Ruby master - Bug #11269

ruby_init_setproctitle() should be called before require_libraries()

06/16/2015 03:00 PM - apoikos (Apollon Oikonomopoulos)

Status:	Assigned		
Priority:	Normal		
Assignee:	kosaki (Motohiro KOSAKI)		
Target version:			
ruby -v:	ruby 2.1.5p273	**Backport:**	2.0.0: UNKNOWN, 2.1: UNKNOWN, 2.2: UNKNOWN

Description

We have come across an issue where setproctitle fails to use the whole length of the process's initial environment. It turns out that this is because modules loaded using -r are processed before ruby_init_setproctitle(). This is easily reproducible using bundler, given the following test script under Linux:

```
File.open('/proc/self/cmdline') do |f|
  @argv_len = f.read.size
end

File.open('/proc/self/environ') do |f|
  @env_len = f.read.size
end

total = @argv_len + @env_len
$0 = 'a' * (total + 2) # should not overflow

File.open('/proc/self/environ') do |f|
  env = f.read
  puts env
  puts env.size
end
```

Calling the script directly, we see that setproctitle works correctly:

```
$ env -i A_VAR=some_value PATH=/bin:/usr/bin ruby test.rb
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
36
```

In this case setproctitle correctly overwrites the whole environment space available. However, if we require bundler/setup (which is what bundle exec essentially does), the following happens:

```
$ env -i A_VAR=some_value PATH=/bin:/usr/bin ruby -rbundler/setup test.rb
aaaaaaaaaaaaaaaaPATH=/bin:/usr/bin
36
```

This is because require 'bundler/setup' is processed before ruby_init_setproctitle() is called and attempts to manipulate PATH. While doing so, the new PATH string is allocated on the heap and the original pointer in environ replaced, thus breaking ruby_init_setproctitle()'s expectation to find a contiguous environment block.

Perhaps ruby_init_setproctitle() can be called before process_options(), so that it can process the environment block before a library gets a chance of modifying it.

History

#1 - 06/17/2015 03:01 AM - kosaki (Motohiro KOSAKI)

- Status changed from Open to Assigned

- Assignee set to kosaki (Motohiro KOSAKI)

I agree.
I'll take a look.