Ruby - Bug #6249

Process.exec doesn't restore the environment if it fails

04/03/2012 08:06 AM - john_firebaugh (John Firebaugh)

Status:	Closed		
Priority:	Normal		
Assignee:	akr (Akira Tanaka)		
Target version:			
ruby -v:	ruby 1.9.3p125 (2012-02-16 revision 34643) [x86_64-darwin11.3.0]	**Backport:**	

Description

```
ENV["foo"]
=> nil
Process.exec({"foo" => "bar"}, "nonexistent")
Errno::ENOENT: No such file or directory - nonexistent
from (irb):2:in exec'  from (irb):2  from /Users/john/.rvm/rubies/ruby-1.9.3-p125/bin/irb:16:in '
ENV["foo"]
=> "bar"
```

I expected that Process.exec would either use execle or execve, or (if it implements environment modification itself), to manually restore the existing environment upon failure.

Associated revisions

Revision f4f28bf75f0aa534de3b1edcf46c8a6ee905e9af - 06/03/2012 08:29 AM - akr (Akira Tanaka)

- use execve() to preserve environment variables when exec method is failed. [ruby-core:44093] [ruby-trunk - Bug #6249]
- include/ruby/intern.h (rb_exec_arg): add envp_str and envp_buf field to store envp of execve().
- process.c (proc_exec_v): takes envp_str as an argument and use it for execve().
  (rb_proc_exec_ne): extended version of rb_proc_exec_n().
  (rb_proc_exec_n): use rb_proc_exec_ne().
  (rb_proc_exec): follow proc_exec_v() change.
  (fill_envp_buf_i): new function.
  (rb_exec_arg_fixup): set up envp_str and envp_buf.
  (save_env_i): removed.
  (save_env): removed.
  (rb_run_exec_options_err): don't modify environment variables.
  (rb_exec_err): use rb_proc_exec_ne().

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35882 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision f4f28bf7 - 06/03/2012 08:29 AM - akr (Akira Tanaka)

- use execve() to preserve environment variables when exec method is failed. [ruby-core:44093] [ruby-trunk - Bug #6249]
- include/ruby/intern.h (rb_exec_arg): add envp_str and envp_buf field to store envp of execve().
- process.c (proc_exec_v): takes envp_str as an argument and use it for execve().

(rb_proc_exec_ne): extended version of rb_proc_exec_n().
(rb_proc_exec_n): use rb_proc_exec_ne().
(rb_proc_exec): follow proc_exec_v() change.
(fill_envp_buf_i): new function.
(rb_exec_arg_fixup): set up envp_str and envp_buf.
(save_env_i): removed.
(save_env): removed.
(rb_run_exec_options_err): don't modify environment variables.
(rb_exec_err): use rb_proc_exec_ne().

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35882 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 2493e48d40bb984f315be0e478bf3b9f0dec588c - 06/04/2012 02:36 AM - U.Nakamura

- process.c (rb_run_exec_options_err): restore save_env() call for non-fork environments.
- process.c (rb_exec_err): restore environments after the failure of exec to fix [ruby-core:44093] [Bug #6249] on non-fork environments

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35897 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 2493e48d - 06/04/2012 02:36 AM - U.Nakamura

- process.c (rb_run_exec_options_err): restore save_env() call for non-fork environments.
- process.c (rb_exec_err): restore environments after the failure of exec to fix [ruby-core:44093] [Bug #6249] on non-fork environments

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@35897 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 04/04/2012 06:23 PM - akr (Akira Tanaka)

2012/4/3 john_firebaugh (John Firebaugh) john.firebaugh@gmail.com:

```
ENV["foo"]
=> nil
Process.exec({"foo" => "bar"}, "nonexistent")
Errno::ENOENT: No such file or directory - nonexistent
from (irb):2:in exec'       from (irb):2       from /Users/john/.rvm/rubies/ruby-1.9.3-p125/bin/irb:16:in '
ENV["foo"]
=> "bar"
```

> I expected that Process.exec would either use execle or execve, or (if it implements environment modification itself), to manually restore the existing environment upon failure.

I see.

I think Process.exec should use execve.

Tanaka Akira

#2 - 04/10/2012 02:38 PM - mame (Yusuke Endoh)

- Status changed from Open to Assigned

- Assignee set to akr (Akira Tanaka)

#3 - 06/03/2012 05:29 PM - akr (Akira Tanaka)

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r35882.
John, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- use execve() to preserve environment variables when exec method is failed. [ruby-core:44093] [ruby-trunk - Bug #6249]

- include/ruby/intern.h (rb_exec_arg): add envp_str and envp_buf field to store envp of execve().

- process.c (proc_exec_v): takes envp_str as an argument and use it for execve().
  (rb_proc_exec_ne): extended version of rb_proc_exec_n().
  (rb_proc_exec_n): use rb_proc_exec_ne().
  (rb_proc_exec): follow proc_exec_v() change.
  (fill_envp_buf_i): new function.
  (rb_exec_arg_fixup): set up envp_str and envp_buf.
  (save_env_i): removed.
  (save_env): removed.
  (rb_run_exec_options_err): don't modify environment variables.
  (rb_exec_err): use rb_proc_exec_ne().