Ruby master - Bug #5306

Application Hangs Due to Recent rb_thread_select Changes

09/10/2011 08:51 AM - cfis (Charlie Savage)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:	1.9.3	
ruby -v:	-	**Backport:**

Description

This commit:

4e9438bc9153f7a1f4ea0af85c8dbe359e1a55d8

Changed the implementation of rb_thread_select.

It causes eventmachine to hang on CentOS 5.5. Not sure what the issue is, but its easily reproduced by by running the test eventmachine/tests/test_epoll.rb.

We noticed this because it also causes the tweetstream gem to hang.

The same setup works on Fedora 14 and an up-to-date arch linux. Specific version information included below.

We temporarily fixed this by reverting the commit.

Since Centos is a common production environment (and the one we are using), this seems to us a blocker for 1.9.3.

We are happy to provide any additional information or test fixes.

Thanks - Charlie

We are running this version of CentOS:

Linux app1.zerista.com 2.6.18-238.19.1.el5.centos.plus #1 SMP Mon Jul 18 10:05:09 EDT 2011 x86_64 x86_64 x86_64 GNU/Linux

And this version of Fedora:

Linux ammonite.internal.zerista.com 2.6.35.14-95.fc14.x86_64 #1 SMP Tue Aug 16 21:01:58 UTC 2011 x86_64 x86_64 x86_64 GNU/Linux

And this version of eventmachine:

eventmachine (1.0.0.beta.3)

And this version of tweetstream:

tweetstream (1.0.4)

Related issues:

Related to Backport193 - Backport #5299: Segmentation fault when using TweetS...	**Closed**	**09/09/2011**

Associated revisions

Revision 2c9375ba - 09/12/2011 11:36 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_thread_select): fix to ignore an argument modification of rb_thread_fd_select(). based on a patch by Eric Wong. [Bug #5306] [ruby-core:39435]

- thread.c (rb_fd_rcopy): New. for reverse fd copy.

- test/-ext-/old_thread_select/test_old_thread_select.rb (test_old_select_false_positive): test for bug5306.

- ext/-test-/old_thread_select/old_thread_select.c (fdset2array):
  New. convert fdsets to array.

- ext/-test-/old_thread_select/old_thread_select.c (old_thread_select):
  return 'read', 'write', 'except' argument of rb_thread_select()
  to ruby script.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@33256 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 33256 - 09/12/2011 11:36 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_thread_select): fix to ignore an argument modification of rb_thread_fd_select(). based on a patch by Eric Wong. [Bug #5306] [ruby-core:39435]

- thread.c (rb_fd_rcopy): New. for reverse fd copy.

- test/-ext-/old_thread_select/test_old_thread_select.rb
  (test_old_select_false_positive): test for bug5306.

- ext/-test-/old_thread_select/old_thread_select.c (fdset2array):
  New. convert fdsets to array.

- ext/-test-/old_thread_select/old_thread_select.c (old_thread_select):
  return 'read', 'write', 'except' argument of rb_thread_select()
  to ruby script.

Revision 33256 - 09/12/2011 11:36 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_thread_select): fix to ignore an argument modification of rb_thread_fd_select(). based on a patch by Eric Wong. [Bug #5306] [ruby-core:39435]

- thread.c (rb_fd_rcopy): New. for reverse fd copy.

- test/-ext-/old_thread_select/test_old_thread_select.rb
  (test_old_select_false_positive): test for bug5306.

- ext/-test-/old_thread_select/old_thread_select.c (fdset2array):
  New. convert fdsets to array.

- ext/-test-/old_thread_select/old_thread_select.c (old_thread_select):
  return 'read', 'write', 'except' argument of rb_thread_select()
  to ruby script.

Revision 33256 - 09/12/2011 11:36 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_thread_select): fix to ignore an argument modification of rb_thread_fd_select(). based on a patch by Eric Wong. [Bug #5306] [ruby-core:39435]

- thread.c (rb_fd_rcopy): New. for reverse fd copy.

- test/-ext-/old_thread_select/test_old_thread_select.rb
  (test_old_select_false_positive): test for bug5306.

- ext/-test-/old_thread_select/old_thread_select.c (fdset2array):
  New. convert fdsets to array.

- ext/-test-/old_thread_select/old_thread_select.c (old_thread_select):
  return 'read', 'write', 'except' argument of rb_thread_select()
  to ruby script.

Revision 33256 - 09/12/2011 11:36 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_thread_select): fix to ignore an argument modification of rb_thread_fd_select(). based on a patch by Eric Wong. [Bug #5306] [ruby-core:39435]

- thread.c (rb_fd_rcopy): New. for reverse fd copy.

- test/-ext-/old_thread_select/test_old_thread_select.rb
  (test_old_select_false_positive): test for bug5306.

- ext/-test-/old_thread_select/old_thread_select.c (fdset2array):
  New. convert fdsets to array.

- ext/-test-/old_thread_select/old_thread_select.c (old_thread_select):
  return 'read', 'write', 'except' argument of rb_thread_select()
  to ruby script.

Revision 33256 - 09/12/2011 11:36 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_thread_select): fix to ignore an argument modification of rb_thread_fd_select(). based on a patch by Eric Wong. [Bug #5306] [ruby-core:39435]

- thread.c (rb_fd_rcopy): New. for reverse fd copy.

- test/-ext-/old_thread_select/test_old_thread_select.rb
  (test_old_select_false_positive): test for bug5306.

- ext/-test-/old_thread_select/old_thread_select.c (fdset2array):
  New. convert fdsets to array.

- ext/-test-/old_thread_select/old_thread_select.c (old_thread_select):
  return 'read', 'write', 'except' argument of rb_thread_select()
  to ruby script.

Revision 33256 - 09/12/2011 11:36 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_thread_select): fix to ignore an argument modification of rb_thread_fd_select(). based on a patch by Eric Wong. [Bug #5306] [ruby-core:39435]

- thread.c (rb_fd_rcopy): New. for reverse fd copy.

- test/-ext-/old_thread_select/test_old_thread_select.rb
  (test_old_select_false_positive): test for bug5306.

- ext/-test-/old_thread_select/old_thread_select.c (fdset2array):
  New. convert fdsets to array.

- ext/-test-/old_thread_select/old_thread_select.c (old_thread_select):
  return 'read', 'write', 'except' argument of rb_thread_select()
  to ruby script.

Revision 167f6b29 - 09/12/2011 11:41 AM - kosaki (Motohiro KOSAKI)

merge revision(s) 33256:

```
    * thread.c (rb_thread_select): fix to ignore an argument
      modification of rb_thread_fd_select().
      based on a patch by Eric Wong. [Bug #5306] [ruby-core:39435]
```

```
    * thread.c (rb_fd_rcopy): New. for reverse fd copy.
```

```
    * test/-ext-/old_thread_select/test_old_thread_select.rb
    (test_old_select_false_positive): test for bug5306.
    * ext/-test-/old_thread_select/old_thread_select.c (fdset2array):
      New. convert fdsets to array.

    * ext/-test-/old_thread_select/old_thread_select.c (old_thread_select):
      return 'read', 'write', 'except' argument of rb_thread_select()
      to ruby script.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@33257 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision d24e1dac - 09/14/2011 02:44 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_fd_rcopy): added an argument guard. Patch by NAKAMURA Usaku. [Bug #5306] [ruby-core:39435]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@33266 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 33266 - 09/14/2011 02:44 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_fd_rcopy): added an argument guard. Patch by NAKAMURA Usaku. [Bug #5306] [ruby-core:39435]

Revision 33266 - 09/14/2011 02:44 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_fd_rcopy): added an argument guard. Patch by NAKAMURA Usaku. [Bug #5306] [ruby-core:39435]

Revision 33266 - 09/14/2011 02:44 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_fd_rcopy): added an argument guard. Patch by NAKAMURA Usaku. [Bug #5306] [ruby-core:39435]

Revision 33266 - 09/14/2011 02:44 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_fd_rcopy): added an argument guard. Patch by NAKAMURA Usaku. [Bug #5306] [ruby-core:39435]

Revision 33266 - 09/14/2011 02:44 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_fd_rcopy): added an argument guard. Patch by NAKAMURA Usaku. [Bug #5306] [ruby-core:39435]

Revision 33266 - 09/14/2011 02:44 AM - kosaki (Motohiro KOSAKI)

- thread.c (rb_fd_rcopy): added an argument guard. Patch by NAKAMURA Usaku. [Bug #5306] [ruby-core:39435]

Revision fe3306bf - 09/14/2011 02:46 AM - kosaki (Motohiro KOSAKI)

merge revision(s) 33266:

```
    * thread.c (rb_fd_rcopy): added an argument guard.
      Patch by NAKAMURA Usaku. [Bug #5306] [ruby-core:39435]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@33268 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 09/10/2011 09:34 AM - kosaki (Motohiro KOSAKI)

I think it's duplicated with 5299.

#2 - 09/10/2011 11:28 AM - cfis (Charlie Savage)

Unfortunately it is not. That was the first problem - and resulted in segmentation faults. We manually backported the fix for #5299 to our local copy of ruby 193. Once we did that, it fixed the segmentation faults, but resulted in this problem.

So this is a new problem with that particular commit.

#3 - 09/10/2011 12:53 PM - normalperson (Eric Wong)

Charlie Savage cfis@savagexi.com wrote:

> It causes eventmachine to hang on CentOS 5.5. Not sure what the issue
> is, but its easily reproduced by by running the test
> eventmachine/tests/test_epoll.rb.

I have CentOS 5.4, x86_64, kernel 2.6.18-164.11.1.el5

```
rake compile
ruby -I .:lib:tests/ tests/test_epoll.rb
```

Works for me on an unpacked eventmachine-1.0.0.beta.3 tree with
ruby_1_9_3 branch. However, only 2 tests appeared enabled.

> We noticed this because it also causes the tweetstream gem to hang.
>
> The same setup works on Fedora 14 and an up-to-date arch linux.
> Specific version information included below.

Yes, unable to reproduce on a more modern Debian testing machine
(x86_64)

> Linux app1.zerista.com 2.6.18-238.19.1.el5.centos.plus #1 SMP Mon Jul
> 18 10:05:09 EDT 2011 x86_64 x86_64 x86_64 GNU/Linux

I'll try to find a machine closer to the above.

#4 - 09/10/2011 12:59 PM - cfis (Charlie Savage)

Hi Eric,

> > It causes eventmachine to hang on CentOS 5.5.

Sorry, these machines are actually CentOS 5.6. The latest patches were applied via yum update about a week ago, so its pretty up-to-date.

> I have CentOS 5.4, x86_64, kernel 2.6.18-164.11.1.el5
>
> rake compile
> ruby -I .:lib:tests/ tests/test_epoll.rb
>
> Works for me on an unpacked eventmachine-1.0.0.beta.3 tree with
> ruby_1_9_3 branch. However, only 2 tests appeared enabled.

So what we see is this test hanging:

```
def test_datagrams
$in = $out = ""
EM.run {
EM.open_datagram_socket "127.0.0.1", @port, TestDatagramServer
EM.open_datagram_socket "127.0.0.1", 0, TestDatagramClient, @port
}
assert_equal( "1234567890", $in )
assert_equal( "abcdefghij", $out )
end
```

It hangs on the first EM.open_datagram_socket call.

Here is another one, this time from test_pure_ruby.rb (which in fact seems misnamed, it is using the C code):

```
def test_connrefused
assert_nothing_raised do
EM.run {
setup_timeout(2)
EM.connect "127.0.0.1", @port, TestConnrefused
}
end
```

In this one, its the EM connect call that hangs.

> I'll try to find a machine closer to the above.

Probably a yum update will get you there...

Let me know if there is anything we can do to help debug this. Its happens across 8 servers (all of which are at the same CentOS release, albeit they did start as the same VM image a while back).

Charlie

#5 - 09/10/2011 03:29 PM - normalperson (Eric Wong)

Charlie Savage cfis@savagexi.com wrote:

> Sorry, these machines are actuall CentOS 5.6. The latest patches were
> applied via yum update about a week ago, so its pretty up-to-date.

OK, I'm closer with 2.6.18-238.9.1.el5xen but still can't reproduce it.

I don't have permission to upgrade kernels on CentOS images,
unfortunately. It's the weekend so the folks that do have permission
aren't around...

> So what we see is this test hanging:
>
> ```
> def test_datagrams
> $in = $out = ""
> EM.run {
> EM.open_datagram_socket "127.0.0.1", @port, TestDatagramServer
> EM.open_datagram_socket "127.0.0.1", 0, TestDatagramClient, @port
> }
> assert_equal( "1234567890", $in )
> assert_equal( "abcdefghij", $out )
> end
> ```
>
> It hangs on the first EM.open_datagram_socket call.

Can you show us "strace -f -v" output from that test?

Maybe sprinkle some `fprintf(stderr, "%s:%d\n", **FILE**, **LINE**);'
or similar inside EventMachine_t::OpenDatagramSocket and see where it
gets to? It shouldn't hit gethostbyname()...

> Here is another one, this time from test_pure_ruby.rb (which in fact seems misnamed, it is using the C code):
>
> ```
> def test_connrefused
> assert_nothing_raised do
> EM.run {
> setup_timeout(2)
> EM.connect "127.0.0.1", @port, TestConnrefused
> }
> end
> ```
>
> In this one, its the EM connect call that hangs.

I can't reproduce this, either...

Also, can you extract these tests and run with a hand-picked port?

> Let me know if there is anything we can do to help debug this. Its
> happens across 8 servers (all of which are at the same CentOS release,
> albeit they did start as the same VM image a while back).

I assume you tried a clean build/install of Ruby to make sure all
objects got rebuilt and reinstalled?

Can you also try running `pmap $PID' on the hung processes to make sure
it's loading the correct libs + versions?

#6 - 09/10/2011 05:00 PM - cfis (Charlie Savage)

- File strace_hangs.log added

- File strace_completes.log added

- File strace_pure.log added

- File pmap.log added

Ok, on the first test, strange results. Running this command:

```
strace -f -v ruby -I.:lib:tests tests/test_epoll.rb -n test_datagrams
```

Hangs the test as expected. But running this command:

```
strace -f -v ruby -I.:lib:tests tests/test_epoll.rb -n test_datagrams &> /tmp/strace1.log
```

Causes the test runs to completion. And then annoyingly enough that one particular test works after that. If I reboot the machine, then the test hangs again.

I have attached 2 logs, strace_completes.log and strace_hangs.log. stace_hangs.log is only the last few hundred lines (rest scrolled off the top), but what I saw matches strace_completes.log to line 2,271. After that, the two diverge.

The story is different for the second test, it always hangs:

```
strace -v -v ruby -I.:lib:tests tests/test_pure.rb -n test_connrefused 2>&1 | tee /tmp/strace_pure.log
```

That log is attached.

As for your other questions:

> Also, can you extract these tests and run with a hand-picked port?

Sure. The connection refused one is intentionally picking the first unused port. It turns out to be 9001.

> I assume you tried a clean build/install of Ruby to make sure all > objects got rebuilt and reinstalled?

Yes.

```
$cd /usr/src/ruby
$git pull (on the ruby 193 branch)
$git clean -fx
$autoconf
$./configure --prefix=/usr --enable-shared=true
$make
$make install
```

> Can you also try running `pmap $PID' on the hung processes to make > sure it's loading the correct libs + versions?

```
$ps -ef | grep ruby
cfis     16185 15381  4 01:51 pts/1    00:00:00 ruby -I.:lib:tests

$pmap 16185
(see attached log)
```

Hope this info helps.

#7 - 09/10/2011 05:17 PM - cfis (Charlie Savage)

And a bit more info. Running the datagrams test under GDB.

```
$gdb --args ruby -I.:lib:tests tests/test_epoll.rb -n test_datagrams
(gdb) run

... hangs ...
hit ctrl+c

Program received signal SIGINT, Interrupt.
0x000000375200d91b in read () from /lib64/libpthread.so.0

(gdb) bt
#0  0x000000375200d91b in read () from /lib64/libpthread.so.0
#1  0x00002aaaae9ea3ce in EventMachine_t::_ReadLoopBreaker (this=0xd61b50)
at em.cpp:998
#2  0x00002aaaae9ebc9a in EventMachine_t::_RunSelectOnce (this=0xd61b50)
at em.cpp:935
#3  0x00002aaaae9ec4f5 in EventMachine_t::_RunOnce (this=0x9) at em.cpp:498
#4  0x00002aaaae9ee183 in EventMachine_t::Run (this=0xd61b50) at em.cpp:478
#5  0x00002aaaae9e86a9 in t_run_machine_without_threads (self=9)
at rubymain.cpp:219
#6  0x00002aaaaac1b2d0 in vm_call_cfunc (th=0x602520, cfp=0x2aaaae5c7778,
num=0, blockptr=0x1, flag=24, id=0, me=0x8e9f90, recv=9127040)
at vm_insnhelper.c:404
etc.

(gdb) frame 1
#1  0x00002aaaae9ea3ce in EventMachine_t::_ReadLoopBreaker (this=0xd61ce0)
at em.cpp:998
998             read (LoopBreakerReader, buffer, sizeof(buffer));
(gdb) list
993             /* The loop breaker has selected readable.
994              * Read it ONCE (it may block if we try to read it twice)
995              * and send a loop-break event back to user code.
996              */
997             char buffer [1024];
998             read (LoopBreakerReader, buffer, sizeof(buffer));
999             if (EventCallback)
```

Running the other test, gdb --args ruby -I.:lib:tests tests/test_pure.rb -n test_connrefused, shows the same backtrace in gdb.

#8 - 09/11/2011 09:30 AM - normalperson (Eric Wong)

- File 0001-thread.c-rb_thread_select-mark-original-fd_sets-prop.patch added

Thanks for the straces, I was able to tell the EM pipe was stuck on a
false-positive and calling a blocking read() on a pipe that had no data.

Attached is a patch which should fix the issue, sorry for the bug :x

#9 - 09/11/2011 11:59 AM - kosaki (Motohiro KOSAKI)

- ruby -v changed from ruby 1.9.3dev (2011-09-09 revision 33236) [x86_64-linux] to -

2011/9/11 Eric Wong normalperson@yhbt.net:

> Issue #5306 has been updated by Eric Wong.
>
> File 0001-thread.c-rb_thread_select-mark-original-fd_sets-prop.patch added
>
> Thanks for the straces, I was able to tell the EM pipe was stuck on a
> false-positive and calling a blocking read() on a pipe that had no data.
>
> Attached is a patch which should fix the issue, sorry for the bug :x

Your patch will break non linux platform. I can't apply it. :x

#10 - 09/11/2011 02:56 PM - cfis (Charlie Savage)

Ok, some questions so I can understand this code:

How is the false-positive happening?

Why does this break on non-linux platforms?

And then obviously, what is the next step?

Thanks for looking into this and the quick responses.

#11 - 09/11/2011 03:12 PM - normalperson (Eric Wong)

- File 0001-thread.c-rb_thread_select-mark-original-fd_sets-prop.patch added

Hopefully a better patch is attached. I have no way of testing on non-Linux,
but I did test successfully without HAVE_RB_FD_INIT defined. _WIN32 tester (and
potential fixer) is needed.

#12 - 09/11/2011 03:23 PM - normalperson (Eric Wong)

Charlie Savage cfis@savagexi.com wrote:

> Ok, some questions so I can understand this code:
>
> How is the false-positive happening?

rb_thread_select() needs to modify the arguments passed to it (and clear out not-ready descriptors). My patch fixed that for Linux and other platforms with NFDBITS && HAVE_RB_FD_INIT.

> Why does this break on non-linux platforms

I missed the (NFDBITS && HAVE_RB_FD_INIT) code paths completely.

> And then obviously, what is the next step?

I am testing a patch, I manually disabled the HAVE_RB_FD_INIT code paths to test, but I cannot test _WIN32 path.

#13 - 09/11/2011 03:40 PM - cfis (Charlie Savage)

Thanks for the explanations.

I can test on windows - I have mswin and mingw builds. How to test though? Are there any tests in the test suite I should run to verify? Would love to run the whole test suite, but sadly that doesn't work on windows.

#14 - 09/11/2011 03:53 PM - normalperson (Eric Wong)

Charlie Savage cfis@savagexi.com wrote:

> I can test on windows - I have mswin and mingw builds. How to test though? Are there any tests in the test suite I should run to verify? Would love to run the whole test suite, but sadly that doesn't work on windows.

./ruby -I .ext/$PLATFORM test/-ext-/old_thread_select/test_old_thread_select.rb

For me, I have PLATFORM=x86_64-linux

#15 - 09/11/2011 03:56 PM - cfis (Charlie Savage)

Hmm, I take is this is against head? On the 1.9.3 branch there is already this method (line 2384):

void
rb_fd_copy(rb_fdset_t *dst, const fd_set *src, int max)

The patch then adds this right below it (line 2399):

static void
rb_fd_rcopy(fd_set *dst, rb_fdset_t *src)

And then lower down (line 2690):

if (read) {
rfds = &fdsets[0];
rb_fd_init(rfds);
rb_fd_copy(rfds, read, max);
}

So that rb_fd_copy call would no longer work.

#16 - 09/11/2011 04:23 PM - normalperson (Eric Wong)

Charlie Savage cfis@savagexi.com wrote:

> Issue #5306 has been updated by Charlie Savage.
>
> Hmm, I take is this is against head? On the 1.9.3 branch there is already this method (line 2384):

It should apply cleanly to r33236 (ruby_1_9_3)

> void
> rb_fd_copy(rb_fdset_t *dst, const fd_set *src, int max)
>
> The patch then adds this right below it (line 2399):

```
static void
rb_fd_rcopy(fd_set *dst, rb_fdset_t *src)
```

The new function is "rcopy" (reverse copy). I named it based on memrchr()
vs memchr(). Maybe someone can think of a better name?

#17 - 09/11/2011 05:15 PM - cfis (Charlie Savage)

Ah, totally missed that r - its not obvious if you aren't looking for it.

Patch doesn't compile on Windows:

```
thread.c
./../thread.c(2466) : error C2143: syntax error : missing ')' before ';'
NMAKE : fatal error U1077: '"c:\Program Files (x86)\Microsoft Visual Studio 10.0\VC\BIN\cl.EXE"' : return code '0x2'
Stop.
```

As the error says, easy to fix, its missing a) at the end of the line.

The test passes on mswin. One fails on mingw:

```
1) Failure:
test_old_select_false_positive(TestOldThreadSelect) [ruby/test/-ext-/old_thread_select/test_old_thread_select.rb:34]:
[ruby-core:39435].
<[5]> expected but was
<[3, 5]>.
4 tests, 12 assertions, 1 failures, 0 errors, 0 skips
```

#18 - 09/12/2011 02:27 AM - kosaki (Motohiro KOSAKI)

- File old_thread_select.patch added

```
static void
rb_fd_rcopy(fd_set *dst, rb_fdset_t *src)
{
size_t size = howmany(rb_fd_max(src), NFDBITS) * sizeof(fd_mask);
if (size < sizeof(fd_set)) size = sizeof(fd_set);
memcpy(dst, rb_fd_ptr(src), size);
}
```

If size > sizeof(fd_set), this code makes memory corruption.

```
static void
rb_fd_rcopy(fd_set *dst, rb_fdset_t *src)
{
memcpy(dst->fd_array, src->fdset->fd_array,
dst->fd_count * sizeof(dst->fd_array[0]);
dst->fd_count = src->fdset->fd_count;
}
```

Bad indentation of coding style violation.
Also, if src->fdset->fd_count > FD_SETSIZE, we should return an error or raise an exception.

Attached new patch. It works both linux and windows. Can you please review it?

#19 - 09/12/2011 08:23 AM - normalperson (Eric Wong)

Motohiro KOSAKI kosaki.motohiro@gmail.com wrote:

> Attached new patch. It works both linux and windows. Can you please
> review it?

Thanks! I can confirm it's good on Linux, Charlie?

#20 - 09/12/2011 08:36 PM - kosaki (Motohiro KOSAKI)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r33256.
Charlie, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- thread.c (rb_thread_select): fix to ignore an argument modification of rb_thread_fd_select(). based on a patch by Eric Wong. [Bug #5306] [ruby-core:39435]

- thread.c (rb_fd_rcopy): New. for reverse fd copy.

- test/-ext-/old_thread_select/test_old_thread_select.rb (test_old_select_false_positive): test for bug5306.

- ext/-test-/old_thread_select/old_thread_select.c (fdset2array): New. convert fdsets to array.

- ext/-test-/old_thread_select/old_thread_select.c (old_thread_select): return 'read', 'write', 'except' argument of rb_thread_select() to ruby script.

#21 - 09/12/2011 08:42 PM - kosaki (Motohiro KOSAKI)

committed both trunk and ruby_1_9_3.

#22 - 09/13/2011 01:29 AM - cfis (Charlie Savage)

- File mingw_backtrace.txt added

Thanks for all the effort. But sorry, not fixed yet. This version segfaults on MinGW. Trace attached. Will check mswin next.

#23 - 09/13/2011 03:53 AM - usa (Usaku NAKAMURA)

Hello,

In message "[ruby-core:39483] [Ruby 1.9 - Bug #5306] Application Hangs Due to Recent rb_thread_select Changes"
on Sep.13,2011 01:29:28, cfis@savagexi.com wrote:

> File mingw_backtrace.txt added
>
> Thanks for all the effort. But sorry, not fixed yet. This version segfaults on MinGW. Trace attached. Will check mswin next.

Hmm, did you do make install before running the test?
C level backtrace information of your trace says that the ruby
core dll is c:\MinGW\local\ruby\bin\msvcrt-ruby191.dll .
I guess that your build path is c:/MinGW/local/src/ruby, and
the built ruby core dll is c:/MinGW/local/src/ruby/msvcrt-ruby191.dll .

If you want to test safely, run as follows:
make test-all TESTS="-- -ext-/old_thread_select"

BTW, I've checked kosaki-san's patch with x64-mswin64.
No problem was reported in test.

P.S.
kosaki-san, I want to add a guard to your patch.

--- thread.c.bak 2011-09-13 03:40:05.948172400 +0900
+++ thread.c 2011-09-13 03:40:24.308222500 +0900
@@ -2469,7 +2469,9 @@ rb_fd_rcopy(fd_set *dst, rb_fdset_t *src
{
int max = rb_fd_max(src);

- if (max > FD_SETSIZE) {
- /* we assume src is the result of select() with dst, so dst should be
- * larger or equal than src. */
- if (max > FD_SETSIZE || max > dst->fd_count) { rb_raise(rb_eArgError, "too large fdsets"); }

Regards,
--
U.Nakamura usa@garbagecollect.jp

#24 - 09/13/2011 05:38 AM - cfis (Charlie Savage)

Ok, I rebuilt everything from scratch and did not encounter any errors - sorry for the false alarm. mswin also checked out fine.

We will next test this fix on the original servers where we encountered the problem. If any issues remain, I will reopen the ticket.

Thanks again for the help.

#25 - 09/13/2011 05:30 PM - normalperson (Eric Wong)

Motohiro KOSAKI kosaki.motohiro@gmail.com wrote:

> File old_thread_select.patch added
>
>> static void
>> rb_fd_rcopy(fd_set *dst, rb_fdset_t *src)
>> {
>> size_t size = howmany(rb_fd_max(src), NFDBITS) * sizeof(fd_mask);
>> if (size < sizeof(fd_set)) size = sizeof(fd_set);
>> memcpy(dst, rb_fd_ptr(src), size);
>> }
>
> If size > sizeof(fd_set), this code makes memory corruption.

I just thought of this again and think rb_bug() is better than
rb_raise() here. While unlikely to hit either case, rb_raise()
will leak memory since the rb_fd_term() call gets skipped.

#26 - 09/14/2011 12:23 PM - kosaki (Motohiro KOSAKI)

> BTW, I've checked kosaki-san's patch with x64-mswin64.
> No problem was reported in test.
>
> P.S.
> kosaki-san, I want to add a guard to your patch.
>
> --- thread.c.bak Â Â Â Â 2011-09-13 03:40:05.948172400 +0900
> +++ thread.c Â Â 2011-09-13 03:40:24.308222500 +0900
> @@ -2469,7 +2469,9 @@ rb_fd_rcopy(fd_set *dst, rb_fdset_t *src
> Â {
> Â Â int max

#27 - 09/14/2011 12:23 PM - kosaki (Motohiro KOSAKI)

2011/9/13 Eric Wong normalperson@yhbt.net:

> Motohiro KOSAKI kosaki.motohiro@gmail.com wrote:
>
>> File old_thread_select.patch added
>>
>>> static void
>>> rb_fd_rcopy(fd_set *dst, rb_fdset_t *src)
>>> {
>>> Â Â size_t size

Files

strace_hangs.log	3.87 KB	09/10/2011	cfis (Charlie Savage)
strace_completes.log	307 KB	09/10/2011	cfis (Charlie Savage)
strace_pure.log	286 KB	09/10/2011	cfis (Charlie Savage)
pmap.log	7.19 KB	09/10/2011	cfis (Charlie Savage)
0001-thread.c-rb_thread_select-mark-original-fd_sets-prop.patch	3.43 KB	09/11/2011	normalperson (Eric Wong)
0001-thread.c-rb_thread_select-mark-original-fd_sets-prop.patch	4.08 KB	09/11/2011	normalperson (Eric Wong)

old_thread_select.patch	3.09 KB	09/12/2011	kosaki (Motohiro KOSAKI)
mingw_backtrace.txt	7.09 KB	09/13/2011	cfis (Charlie Savage)