault with closing socket on Windows

* io.c (fptr_finalize_flush): add an argument to keep GVL.
* io.c (fptr_finalize): adjust for above change.
* io.c (io_close_fptr): closing without GVL causes another
  exception while raising exception in another thread. This causes
  segfault on Windows. Keep GVL while closing when another thread
  raises.
  [Bug #13856] [ruby-core:82602]
```

History

#1 - 09/15/2017 08:13 AM - h.shirosaki (Hiroshi Shirosaki)

- ruby -v set to ruby 2.5.0dev (2017-09-11 trunk 59829) [x64-mingw32]

Segmentation fault is caused by RBASIC_CLASS(err) access in hook_before_rewind() in vm.c:1667.
err is not a valid pointer.

Here is gdb output.

```
Run options: "--ruby=./miniruby.exe -I../snapshot/lib -I. -I.ext/common  ../snapshot/tool/runruby.rb --extout=
.ext --debugger -- --disable-gems" --excludes-dir=../snapshot/test/excludes --name=!/memory_leak/ -v -ntest_cl
osed_read

# Running tests:

[1/1] TestSocket#test_closed_read[New Thread 16292.0x331c]
[New Thread 16292.0x57f4]
[Thread 16292.0x331c exited with code 0]

Thread 9 received signal SIGSEGV, Segmentation fault.
[Switching to Thread 16292.0x57f4]
0x00000000680aad1a in hook_before_rewind (th=th@entry=0x33671d0, will_finish_vm_exec=will_finish_vm_exec@entry
=0, state=6, err=err@entry=0xc0000241,
    cfp=<optimized out>) at ../snapshot/vm.c:1667
1667        if (state == TAG_RAISE && RBASIC_CLASS(err) == rb_eSysStackError) {
(gdb) bt
#0  0x00000000680aad1a in hook_before_rewind (th=th@entry=0x33671d0, will_finish_vm_exec=will_finish_vm_exec@e
ntry=0, state=6, err=err@entry=0xc0000241,
    cfp=<optimized out>) at ../snapshot/vm.c:1667
#1  0x00000000680b6ffd in vm_exec (th=0x31e000, th@entry=0x33671d0) at ../snapshot/vm.c:2003
...
```

err seems to come from rb_threadptr_execute_interrupts() in thread.c.
Adding volatile as the following suppresses segfault on my test.
But this patch sometimes causes another error (TypeError: exception class/object expected).
I don't know the reason.

```
diff --git a/thread.c b/thread.c
index d706ee469b..fd1ee78933 100644
--- a/thread.c
+++ b/thread.c
@@ -2058,7 +2058,7 @@ rb_threadptr_execute_interrupts(rb_thread_t *th, int blocking_timing)

     /* exception from another thread */
     if (pending_interrupt && rb_threadptr_pending_interrupt_active_p(th)) {
-
        VALUE err = rb_threadptr_pending_interrupt_deque(th, blocking_timing ? INTERRUPT_ON_BLOCKING : INTERRUP
T_NONE);
+       volatile
VALUE err = rb_threadptr_pending_interrupt_deque(th, blocking_timing ? INTERRUPT_ON_BLOCKING : INTERRUPT_NONE)
;
        thread_debug("rb_thread_execute_interrupts: %"PRIdVALUE"\n", err);

        if (err == Qundef) {


  1) Failure:
TestSocket#test_closed_read [C:/Users/h.shirosaki/work/rubyinstaller2-packages/mingw-w64-ruby25/src/snapshot/t
est/socket/test_socket.rb:543]:
.
[IOError] exception expected, not.
Class: <TypeError>
Message: <"exception class/object expected">
---Backtrace---
C:/Users/h.shirosaki/work/rubyinstaller2-packages/mingw-w64-ruby25/src/snapshot/test/socket/test_socket.rb:537
:in `readline'
C:/Users/h.shirosaki/work/rubyinstaller2-packages/mingw-w64-ruby25/src/snapshot/test/socket/test_socket.rb:537
:in `block in test_closed_read'
---------------
```

#2 - 09/21/2017 10:36 AM - h.shirosaki (Hiroshi Shirosaki)

- File 0001-io.c-fix-segfault-with-closing-socket-on-MinGW.patch added

I found that another exception raises while executing rb_exc_raise() in rb_threadptr_execute_interrupts().
This reentering exception would cause segfault for some reason.
Socket is closed releasing GVL before rb_exc_raise() in rb_threadptr_execute_interrupts().
If keeping GVL with close, reentering exception and segfault is not raised on my test. The patch is attached.
This may be related to #4558?

#3 - 09/21/2017 11:39 AM - usa (Usaku NAKAMURA)

Shirosaki-san, your patch seems ok.
Could you check in?

#4 - 09/21/2017 01:49 PM - MSP-Greg (Greg L)

Shirosaki-san,

'Breakfast build' is finished, passed test-all, and the following also passed:

ruby runner.rb -I../lib -I. --repeat-count=50 --show-skip socket/test_socket.rb -ntest_closed_read

This has been an intermittent failure, but I've never had it complete 50 before.

Hence, along with usa, I believe you've fixed the issue. Thanks again for your work.

I'm not sure how everyone would prefer to be addressed. If you'd like to address me, please call me Greg. The first code I wrote was on a teletype...

EDIT: A little rushed this morning.

I applied the patch to:

```
ruby 2.5.0dev (2017-09-21 trunk 59985) [x64-mingw32]
```

I checked previous builds (**without** the patch), the following builds failed:

```
ruby 2.5.0dev (2017-09-20 trunk 59974) [x64-mingw32]  (previous build without patch)
```

```
ruby 2.4.2p198 (2017-09-14 revision 59899) [x64-mingw32]
```

```
ruby 2.3.5p376 (2017-09-14 revision 59905) [x64-mingw32]
```

I don't know if this could be backported or not...

#5 - 09/28/2017 01:43 PM - Anonymous

- Status changed from Open to Closed

Applied in changeset trunk|r60055.

---

io.c: fix segfault with closing socket on Windows

- io.c (fptr_finalize_flush): add an argument to keep GVL.
- io.c (fptr_finalize): adjust for above change.
- io.c (io_close_fptr): closing without GVL causes another exception while raising exception in another thread. This causes segfault on Windows. Keep GVL while closing when another thread raises. [Bug #13856]

#6 - 09/29/2017 11:18 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN to 2.2: UNKNOWN, 2.3: REQUIRED, 2.4: REQUIRED

#7 - 03/07/2018 02:05 PM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 2.2: UNKNOWN, 2.3: REQUIRED, 2.4: REQUIRED to 2.2: UNKNOWN, 2.3: REQUIRED, 2.4: DONE

ruby_2_4 r62690 merged revision(s) 60055.

Files

segv-test-socket-test_socket.rb.patch	782 Bytes	09/01/2017	MSP-Greg (Greg L)
0001-io.c-fix-segfault-with-closing-socket-on-MinGW.patch	1.76 KB	09/21/2017	h.shirosaki (Hiroshi Shirosaki)