

Ruby - Bug #6725

variables changed in signal handlers are left unchanged in main code

07/12/2012 04:59 PM - 375gnu (Hleb Valoshka)

Status:	Rejected	
Priority:	Normal	
Assignee:		
Target version:		
ruby -v:	1.9.3p194	Backport:
Description		
Test case:		
mkfifo myfifo ruby -e "a=0;trap(:USR2) {a+=1}; IO.readlines('myfifo') rescue nil; puts a" killall -USR2 ruby		
Output may be 0 or 1.		
It's tested with USR1, USR2 and ALRM signals with ruby 1.9.3p194 and today's 2.0.0dev on Debian GNU/Linux on AMD64.		
But if we add sleep(0) before puts, it will always output 1.		
More interesting thing is that we can wait some time and variable will be set. See attached file, here a part of it.		
alarm = 0 interval = (ENV["interval"] or 1).to_i delta = interval + 0.01 # on my box it's 0.001 for 1.9.3p194 and 0.01 for 2.0.0dev t0 = t1 = Time.now trap(:ALRM) { alarm += 1; t1 = Time.now } LibC::alarm(interval) ret = IO.readlines(ARGV[0]) rescue nil puts "alarm=#{alarm}, time=#{t1}, interval=#{t1-t0}" while Time.now - t0 < delta; end if ENV["interval"] puts "alarm=#{alarm}, time=#{t1}, interval=#{t1-t0}"		
And example output: alarm=0, time=2012-07-12 10:37:15 +0300, interval=0.0 alarm=1, time=2012-07-12 10:37:16 +0300, interval=1.000923864		
If variable delta is less than interval + upper_bound_for_actual_code_execution_time then the second output will be the same as the first one. In this example the first output is incorrect in >99%.		
(Instead of while sleep(0) may be used too.)		
Not every platform is affected by this bug. I've tested (all OS run on AMD64, real or under KVM): GNU/Linux, ruby 1.9.3p194, 2.0.0dev -- affected GNU/kFreeBSD ruby 1.9.3p194 -- not FreeBSD 9, ruby 1.9.3p0 -- not OpenBSD 5, ruby 1.9.3p0 -- affected		
Ruby 1.8 isn't affected, so it seems to be a bug with thread code.		

History

#1 - 07/12/2012 08:38 PM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Rejected

Asynchronous signals may not dealt with in the order you expect.

Files

test-signal.rb	562 Bytes	07/12/2012	375gnu (Hleb Valoshka)
----------------	-----------	------------	------------------------