Ruby master - Bug #12939
MRI crashes when a hash table becomes huge
11/14/2016 11:03 PM - vmakarov (Vladimir Makarov)

Status: Closed
Priority: Normal
Assignee: 
Target version: 
ruby -v: 
Backport: 2.1: UNKNOWN, 2.2: UNKNOWN, 2.3: UNKNOWN

Description
Checking huge hash tables on the trunk, I found a bug in my code.
Here is the test case:

```
h = {}; 0x100000000.times{|i| h[i] = i}
```

MRI diagnostics for the test case is at the report end.

You need a machine with huge memory (at least 64 GB on my estimation) to reproduce the bug. I used a machine with 256 GB memory.

So there is no sense to add a test (make test) for the bug.

The probability for the bug occurrence is quite small. Still it would be nice to fix it. The patch in the attachment fixes the problem.

I also have some further memory and performance improvements for hash tables but I am going to submit it later for 2.5 release.

```
ruby 2.4.0dev (2016-11-09 trunk 56677) [x86_64-linux]
```

```
-e:1: [BUG] Segmentation fault at 0x0000000000000000
```

```
-- Control frame information -----------------------------------------------
c:0004 p:0013 s:0017 e:000016 BLOCK -e:1 [FINISH]
c:0003 p:--- s:0012 e:000011 CFUNC :times
```

```
-- Machine register context --------------------------------------------------
RIP: 0x00007f845d2d7a98 RBP: 0x00007f845e2691a0 RSP: 0x00007f845d2d7a98
RAX: 0x0000000000000000 RXB: 0x0000000000000000 RDX: 0xffffffff7fffffff
```

```
-- C level backtrace information ---------------------------------------------
/home/vmakarov/GITRUBY/build-tor/miniruby(sigsegv+0x3e) [0x7f845d2ce56e] /home/cygnus/vmakarov/GITRUBY/trunk/signal.c:897
```

```
/home/vmakarov/GITRUBY/build-tor/miniruby(rb_vm_bugreport+0x4f3) [0x7f845d362c53] /home/cygnus/vmakarov/GITRUBY/trunk/vm_dump.c:679
```

```
/home/vmakarov/GITRUBY/build-tor/miniruby(rb_bug_context+0xd4) [0x7f845d1d1084] /home/cygnus/vmakarov/GITRUBY/trunk/error.c:426
```

```
/home/vmakarov/GITRUBY/build-tor/miniruby(rb_vm_bugreport+0x4f3) [0x7f845d362c53] /home/cygnus/vmakarov/GITRUBY/trunk/signal.c:897
```
Associated revisions

Revision f46a43fc - 11/15/2016 12:27 AM - normal

st.c: fix crashes on huge hash tables

From: Vladimir Makarov vmakarov@redhat.com

By Vladimir's estimation, this requires at least 64 GB of memory to reproduce this bug due to the hash sizes required. So there is no new test case (and I am unable to test it, myself).

- st.c (get_bins_num): avoid out-of-bounds on shift by using correct type [ruby-core:78139] [Bug #12939]
- st.c (get_allocated_entries): ditto

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@56793 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 56793 - 11/15/2016 12:27 AM - normal

st.c: fix crashes on huge hash tables

From: Vladimir Makarov vmakarov@redhat.com

By Vladimir's estimation, this requires at least 64 GB of memory to reproduce this bug due to the hash sizes required. So there is no new test case (and I am unable to test it, myself).

- st.c (get_bins_num): avoid out-of-bounds on shift by using correct type [ruby-core:78139] [Bug #12939]
- st.c (get_allocated_entries): ditto

Revision 56793 - 11/15/2016 12:27 AM - normal

st.c: fix crashes on huge hash tables

From: Vladimir Makarov vmakarov@redhat.com

By Vladimir's estimation, this requires at least 64 GB of memory to reproduce this bug due to the hash sizes required. So there is no new test case (and I am unable to test it, myself).

- st.c (get_bins_num): avoid out-of-bounds on shift by using correct type [ruby-core:78139] [Bug #12939]
By Vladimir's estimation, this requires at least 64 GB of memory to reproduce this bug due to the hash sizes required. So there is no new test case (and I am unable to test it, myself).

- st.c (get_bins_num): avoid out-of-bounds on shift by using correct type [ruby-core:78139] [Bug #12939]
- st.c (get_allocated_entries): ditto

Revision 56793 - 11/15/2016 12:36 AM - normal
st.c: fix crashes on huge hash tables
From: normalperson (Eric Wong)

Thanks, committed as r56793

Files
st.patch 729 Bytes 11/14/2016 vmakarov (Vladimir Makarov)