Ruby master - Bug #6344

1.9.3 p125, p194 ruby causes SEGV with test_massign.rb on ppc/ppc64

04/23/2012 11:26 PM - mtasaka (Mamoru Tasaka)

Status:	Rejected		
Priority:	Normal		
Assignee:			
Target version:	1.9.3		
ruby -v:	ruby1.9.3p125	**Backport:**	

Description

1.9.3 p125, p194 ruby causes SEGV with test_massign.rb on ppc/ppc64.
Short reproducer and backtrace with ruby 1.9.3 p125:

[tasaka@localhost ruby-1.9.3-p125]$ cat rubydev-32581.rb
a,s=[],"aaa"
300.times { a<...
Reading symbols from
/home/tasaka/rpmbuild/BUILD/ruby-1.9.3-p125/miniruby...done.
(gdb) run -Ilib -I. --disable-gems ./rubydev-32581.rb
Starting program: /home/tasaka/rpmbuild/BUILD/ruby-1.9.3-p125/miniruby -Ilib
-I. --disable-gems ./rubydev-32581.rb
[Thread debugging using libthread_db enabled]
[New Thread 0xfffb1dff200 (LWP 1759)]

Program received signal SIGSEGV, Segmentation fault.
0x00000080db556b20 in .__makecontext () from /lib64/libc.so.6
Missing separate debuginfos, use: debuginfo-install
nss-softokn-freebl-3.12.9-3.el7.ppc64
(gdb) thread apply all bt

Thread 2 (Thread 0xfffb1dff200 (LWP 1759)):
#0 0x00000080db5fe054 in .**select () from /lib64/libc.so.6**
#1 0x000000001018911c in thread_timer (p=0x10280f68) at thread_pthread.c:1155
#2 0x00000080db70b330 in start_thread (arg=0xfffb1dff200) at
pthread_create.c:299
#3 0x00000080db6076ec in .clone () from /lib64/libc.so.6

Thread 1 (Thread 0x80db4d7010 (LWP 1756)):
#0 0x00000080db556b20 in .__makecontext () from /lib64/libc.so.6
#1 0x000000001018ff80 in fiber_initialize_machine_stack_context
(fib=0x103ab970, size=65536) at cont.c:606
#2 0x0000000010190094 in fiber_setcontext (newfib=0x103ab970,
oldfib=0x103ac940) at cont.c:623
#3 0x0000000010190214 in fiber_store (next_fib=0x103ab970) at cont.c:1234
#4 0x00000000101903f8 in fiber_switch (fibval=271105960, argc=, argv=0xfffb1e00098) at cont.c:1319
#5 rb_fiber_resume (fibval=271105960, argc=,
argv=0xfffb1e00098) at cont.c:1347
#6 0x00000000101905e4 in rb_fiber_m_resume (argc=,
argv=, fib=) at cont.c:1404
#7 0x000000001016c754 in call_cfunc (func=@0x10255a90: 0x101905c0
, recv=271105960, len=, argc=,
argv=) at vm_insnhelper.c:326
#8 0x0000000010171c74 in vm_call_cfunc (th=0x10281560, cfp=0xfffb1effe00,
num=, blockptr=, flag=0,
id=, me=0x1039b8f0, recv=271105960) at
vm_insnhelper.c:404
#9 vm_call_method (th=0x10281560, cfp=0xfffb1effe00, num=, blockptr=, flag=0, id=,
me=0x1039b8f0,
recv=271105960) at vm_insnhelper.c:534
#10 0x00000000101734f4 in vm_exec_core (th=0x10281560, initial=) at insns.def:1015
#11 0x0000000010178da8 in vm_exec (th=0x10281560) at vm.c:1220
#12 0x0000000010179480 in eval_string_with_cref (self=271477440, src=271373360,

scope=4, cref=0x0, file=0x101b7fd8 "(eval)", line=1) at vm_eval.c:1050
#13 0x0000000010179b20 in eval_string (argc=, argv=, self=271477440) at vm_eval.c:1091
#14 rb_f_eval (argc=, argv=,
self=271477440) at vm_eval.c:1139
#15 0x000000001016c754 in call_cfunc (func=@0x10254660: 0x101799a0 ,
recv=271477440, len=, argc=,
argv=) at vm_insnhelper.c:326
#16 0x0000000010171c74 in vm_call_cfunc (th=0x10281560, cfp=0xfffb1efff08,
num=, blockptr=, flag=8,
id=, me=0x1030f710, recv=271477440) at
vm_insnhelper.c:404
#17 vm_call_method (th=0x10281560, cfp=0xfffb1efff08, num=, blockptr=, flag=8, id=,
me=0x1030f710,
recv=271477440) at vm_insnhelper.c:534
#18 0x00000000101734f4 in vm_exec_core (th=0x10281560, initial=) at insns.def:1015
#19 0x0000000010178da8 in vm_exec (th=0x10281560) at vm.c:1220
#20 0x0000000010179078 in rb_iseq_eval_main (iseqval=271386440) at vm.c:1461
#21 0x00000000100559a8 in ruby_exec_internal (n=0x102d0748) at eval.c:204
#22 0x00000000100559f8 in ruby_exec_node (n=value has been optimized out
) at eval.c:251
#23 0x0000000010057650 in ruby_run_node (n=0x102d0748) at eval.c:244
#24 0x0000000010015664 in main (argc=5, argv=0xfffffff528) at main.c:38

1.9.3 p194 causes the same segv.

The attached patch seems to suppress this segv.

Associated revisions

Revision 5c567691 - 05/18/2012 08:32 AM - kosaki (Motohiro KOSAKI)

- cont.c: bump up fiber machine stack size when running on 64bit arch. [Bug #6344] [ruby-dev:45554]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35694 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 35694 - 05/18/2012 08:32 AM - kosaki (Motohiro KOSAKI)

- cont.c: bump up fiber machine stack size when running on 64bit arch. [Bug #6344] [ruby-dev:45554]

Revision 35694 - 05/18/2012 08:32 AM - kosaki (Motohiro KOSAKI)

- cont.c: bump up fiber machine stack size when running on 64bit arch. [Bug #6344] [ruby-dev:45554]

Revision 35694 - 05/18/2012 08:32 AM - kosaki (Motohiro KOSAKI)

- cont.c: bump up fiber machine stack size when running on 64bit arch. [Bug #6344] [ruby-dev:45554]

Revision 35694 - 05/18/2012 08:32 AM - kosaki (Motohiro KOSAKI)

- cont.c: bump up fiber machine stack size when running on 64bit arch. [Bug #6344] [ruby-dev:45554]

Revision 35694 - 05/18/2012 08:32 AM - kosaki (Motohiro KOSAKI)

- cont.c: bump up fiber machine stack size when running on 64bit arch. [Bug #6344] [ruby-dev:45554]

Revision 35694 - 05/18/2012 08:32 AM - kosaki (Motohiro KOSAKI)

- cont.c: bump up fiber machine stack size when running on 64bit arch. [Bug #6344] [ruby-dev:45554]

Revision 469cc485 - 05/18/2012 09:10 AM - kosaki (Motohiro KOSAKI)

decrease fiber stack size. 1MB is too large for windows. [Bug #6344]
[ruby-dev:45554]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35697 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 35697 - 05/18/2012 09:10 AM - kosaki (Motohiro KOSAKI)

decrease fiber stack size. 1MB is too large for windows. [Bug #6344]
[ruby-dev:45554]

Revision 35697 - 05/18/2012 09:10 AM - kosaki (Motohiro KOSAKI)

decrease fiber stack size. 1MB is too large for windows. [Bug #6344]
[ruby-dev:45554]

Revision 35697 - 05/18/2012 09:10 AM - kosaki (Motohiro KOSAKI)

decrease fiber stack size. 1MB is too large for windows. [Bug #6344]
[ruby-dev:45554]

Revision 35697 - 05/18/2012 09:10 AM - kosaki (Motohiro KOSAKI)

decrease fiber stack size. 1MB is too large for windows. [Bug #6344]
[ruby-dev:45554]

Revision 35697 - 05/18/2012 09:10 AM - kosaki (Motohiro KOSAKI)

decrease fiber stack size. 1MB is too large for windows. [Bug #6344]
[ruby-dev:45554]

Revision 35697 - 05/18/2012 09:10 AM - kosaki (Motohiro KOSAKI)

decrease fiber stack size. 1MB is too large for windows. [Bug #6344]
[ruby-dev:45554]

History

#1 - 04/23/2012 11:38 PM - mame (Yusuke Endoh)

- Status changed from Open to Feedback

[illegible]

[illegible] ppc [illegible]
[illegible]
[illegible]
[illegible] workaround [illegible]
[illegible]

[illegible]
[illegible]

--
Yusuke Endoh mame@tsg.ne.jp

#2 - 04/23/2012 11:52 PM - mtasaka (Mamoru Tasaka)

(Well, as RedHat people are also seeing this...)
So first of all, I think the current value "0x10000" is
almost baseless and theoretically this value should be
assigned dynamically. BTW Vit says that this segv happens
also on 2.0.0dev (2012-04-23 trunk 35432)
x86_64-linux and the above patch
seems to work.

Would you know how the current value is selected first of
all?

#3 - 04/24/2012 12:01 AM - mame (Yusuke Endoh)

mtasaka (Mamoru Tasaka) wrote:

> (Well, as RedHat people are also seeing this...)
> So first of all, I think the current value "0x10000" is
> almost baseless and theoretically this value should be
> assigned dynamically. BTW Vit says that this segv happens
> also on 2.0.0dev (2012-04-23 trunk 35432)
> x86_64-linux and the above patch
> seems to work.

What ticket (or mail) are you talking about?
Please give me a pointer.

--
Yusuke Endoh mame@tsg.ne.jp

#4 - 04/24/2012 12:03 AM - mtasaka (Mamoru Tasaka)

Downstream bug:
https://bugzilla.redhat.com/show_bug.cgi?id=803698

#5 - 04/24/2012 12:19 AM - mame (Yusuke Endoh)

- Status changed from Feedback to Assigned

- Assignee set to ko1 (Koichi Sasada)

TL;DR :-)

Assigning this to ko1, as I heard ko1 is planning to change the code so that the value is dynamically determined.
But I don't know ko1 can work soon. It might be helpful to elaborate the mechanism of the issue.

--
Yusuke Endoh mame@tsg.ne.jp

#6 - 04/24/2012 03:19 AM - kosaki (Motohiro KOSAKI)

If increasing machine stack, it definitly reduce limit of number of fibers.
But, of course, RISC need more stack than CISC and 64bit need more stack than 32bit. then I'm not surprised this patch solved ppc issue. So, I suspect the best way is,

```
#if 64BIT
#define FIBER_MACHINE_STACK_ALLOCATION_SIZE  (0x20000)
#else

#define FIBER_MACHINE_STACK_ALLOCATION_SIZE  (0x10000)
#endif
```

or likewise. I don't think dynamic fiber stack feature fit 1.9.3 branch.

#7 - 04/24/2012 03:21 AM - kosaki (Motohiro KOSAKI)

I meant, if a patch has negative impact against 32bit x86, I can't agree it. You should think 80%+ people are using 32bit x86.

#8 - 04/24/2012 12:30 PM - mame (Yusuke Endoh)

- Assignee changed from ko1 (Koichi Sasada) to kosaki (Motohiro KOSAKI)

Hello,

Okay, Kosaki-san, I leave this up to you.
But personally I'm against changing the parameter blindly, without
credible explanation and evidence about the mechanism of the problem.

2012/4/24, kosaki (Motohiro KOSAKI) kosaki.motohiro@gmail.com:

> If increasing machine stack, it definitly reduce limit of number of fibers.
> But, of course, RISC need more stack than CISC and 64bit need more stack
> than 32bit. then I'm not surprised this patch solved
> ppc issue.

OP said the same issue occurred on x86_64-linux, but I couldn't reproduce.
Can you?

--
Yusuke Endoh mame@tsg.ne.jp

#9 - 04/25/2012 12:21 AM - kosaki (Motohiro KOSAKI)

> OP said the same issue occurred on x86_64-linux, but I couldn't reproduce.
> Can you?

No I can't. we need more feedback, I think.

#10 - 05/18/2012 05:32 PM - kosaki (Motohiro KOSAKI)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r35694.
Mamoru, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- cont.c: bump up fiber machine stack size when running on 64bit arch. [Bug #6344] [ruby-dev:45554]

#11 - 05/19/2012 05:27 AM - kosaki (Motohiro KOSAKI)

- Status changed from Closed to Feedback

If anyone send me a feedback of ppc test result, I'll backport this.

#12 - 06/25/2012 05:23 PM - vo.x (Vit Ondruch)

Weird, it still/again crashes with ruby 2.0.0dev (2012-06-25 trunk 36213) [powerpc-linux]

http://ppc.koji.fedoraproject.org/koji/getfile?taskID=597002&name=build.log

#13 - 09/14/2012 05:02 AM - kosaki (Motohiro KOSAKI)

- Assignee deleted (kosaki (Motohiro KOSAKI))

#14 - 11/05/2012 07:57 PM - mame (Yusuke Endoh)

- Status changed from Feedback to Rejected

After all, increasing the value is not an essential fix, I think.

Again, there is no powerpc maintainer. I'm closing this ticket.
Feel free to reopen or open a new ticket if you can make a patch.

--
Yusuke Endoh mame@tsg.ne.jp

Files

ruby-1.9.3-p125-increase-stack-allocation-size-ppc.patch	418 Bytes	04/23/2012	mtasaka (Mamoru Tasaka)