Backport193 - Backport #5700

fork {} segfaults during VM cleanup when run inside Fiber

12/03/2011 10:37 AM - normalperson (Eric Wong)

Status:	Closed	
Priority:	Normal	
Assignee:		

Description

The issue is very easy to reproduce:

```
Fiber.new do
  p Process.waitpid2(fork {})
end.resume
```

Parent process successfully exits, child process segfaults while
it is exiting.

Backtrace is attached (gdb_bt.txt)

MALLOC_CHECK_=3 with GNU libc malloc() implementation detects an
attempt to free invalid pointer (attachment: malloc_check_3.txt)

I can also reproduce this with 1.9.2-p290 and 1.9.3-p0 as well as
latest trunk, so it has been around a while and a fix needs to be
backported.

Associated revisions

Revision 8cdf5c41 - 02/15/2012 10:34 PM - naruse (Yui NARUSE)

merge revision(s) 34629,34630:

```
    * cont.c (rb_fiber_reset_root_local_storage): add a new function to
      restore rb_thread_t::local_storage.

    * cont.c (rb_obj_is_fiber): add a new function to tell finalizer to
      prevent fibers from destroy.

    * gc.c (rb_objspace_call_finalizer): don't sweep fibers at finalizing
      objspace.

    * internal.h (rb_fiber_reset_root_local_storage, rb_obj_is_fiber):
      add prototypes.

    * vm.c (ruby_vm_destruct): reset main thread's local_storage before
      free main thread. rb_thread_t::local_storage is replaced by fiber's
      local storage when forked from fiber, and it should be already freed
      when the fiber was destroyed.

    * test/ruby/test_fiber.rb (test_fork_from_fiber): add test for fork
      from fiber.
      when the fiber was destroyed. [ruby-core:41456] [Bug #5700]
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@34637 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 34637 - 02/15/2012 10:34 PM - naruse (Yui NARUSE)

merge revision(s) 34629,34630:

```
* cont.c (rb_fiber_reset_root_local_storage): add a new function to
  restore rb_thread_t::local_storage.

* cont.c (rb_obj_is_fiber): add a new function to tell finalizer to
  prevent fibers from destroy.

* gc.c (rb_objspace_call_finalizer): don't sweep fibers at finalizing
  objspace.
```

```
* internal.h (rb_fiber_reset_root_local_storage, rb_obj_is_fiber):
  add prototypes.

* vm.c (ruby_vm_destruct): reset main thread's local_storage before
  free main thread. rb_thread_t::local_storage is replaced by fiber's
  local storage when forked from fiber, and it should be already freed
  when the fiber was destroyed.

* test/ruby/test_fiber.rb (test_fork_from_fiber): add test for fork
  from fiber.
  when the fiber was destroyed. [ruby-core:41456] [Bug #5700]
```

History

#1 - 02/08/2012 05:53 AM - normalperson (Eric Wong)

Eric Wong normalperson@yhbt.net wrote:

> http://redmine.ruby-lang.org/issues/5700

Hello, I would like to see this fixed for the next 1.9.3 release.

I'm pretty sure this is an easy fix for somebody already familiar with
the Fiber/continuation code (unfortunately I am not familiar with it).

#2 - 02/10/2012 12:30 PM - nagachika (Tomoyuki Chikanaga)

- File bug5700.patch added

Hmm, it don't seems easy problem for me... Apparently I'm not so familiar with fiber.

Anyway I've written a patch for this issue. I know it's somewhat ad-hoc, but it works.
ko1 and nobu how do you think?

#3 - 02/11/2012 12:23 PM - normalperson (Eric Wong)

Tomoyuki Chikanaga nagachika00@gmail.com wrote:

> Anyway I've written a patch for this issue. I know it's somewhat ad-hoc, but it works.
> ko1 and nobu how do you think?

Thanks for looking into this. This fix works for me, too :>

#4 - 02/11/2012 12:59 PM - nobu (Nobuyoshi Nakada)

One particular case is OK, two cases would be still OK, but three cases are not exceptional already.
We'll need some standard way to tell if a given object can be discarded immediately or not, I guess.

Anyway, it would be enough for the time being.

#5 - 02/14/2012 12:44 PM - nagachika (Tomoyuki Chikanaga)

I agree. In fact we don't have to protect all Fibers but only a root Fiber of main Thread.

But can I check-in previous patch for a temporary workaround? The next release of 1.9.3 is almost coming.

#6 - 02/15/2012 11:05 PM - nagachika (Tomoyuki Chikanaga)

- Tracker changed from Bug to Backport

- Project changed from Ruby master to Backport193

- Category deleted (core)

- Target version deleted (2.0.0)

I've committed at r34629, r34630. Please backport them.

#7 - 02/16/2012 07:34 AM - naruse (Yui NARUSE)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r34637.
Eric, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

---

merge revision(s) 34629,34630:

```
* cont.c (rb_fiber_reset_root_local_storage): add a new function to
  restore rb_thread_t::local_storage.

* cont.c (rb_obj_is_fiber): add a new function to tell finalizer to
  prevent fibers from destroy.

* gc.c (rb_objspace_call_finalizer): don't sweep fibers at finalizing
  objspace.

* internal.h (rb_fiber_reset_root_local_storage, rb_obj_is_fiber):
  add prototypes.

* vm.c (ruby_vm_destruct): reset main thread's local_storage before
  free main thread. rb_thread_t::local_storage is replaced by fiber's
  local storage when forked from fiber, and it should be already freed
  when the fiber was destroyed.

* test/ruby/test_fiber.rb (test_fork_from_fiber): add test for fork
  from fiber.
  when the fiber was destroyed. [ruby-core:41456] [Bug #5700]
```

Files

gdb_bt.txt	2.57 KB	12/03/2011	normalperson (Eric Wong)
malloc_check_3.txt	6.12 KB	12/03/2011	normalperson (Eric Wong)
fiber_fork.rb	54 Bytes	12/03/2011	normalperson (Eric Wong)
bug5700.patch	2.06 KB	02/10/2012	nagachika (Tomoyuki Chikanaga)