

Ruby trunk - Bug #3005

Ruby core dump - [BUG] rb_sys_fail() - errno == 0

03/24/2010 09:24 PM - syepes (Sebastian YEPES)

Status: Closed	
Priority: Normal	
Assignee:	
Target version: 2.0.0	
ruby -v: 1.9.1p376	Backport:
Description	
<pre>=begin My ruby script is generating cores, this is the dump: -- sv.rb:268: [BUG] rb_sys_fail() - errno == 0 ruby 1.9.1p376 (2009-12-07 revision 26041) [ia64-hpux11.31] -- control frame ----- c:0005 p:---- s:0016 b:0016 l:000015 d:000015 CFUNC :gets c:0004 p:0210 s:0013 b:0013 l:000012 d:000012 METHOD sv.rb:268 c:0003 p:0653 s:0006 b:0006 l:0009b4 d:001a54 EVAL sv.rb:780 c:0002 p:---- s:0004 b:0004 l:000003 d:000003 FINISH c:0001 p:0000 s:0002 b:0002 l:0009b4 d:0009b4 TOP -- Ruby level backtrace information----- sv.rb:268:in 'gets' sv.rb:268:in \$vcml_listener' sv.rb:780:in `` [NOTE] You may encounter a bug of Ruby interpreter. Bug reports are welcome. For details: http://www.ruby-lang.org/bugreport.html</pre>	
<pre>(gdb) bt #0 0x60000000c01e93f0:0 in _lwp_kill+0x30 () from /usr/lib/hpux32/libpthread.so.1 #1 0x60000000c0153310:0 in pthread_kill+0x8d0 () from /usr/lib/hpux32/libpthread.so.1 #2 0x60000000c02f05a0:0 in raise+0xe0 () from /usr/lib/hpux32/libc.so.1 #3 0x60000000c0410cd0:0 in abort+0x170 () from /usr/lib/hpux32/libc.so.1 #4 0x60000000d663ef00:0 in rb_bug (fmt=) at error.c:233 #5 0x60000000d66434e0:0 in rb_sys_fail (mesg=) at error.c:1140 #6 0x60000000d6696cf0:0 in io_fillbuf (fptr=) at io.c:1169 #7 0x60000000d669b5d0:0 in rb_io_getline_fast (fptr=, enc=) at io.c:2151 #8 0x60000000d669c360:0 in rb_io_getline_1 (rs=, limit=-1, io=) at io.c:2229 #9 0x60000000d669c550:2 in inline rb_io_getline () at io.c:2312 #10 0x60000000d669c530:0 in rb_io_gets_m (argc=, argv=, io=) at io.c:2347 #11 0x60000000d686f560:0 in call_cfunc (func=, recv=, len=, argc=, argv=) at vm_insnhelper.c:286 #12 0x60000000d6870270:0 in vm_call_cfunc (th=, reg_cfp=, num=, id=, oid=, recv=, klass=, mn=, blockptr=) at vm_insnhelper.c:384 ---Type to continue, or q to quit--- #13 0x60000000d6872190:0 in vm_call_method (th=, cfp=, num=, blockptr=,</pre>	

```
flag=, id=, mn=,
recv=) at vm_inshelper.c:517
#14 0x60000000d6879dd0:0 in vm_exec_core (th=) at insns.def:995
#15 0x60000000d68881c0:0 in vm_exec () at vm.c:1080
#16 0x60000000d68957d0:0 in rb_iseq_eval_main (iseqval=)
at vm.c:1304
#17 0x60000000d6647c80:0 in ruby_exec_node (n=) at eval.c:205
#18 0x60000000d6647e00:0 in ruby_run_node (n=0x402f8400) at eval.c:226
#19 0x4000b00:0 in main (argc=2, argv=0x7ffedb4) at main.c:35
```

```
=end
```

History

#1 - 03/25/2010 09:21 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Feedback

```
=begin
How to reproduce it?
=end
```

#2 - 03/25/2010 06:46 PM - syepes (Sebastian YEPES)

- File sv.rb added

```
=begin
Ok, I have made a reduced version of the script that always reproduces the core dumps.
The problem must be related with the oci8, because if I comment out the lines 23 and 25 it does not generate cores.
```

Hope some one can help me solve this issue.

Environment:
HP-UX B.11.31 U ia64
ruby-oci8-2.0.4
ruby 1.9.1p376 (2009-12-07 revision 26041) [ia64-hpux11.31]
Oracle 11gR2

Reproduction:
ruby sv.rb
Send a HUP to the ruby process (kill -HUP
-> Here comes the core ;-)

```
# ruby sv.rb
```

```
cfgfile_load: LOADING CONFIGURATION DATA
xml_listener: Processing XML
xml_register_hup: RELOADING CONFIGURATION DATA
cfgfile_load: LOADING CONFIGURATION DATA
sv.rb:44: [BUG] rb_sys_fail() - errno == 0
ruby 1.9.1p376 (2009-12-07 revision 26041) [ia64-hpux11.31]
```

```
-- control frame -----
c:0005 p:---- s:0014 b:0014 l:000013 d:000013 CFUNC :gets
c:0004 p:0098 s:0011 b:0011 l:000010 d:000010 METHOD sv.rb:44
c:0003 p:0131 s:0006 b:0006 l:0009b4 d:002524 EVAL sv.rb:60
c:0002 p:---- s:0004 b:0004 l:000003 d:000003 FINISH
c:0001 p:0000 s:0002 b:0002 l:0009b4 d:0009b4 TOP
```

```
-- Ruby level backtrace information-----
sv.rb:44:in `gets'
sv.rb:44:in `xml_listener'
sv.rb:60:in ``
```

[NOTE]
You may encounter a bug of Ruby interpreter. Bug reports are welcome.
For details: <http://www.ruby-lang.org/bugreport.html>

Abort(coredump)

=end

#3 - 04/06/2010 05:41 PM - syepes (Sebastian YEPES)

=begin

I have just tested this problem with the last version of Ruby 1.9.1p378 and it also generates core dumps.

Anyone know what can be causing this issue?

```
ruby t.rb
cfgfile_load: LOADING CONFIGURATION DATA
xml_listener: Processing XML
xml_register_hup: RELOADING CONFIGURATION DATA
cfgfile_load: LOADING CONFIGURATION DATA
t.rb:44: [BUG] rb_sys_fail() - errno == 0
ruby 1.9.1p378 (2010-01-10 revision 26273) [ia64-hpux11.31]

-- control frame -----
c:0005 p:---- s:0014 b:0014 l:000013 d:000013 CFUNC :gets
c:0004 p:0098 s:0011 b:0011 l:000010 d:000010 METHOD t.rb:44
c:0003 p:0131 s:0006 b:0006 l:0009b4 d:0023c4 EVAL t.rb:60
c:0002 p:---- s:0004 b:0004 l:000003 d:000003 FINISH
c:0001 p:0000 s:0002 b:0002 l:0009b4 d:0009b4 TOP
```

```
-- Ruby level backtrace information-----
t.rb:44:in gets'
t.rb:44:inxml_listener'
t.rb:60:in ``
```

[NOTE]
You may encounter a bug of Ruby interpreter. Bug reports are welcome.
For details: <http://www.ruby-lang.org/bugreport.html>

Abort

=end

#4 - 04/07/2010 11:02 PM - kubo (Takehiro Kubo)

=begin

If it is caused by the signal handler setting changed by Oracle client library, it may be fixed by adding BEQUEATH_DETACH=yes to sqlnet.ora.

http://download.oracle.com/docs/cd/B19306_01/network.102/b14213/sqlnet.htm#sthref374

=end

#5 - 04/07/2010 11:31 PM - syepes (Sebastian YEPES)

=begin

Thanks for the info, I have just tested adding this setting to: /opt/oracle/client/product/11.1.0/network/admin/sqlnet.ora but I am still getting the cores ;-(

Do you know what we can do to debug this issue?

=end

#6 - 04/08/2010 10:30 PM - kubo (Takehiro Kubo)

=begin

If I do, I'll install tusc(*1) and run the following command to trace system calls.

```
tusc -o tusc.log ruby sv.rb
```

*1 <ftp://ftp.cup.hp.com/dist/networking/tools/>

=end

#7 - 04/09/2010 12:51 AM - kubo (Takehiro Kubo)

=begin

The issue will be fixed the following patch.

=====

-- io.c~ 2009-08-09 00:34:48.000000000 +0900

+++ io.c 2010-04-09 00:37:35.000000000 +0900

@@ -496,6 +496,7 @@

```
struct io_internal_struct {
int fd;
```

- int saved_errno; void buf; size_t capa; }; @@ -504,36 +505,46 @@ internal_read_func(void *ptr) { struct io_internal_struct *iis = (struct io_internal_struct)ptr;
- return read(iis->fd, iis->buf, iis->capa);
- int ret = read(iis->fd, iis->buf, iis->capa);
- iis->saved_errno = errno;
- return ret; }

```
static VALUE
```

```
internal_write_func(void ptr)
```

```
{
struct io_internal_struct *iis = (struct io_internal_struct)ptr;
```

- return write(iis->fd, iis->buf, iis->capa);
- int ret = write(iis->fd, iis->buf, iis->capa);
- iis->saved_errno = errno;
- return ret; }

```
static int
```

```
rb_read_internal(int fd, void *buf, size_t count)
```

```
{
struct io_internal_struct iis;
```

- int ret;
- iis.fd = fd;
- iis.buf = buf;
- iis.capa = count;
- return rb_thread_blocking_region(internal_read_func, &iis, RUBY_UBF_IO, 0);
- ret = rb_thread_blocking_region(internal_read_func, &iis, RUBY_UBF_IO, 0);
- errno = iis.saved_errno;
- return ret;
- }

```
static int
```

```
rb_write_internal(int fd, void *buf, size_t count)
```

```
{
struct io_internal_struct iis;
```

- int ret;
- iis.fd = fd;
- iis.buf = buf;
- iis.capa = count;
- return rb_thread_blocking_region(internal_write_func, &iis, RUBY_UBF_IO, 0);
- ret = rb_thread_blocking_region(internal_write_func, &iis, RUBY_UBF_IO, 0);
- errno = iis.saved_errno;
- return ret;
- }

```
static long
```

=====

Ok, I have made a reduced version of the script that always reproduces the core dumps.

The problem must be related with the oci8, because if I comment out the lines 23 and 25 it does not generate cores.

The problem is caused by resetting errno in the signal handler. It can be reproduced by replacing line 23 and 25 with Math.log10(10), which set errno to zero.

=end

#8 - 04/09/2010 01:32 AM - nobu (Nobuyoshi Nakada)

=begin

Hi,

At Fri, 9 Apr 2010 00:51:49 +0900,
Takehiro Kubo wrote in [ruby-core:29340]:

Ok, I have made a reduced version of the script that always reproduces the core dumps. The problem must be related with the oci8, because if I comment out the lines 23 and 25 it does not generate cores.

The problem is caused by resetting errno in the signal handler. It can be reproduced by replacing line 23 and 25 with `Math.log10(10)`, which set errno to zero.

Thank you for your investigation and patch. It seems correct and I've merged it into the trunk.

--

Nobu Nakada

=end

#9 - 04/09/2010 01:37 AM - nobu (Nobuyoshi Nakada)

- *Status changed from Feedback to Closed*

- *% Done changed from 0 to 100*

=begin

This issue was solved with changeset [r27265](#).

Sebastian, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

=end

Files

sv.rb	1.36 KB	03/25/2010	syepes (Sebastian YEPES)
-------	---------	------------	--------------------------