

Ruby master - Bug #13768

SIGCHLD and Thread dead-lock problem

07/26/2017 03:47 AM - keiju (Keiju Ishitsuka)

Status: Open	
Priority: Normal	
Assignee:	
Target version:	
ruby -v:	Backport: 2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

~~~~~

~~~~~Thread~~~~~

~~~~~, ~~~~~

~~~~~trap~~~~~  
~~~~~?

```
% ruby -v
ruby 2.5.0dev (2017-07-25 trunk 59417) [i686-linux]
```

```
q = Queue.new
p = Queue.new
```

```
trap(:SIGCHLD) do
  puts "SIGCHLD"
  q.push 1
end
```

```
Thread.start do
  Process.spawn("/bin/sleep 1")
end
```

```
#Thread.start do
# loop do
# sleep 100
# end
#end
```

```
th = Thread.start{
  p.push q.pop
}
```

```
p.pop
```

----->> ~ ~ <<---
----->> e-mail: keiju@ishitsuka.com <<---

History

#1 - 07/27/2017 04:13 AM - nobu (Nobuyoshi Nakada)

- Description updated

Signal.trap~~~~~
~~~~~

```
diff --git c/signal.c i/signal.c
index 2e69cf08ac..3b026724f9 100644
--- c/signal.c
+++ i/signal.c
```

```

@@ -1235,6 +1235,7 @@ trap(int sig, sighandler_t func, VALUE command)
    sighandler_t oldfunc;
    VALUE oldcmd;
    rb_vm_t *vm = GET_VM();
+   int handle_count = 0;

    /*
     * Be careful. ruby_signal() and trap_list[sig].cmd must be changed
@@ -1261,8 +1262,20 @@ trap(int sig, sighandler_t func, VALUE command)
    break;
    case Qundef:
    oldcmd = rb_str_new2("EXIT");
+   default:
+   --handle_count;
    break;
    }
    switch (command) {
+   case 0: case Qtrue: case Qnil:
+   break;
+   default:
+   ++handle_count;
+   break;
    }
+   if (vm->trap_count + handle_count < 0) rb_fatal("trap_count underflow");
+
+   vm->trap_count += handle_count;

    vm->trap_list[sig].cmd = command;
    vm->trap_list[sig].safe = rb_safe_level();
diff --git c/thread.c i/thread.c
index b7ee1d8d9b..fdb9221fd3 100644
--- c/thread.c
+++ i/thread.c
@@ -4947,6 +4947,7 @@ rb_check_deadlock(rb_vm_t *vm)
    if (vm_living_thread_num(vm) > vm->sleeper) return;
    if (vm_living_thread_num(vm) < vm->sleeper) rb_bug("sleeper must not be more than vm_living_thread_num(vm)");
    if (patrol_thread && patrol_thread != GET_THREAD()) return;
+   if (vm->trap_count > 0) return;

    list_for_each(&vm->living_threads, th, vmlt_node) {
    if (th->status != THREAD_STOPPED_FOREVER || RUBY_VM_INTERRUPTED(th)) {
diff --git c/vm_core.h i/vm_core.h
index c23f50690e..a2fa25ad50 100644
--- c/vm_core.h
+++ i/vm_core.h
@@ -550,6 +550,7 @@ typedef struct rb_vm_struct {
    VALUE cmd;
    int safe;
    } trap_list[RUBY_NSIG];
+   int trap_count;

    /* hook */
    rb_hook_list_t event_hooks;

```

**#2 - 07/27/2017 04:19 AM - ko1 (Koichi Sasada)**

dead-lock  
Signal.trap

webrick

**#3 - 07/27/2017 06:08 AM - keiju (Keiju Ishitsuka)**

.

[ko1@atdot.net](mailto:ko1@atdot.net) wrote:

Issue [#13768](#) has been updated by ko1 (Koichi Sasada).  
dead-lock

signal, select, (, )  
?  
?

Signal.trap

webrick

SIGINT

trap

ConditionVariable(SigCV)

SigCV.wait, signal, SigCV, signal, SigCV.wait

?

>> <<---  
>> e-mail: [keiju@ishitsuka.com](mailto:keiju@ishitsuka.com) <<---