Ruby - Bug #4047

Embedded Ruby issues

11/12/2010 02:14 PM - garthy (Garthy X)

Status:	Third Party's Issue		
Priority:	Normal		
Assignee:			
Target version:	2.0.0		
ruby -v:	ruby 1.9.3dev (2010-11-10 trunk 29738) [i686-linux]	**Backport:**	

Description

=begin
These bug(s) have been around since at least early 2007. I've updated the test code I'd developed to demonstrate them to work with the latest Ruby code. I'm passing it on in case it is useful to someone. To try it for yourself, edit the Makefile to contain the right paths, run make, then "./foo".

The included program ("foo") simulates making Ruby calls at various stack depths. If you build this against the nightly and stable snapshots as of 20101112 (Australia), you'll get a very fast crash (see [1] for output). Some notes:

- You can disable the use of Ruby with the "-b" flag. This causes the program to begin working, and can be used to confirm the rest of the program is fine, it's just the Ruby calls causing problems.
- You can disable the call to the Ruby garbage collector with the "-c" flag. This causes the program to last longer, but if I recall correctly, it'll crash eventually.
- You can disable the stack depth changing with the "-s" flag. This *used* to stop the problem from occurring (ie. showing the bug manifested only in changing stack depths), but presently doesn't stop it from crashing.

If the cause of the fast crash is found, try rerunning "foo" again, and leaving it running for a while. With Ruby calls enabled (no flags), it used to crash eventually. With the calls disabled, it used to pretty-much run indefinitely. The problem used to only occur when you used "-pg". Based on the new behaviour I am guessing there are multiple bugs.

The code included simulates use in a real program, which may call the Ruby code from different stack depths. When trying to embed Ruby code in a reliable program, it became crash-prone immediately, and behaved again when I disabled it. I tried making a start diagnosing the problem some years ago, but was hopelessly out of my depth.

I hope this is of use to someone. Good luck.

[1]:

```
Messy enabled. Ruby enabled. Verbose disabled. GC enabled.
Rand vals enabled. Rand stack size enabled.
Empty test...
Initialising...
Thrashing...
0:  12: [BUG] Segmentation fault
ruby 1.9.3dev (2010-11-10 trunk 29738) [i686-linux]

-- Control frame information -----------------------------------------------
c:0001 p:0000 s:0002 b:0002 l:000dc4 d:000dc4 TOP

-- C level backtrace information -------------------------------------------
./foo [0x816afcd]
./foo [0x8066229]
./foo [0x80662d8]
./foo [0x80fd978]
[0x8fd410]
./foo [0x815985e]
./foo [0x8162ff0]
./foo [0x804c707]
./foo [0x804c725]
```

```
./foo [0x804c9e5]
./foo [0x804c98f]
./foo [0x804c98f]
./foo [0x804c98f]
./foo [0x804c98f]
./foo [0x804c98f]
./foo [0x804c98f]
./foo [0x804c98f]
./foo [0x804c98f]
./foo [0x804cb63]
./foo [0x804d2a7]
/lib/tls/i686/cmov/libc.so.6(__libc_start_main+0xe6) [0xe8db56]
./foo [0x804c361]

-- Other runtime information -----------------------------------------------

SEGV received in SEGV handler
=end
```

History

#1 - 11/15/2010 06:45 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Third Party's Issue

=begin

```
static VALUE obj;
```

This object will be collected, you have to register the variable or the value to prevent from GC.

```
static void InitRuby4(void *t)
{
(void)t;
VALUE c_foo = rb_const_get(rb_cObject, rb_intern("Foo"));
VALUE c_bar = INT2NUM(555);
rb_gc_register_address(&obj); // this or
obj = rb_funcall(c_foo, rb_intern("new"), 1, c_bar);
rb_gc_register_mark_object(obj); // this.
}
```

=end

#2 - 11/15/2010 07:26 AM - garthy (Garthy X)

=begin
Thankyou! :) I have had no luck reporting this issue in the past. Thankyou for taking the time to looking at it, and especially for finding the fault in my test code.

I can confirm that this change solves the fast crash I was experiencing, and seems to work for longer runs as well (just a few minutes worth though).

I am going to set up this test under some different environments and for a few different Ruby versions, and see how they behave. I'll update again with results when they are ready.

=end

#3 - 11/15/2010 11:36 AM - garthy (Garthy X)

- File crashtest-20101115a.tgz added

=begin
Short version: Whatever the original cause of the crash, things seem to be working fine now.

Long version:

Okay, I've run through around 4000 iterations each with these configurations:

- CentOS 5.3 (base install, not updated) Ruby 1.8.3
- CentOS 5.3 (base install, not updated) Ruby 1.9.2-p0
- CentOS 5.3 (base install, not updated) Ruby 1.9.1-p378

- CentOS 5.3 (base install, not updated) Ruby Nightly snapshot 20101112
- Ubuntu 9.10 (updated fairly recently) Ruby 1.8.3
- Ubuntu 9.10 (updated fairly recently) Ruby Nightly snapshot 20101112

The original symptoms were a crash usually after about 100-200 iterations. 4000 iterations suggests the problem may be gone. The particular system it was run on seemed to affect the results (some seemed immune). It also only manifested with "-pg", as far as I could tell.

It is possible that the systems I tried it on are sufficiently different to the older ones that the bug no longer exists. Unfortunately, these systems are now long gone, so I cannot try them out.

It could also have been related to the compiler or compilation; I had to patch math.c in the 1.8.* series and cont.c in the 1.9* series to get them to build this time around (gcc 4.4.1 on Ubuntu 9.10, updated). I didn't have to do this previously.

It is also possible that I had a similar problem in my project at the time that was similar to the one in the test code, and it just happened to work most of the time on certain systems, whilst crashing frequently on others.

I'll run some longer overnight tests the next few days to be sure, but if there is indeed a bug in Ruby at this point, I can't reproduce it any more. Whatever the original cause, things appear to be working fine now.

Updated crash test code included as attachment.

Thankyou again Nobuyoshi Nakada for having a look at it and for finding and fixing the fault in my sample code.

=end

Files

crashtest-20101112.tgz	2.17 KB	11/12/2010	garthy (Garthy X)
crashtest-20101115a.tgz	2.49 KB	11/15/2010	garthy (Garthy X)