Ruby master - Feature #5041

Set FD_CLOEXEC for all fds (except 0, 1, 2)

07/18/2011 01:15 AM - akr (Akira Tanaka)

Status:	Closed
Priority:	Normal
Assignee:	akr (Akira Tanaka)
Target version:	

Description

I'd like to set FD_CLOEXEC for all file descriptors (except 0, 1, 2,
i.e. standard input/output/error).

I talked this issue with kosaki and matz at RubyKaigi 2011 and
matz said "do it" and see that someone will cry or not.

FD_CLOEXEC prevents fd leakage for command execution.
See the problem of fd leakage for
"FIO42-C. Ensure files are properly closed when they are no longer needed".
https://www.securecoding.cert.org/confluence/display/seccode/FIO42-C.+Ensure+files+are+properly+closed+when+they+are+no+longer+needed

This is an incompatible change for programs which use fd leakage intentionally.
For example, valgrind has options such as --log-fd=, --input-fd=, etc.
gpg has --status-fd, --logger-fd, etc.
openssl command has -passin fd:number and -passout fd:number.
xterm has -S option which takes a fd.
...

Currently, system() and exec() method leak fds.
But IO.popen() and spawn() doesn't leak fds.
Windows doesn't inherit fds for child processes.
So this issue is only affected to system() and exec() on Unix.

(spawn(), which is available since Ruby 1.9.1, doesn't leak fds because
:close_others option is true by default.
IO.popen() doesn't leak fds since [ruby-dev:457].
The behavior is preserved for Ruby 1.9 by :close_others is true by default
for IO.popen().)

If a program uses fd leakage, system() and exec() call should be changed.
For example,
system("valgrind", "--log-fd=#{N}", ...) should be changed to
system("valgrind", "--log-fd=#{N}", ..., N=>N).
See the document of spawn() for details of the option N=>N.
This option, N=>N, can be used since Ruby 1.9.1.

FD_CLOEXEC is set by fcntl(F_SETFD) on Unix.
However Ruby can use O_CLOEXEC, dup3 and other new mechanisms to avoid
race conditions if they are available.

The race condition is real problem because Ruby invokes open() system call in
a blocking region to open a named pipe without stucking.
So, new fd can be born at any point.
This means the new fd (without FD_CLOEXEC) may be born just before fork().
This race can be fixed by O_CLOEXEC (if available).
The semantics of "FD_CLOEXEC for all fds" makes us possible to use O_CLOEXEC
without harm.

Subtasks:

Bug # 5475: r33507□□Solaris□PTY□□□□□	**Closed**

Related issues:

Related to Ruby master - Feature #4512: [PATCH] ext/fcntl/fcntl.c: add F_DUPF...	**Closed**	**03/20/2011**

Has duplicate Ruby master - Feature #1291: O_CLOEXEC flag missing for Kernel:...	**Closed**	**03/15/2009**

Associated revisions

Revision b574a4d4 - 10/22/2011 09:58 AM - akr (Akira Tanaka)

- include/ruby/intern.h (rb_fd_set_cloexec): declared.
- io.c (rb_fd_set_cloexec): new function.
  (ruby_dup): call rb_fd_set_cloexec to set close-on-exec flag.
  (rb_sysopen_internal): ditto.
  (rb_pipe): ditto.
  (io_reopen): ditto.
  (io_cntl): ditto.
- process.c (rb_f_exec): change the default :close_others option to true.
  (rb_f_system): ditto.
  (move_fds_to_avoid_crash): call rb_fd_set_cloexec to set close-on-exec flag.
  (ruby_setsid): ditto.
  (rb_daemon): ditto.
- thread_pthread.c (rb_thread_create_timer_thread): call rb_fd_set_cloexec to set close-on-exec flag.
- ruby.c (load_file_internal): ditto.
- file.c (rb_file_s_truncate): ditto.
  (file_load_ok): ditto.
- random.c (fill_random_seed): ditto.
- ext/pty/pty.c (chfunc): ditto.
  (get_device_once): ditto.
- ext/openssl/ossl_bio.c (ossl_obj2bio): ditto.
- ext/socket/init.c (rsock_socket): ditto.
  (rsock_s_accept_nonblock): ditto.
  (rsock_s_accept): ditto.
- ext/socket/socket.c (rsock_sock_s_socketpair): ditto.
- ext/socket/ancdata.c (discard_cmsg): ditto.
  (make_io_for_unix_rights): ditto.
- ext/socket/unixsocket.c (unix_recv_io): ditto.
- ext/io/console/console.c (console_dev): ditto.

[ruby-core:38140] [Feature #5041]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@33507 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 33507 - 10/22/2011 09:58 AM - akr (Akira Tanaka)

- include/ruby/intern.h (rb_fd_set_cloexec): declared.
- io.c (rb_fd_set_cloexec): new function.
  (ruby_dup): call rb_fd_set_cloexec to set close-on-exec flag.
  (rb_sysopen_internal): ditto.
  (rb_pipe): ditto.
  (io_reopen): ditto.
  (io_cntl): ditto.
- process.c (rb_f_exec): change the default :close_others option to true.

(rb_f_system): ditto.
(move_fds_to_avoid_crash): call rb_fd_set_cloexec to set close-on-exec flag.
(ruby_setsid): ditto.
(rb_daemon): ditto.

- thread_pthread.c (rb_thread_create_timer_thread): call rb_fd_set_cloexec to set close-on-exec flag.
- ruby.c (load_file_internal): ditto.
- file.c (rb_file_s_truncate): ditto.
  (file_load_ok): ditto.
- random.c (fill_random_seed): ditto.
- ext/pty/pty.c (chfunc): ditto.
  (get_device_once): ditto.
- ext/openssl/ossl_bio.c (ossl_obj2bio): ditto.
- ext/socket/init.c (rsock_socket): ditto.
  (rsock_s_accept_nonblock): ditto.
  (rsock_s_accept): ditto.
- ext/socket/socket.c (rsock_sock_s_socketpair): ditto.
- ext/socket/ancdata.c (discard_cmsg): ditto.
  (make_io_for_unix_rights): ditto.
- ext/socket/unixsocket.c (unix_recv_io): ditto.
- ext/io/console/console.c (console_dev): ditto.

[ruby-core:38140] [Feature #5041]

Revision 33507 - 10/22/2011 09:58 AM - akr (Akira Tanaka)

- include/ruby/intern.h (rb_fd_set_cloexec): declared.
- io.c (rb_fd_set_cloexec): new function.
  (ruby_dup): call rb_fd_set_cloexec to set close-on-exec flag.
  (rb_sysopen_internal): ditto.
  (rb_pipe): ditto.
  (io_reopen): ditto.
  (io_cntl): ditto.
- process.c (rb_f_exec): change the default :close_others option to true.
  (rb_f_system): ditto.
  (move_fds_to_avoid_crash): call rb_fd_set_cloexec to set close-on-exec flag.
  (ruby_setsid): ditto.
  (rb_daemon): ditto.
- thread_pthread.c (rb_thread_create_timer_thread): call rb_fd_set_cloexec to set close-on-exec flag.
- ruby.c (load_file_internal): ditto.
- file.c (rb_file_s_truncate): ditto.
  (file_load_ok): ditto.
- random.c (fill_random_seed): ditto.
- ext/pty/pty.c (chfunc): ditto.
  (get_device_once): ditto.
- ext/openssl/ossl_bio.c (ossl_obj2bio): ditto.
- ext/socket/init.c (rsock_socket): ditto.

(rsock_s_accept_nonblock): ditto.
(rsock_s_accept): ditto.

- ext/socket/socket.c (rsock_sock_s_socketpair): ditto.
- ext/socket/ancdata.c (discard_cmsg): ditto.
(make_io_for_unix_rights): ditto.
- ext/socket/unixsocket.c (unix_recv_io): ditto.
- ext/io/console/console.c (console_dev): ditto.

[ruby-core:38140] [Feature #5041]

Revision 33507 - 10/22/2011 09:58 AM - akr (Akira Tanaka)

- include/ruby/intern.h (rb_fd_set_cloexec): declared.
- io.c (rb_fd_set_cloexec): new function.
(ruby_dup): call rb_fd_set_cloexec to set close-on-exec flag.
(rb_sysopen_internal): ditto.
(rb_pipe): ditto.
(io_reopen): ditto.
(io_cntl): ditto.
- process.c (rb_f_exec): change the default :close_others option to true.
(rb_f_system): ditto.
(move_fds_to_avoid_crash): call rb_fd_set_cloexec to set close-on-exec flag.
(ruby_setsid): ditto.
(rb_daemon): ditto.
- thread_pthread.c (rb_thread_create_timer_thread): call rb_fd_set_cloexec to set close-on-exec flag.
- ruby.c (load_file_internal): ditto.
- file.c (rb_file_s_truncate): ditto.
(file_load_ok): ditto.
- random.c (fill_random_seed): ditto.
- ext/pty/pty.c (chfunc): ditto.
(get_device_once): ditto.
- ext/openssl/ossl_bio.c (ossl_obj2bio): ditto.
- ext/socket/init.c (rsock_socket): ditto.
(rsock_s_accept_nonblock): ditto.
(rsock_s_accept): ditto.
- ext/socket/socket.c (rsock_sock_s_socketpair): ditto.
- ext/socket/ancdata.c (discard_cmsg): ditto.
(make_io_for_unix_rights): ditto.
- ext/socket/unixsocket.c (unix_recv_io): ditto.
- ext/io/console/console.c (console_dev): ditto.

[ruby-core:38140] [Feature #5041]

Revision 33507 - 10/22/2011 09:58 AM - akr (Akira Tanaka)

- include/ruby/intern.h (rb_fd_set_cloexec): declared.

- io.c (rb_fd_set_cloexec): new function.
  (ruby_dup): call rb_fd_set_cloexec to set close-on-exec flag.
  (rb_sysopen_internal): ditto.
  (rb_pipe): ditto.
  (io_reopen): ditto.
  (io_cntl): ditto.

- process.c (rb_f_exec): change the default :close_others option to true.
  (rb_f_system): ditto.
  (move_fds_to_avoid_crash): call rb_fd_set_cloexec to set close-on-exec flag.
  (ruby_setsid): ditto.
  (rb_daemon): ditto.

- thread_pthread.c (rb_thread_create_timer_thread): call rb_fd_set_cloexec to set close-on-exec flag.

- ruby.c (load_file_internal): ditto.

- file.c (rb_file_s_truncate): ditto.
  (file_load_ok): ditto.

- random.c (fill_random_seed): ditto.

- ext/pty/pty.c (chfunc): ditto.
  (get_device_once): ditto.

- ext/openssl/ossl_bio.c (ossl_obj2bio): ditto.

- ext/socket/init.c (rsock_socket): ditto.
  (rsock_s_accept_nonblock): ditto.
  (rsock_s_accept): ditto.

- ext/socket/socket.c (rsock_sock_s_socketpair): ditto.

- ext/socket/ancdata.c (discard_cmsg): ditto.
  (make_io_for_unix_rights): ditto.

- ext/socket/unixsocket.c (unix_recv_io): ditto.

- ext/io/console/console.c (console_dev): ditto.

[ruby-core:38140] [Feature #5041]

Revision 33507 - 10/22/2011 09:58 AM - akr (Akira Tanaka)

- include/ruby/intern.h (rb_fd_set_cloexec): declared.

- io.c (rb_fd_set_cloexec): new function.
  (ruby_dup): call rb_fd_set_cloexec to set close-on-exec flag.
  (rb_sysopen_internal): ditto.
  (rb_pipe): ditto.
  (io_reopen): ditto.
  (io_cntl): ditto.

- process.c (rb_f_exec): change the default :close_others option to true.
  (rb_f_system): ditto.
  (move_fds_to_avoid_crash): call rb_fd_set_cloexec to set close-on-exec flag.
  (ruby_setsid): ditto.
  (rb_daemon): ditto.

- thread_pthread.c (rb_thread_create_timer_thread): call rb_fd_set_cloexec to set close-on-exec flag.

- ruby.c (load_file_internal): ditto.

- file.c (rb_file_s_truncate): ditto.
  (file_load_ok): ditto.

- random.c (fill_random_seed): ditto.
- ext/pty/pty.c (chfunc): ditto.
  (get_device_once): ditto.
- ext/openssl/ossl_bio.c (ossl_obj2bio): ditto.
- ext/socket/init.c (rsock_socket): ditto.
  (rsock_s_accept_nonblock): ditto.
  (rsock_s_accept): ditto.
- ext/socket/socket.c (rsock_sock_s_socketpair): ditto.
- ext/socket/ancdata.c (discard_cmsg): ditto.
  (make_io_for_unix_rights): ditto.
- ext/socket/unixsocket.c (unix_recv_io): ditto.
- ext/io/console/console.c (console_dev): ditto.

[ruby-core:38140] [Feature #5041]

Revision 33507 - 10/22/2011 09:58 AM - akr (Akira Tanaka)

- include/ruby/intern.h (rb_fd_set_cloexec): declared.
- io.c (rb_fd_set_cloexec): new function.
  (ruby_dup): call rb_fd_set_cloexec to set close-on-exec flag.
  (rb_sysopen_internal): ditto.
  (rb_pipe): ditto.
  (io_reopen): ditto.
  (io_cntl): ditto.
- process.c (rb_f_exec): change the default :close_others option to true.
  (rb_f_system): ditto.
  (move_fds_to_avoid_crash): call rb_fd_set_cloexec to set close-on-exec flag.
  (ruby_setsid): ditto.
  (rb_daemon): ditto.
- thread_pthread.c (rb_thread_create_timer_thread): call rb_fd_set_cloexec to set close-on-exec flag.
- ruby.c (load_file_internal): ditto.
- file.c (rb_file_s_truncate): ditto.
  (file_load_ok): ditto.
- random.c (fill_random_seed): ditto.
- ext/pty/pty.c (chfunc): ditto.
  (get_device_once): ditto.
- ext/openssl/ossl_bio.c (ossl_obj2bio): ditto.
- ext/socket/init.c (rsock_socket): ditto.
  (rsock_s_accept_nonblock): ditto.
  (rsock_s_accept): ditto.
- ext/socket/socket.c (rsock_sock_s_socketpair): ditto.
- ext/socket/ancdata.c (discard_cmsg): ditto.
  (make_io_for_unix_rights): ditto.
- ext/socket/unixsocket.c (unix_recv_io): ditto.
- ext/io/console/console.c (console_dev): ditto.

[ruby-core:38140] [Feature #5041]

History

#1 - 07/21/2011 07:53 AM - normalperson (Eric Wong)

Akira Tanaka akr@fsij.org wrote:

> I'd like to set FD_CLOEXEC for all file descriptors (except 0, 1, 2,
> i.e. standard input/output/error).
>
> I talked this issue with kosaki and matz at RubyKaigi 2011 and
> matz said "do it" and see that someone will cry or not.

I support this proposal for Ruby 2.0. Very few applications depend on
FD passing via exec() and they can easily be updated to set
close_on_exec=false.

I've just updated git://bogomips.org/unicorn.git myself.

#2 - 07/21/2011 08:25 AM - akr (Akira Tanaka)

- Assignee set to akr (Akira Tanaka)

Eric Wong wrote:

> I support this proposal for Ruby 2.0. Very few applications depend on
> FD passing via exec() and they can easily be updated to set
> close_on_exec=false.
>
> I've just updated git://bogomips.org/unicorn.git myself.

Thank you for your support for this issue.

My (and matz's) intent is for 1.9.4.
I'm not sure the next version will be 1.9.4 or 2.0, though.

I don't recommend io.close_on_exec = false for multithreaded programs.
There is a race condition which cause fd leakage if another thread invokes
system().
(I guess unicorn has no problem because it is not multithreaded.)

So I may change the default of :close_others to true even for system() and
exec().

#3 - 07/21/2011 08:48 AM - kstephens (Kurt Stephens)

> Eric Wong wrote:
> I don't recommend io.close_on_exec = false for multithreaded programs.
> There is a race condition which cause fd leakage if another thread invokes
> system().
> (I guess unicorn has no problem because it is not multithreaded.)

We commonly dup FD 2 so subprocesses can drill back out to parent's $STDERR, after parent has redirected FD 2 to /dev/null.

Ruby needs a generic callback upon Process.fork.

See:
https://github.com/kstephens/ruby_is_forked
https://github.com/kstephens/rails_is_forked

#4 - 07/21/2011 09:23 AM - normalperson (Eric Wong)

Akira Tanaka akr@fsij.org wrote:

> Eric Wong wrote:
>
> > I support this proposal for Ruby 2.0. Very few applications depend on
> > FD passing via exec() and they can easily be updated to set
> > close_on_exec=false.
> >
> > I've just updated git://bogomips.org/unicorn.git myself.

> I don't recommend io.close_on_exec = false for multithreaded programs.
> There is a race condition which cause fd leakage if another thread invokes
> system().
> (I guess unicorn has no problem because it is not multithreaded.)

Yeah, and even in Rainbows!, threads only get used in the worker
processes, not the master process[1] that calls exec()

> So I may change the default of :close_others to true even for system() and
> exec().

If so, I would like a way specify a set/array/hash of FDs/IOs we don't
want :close_others to automatically close on us.

[1] - Zbatery is a different story, but that's just one extra caveat to
running with (optional) threads :<

--
Eric Wong

#5 - 07/22/2011 11:53 PM - akr (Akira Tanaka)

2011/7/21 Eric Wong normalperson@yhbt.net:

> > So I may change the default of :close_others to true even for system() and
> > exec().
>
> > If so, I would like a way specify a set/array/hash of FDs/IOs we don't
> > want :close_others to automatically close on us.

If you want to disable automatic close fd1, fd2, ..., use follows.

```
h = { fd1 => fd1, fd2 => fd2, ... }
system("command", h)
```

system() (, exec(), IO.popen and spawn()) can take an option hash
to specify fds to inherit to child process.

See the manual of spawn() for details.
--
Tanaka Akira

#6 - 07/22/2011 11:59 PM - akr (Akira Tanaka)

2011/7/21 Kurt Stephens ks.ruby@kurtstephens.com:

> > We commonly dup FD 2 so subprocesses can drill back out to parent's $STDERR, after parent has redirected FD 2 to /dev/null.
>
> > We need a generic callback on Process.fork.

I can't understand how it is related to this issue.
--
Tanaka Akira

#7 - 08/11/2011 02:43 PM - akr (Akira Tanaka)

- File close-on-exec-by-default.patch added

I made a patch to set FD_CLOEXEC by default.
It changes close_others option true by default for system() and exec().

#8 - 08/12/2011 05:23 AM - normalperson (Eric Wong)

Akira Tanaka akr@fsij.org wrote:

> Issue #5041 has been updated by Akira Tanaka.
>
> File close-on-exec-by-default.patch added

I made a patch to set FD_CLOEXEC by default.
It changes close_others option true by default for system() and exec().

Thanks, unicorn.git works great with your patch after the following change:

http://bogomips.org/unicorn.git/patch?id=6ab27beeda

#9 - 10/22/2011 06:58 PM - akr (Akira Tanaka)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r33507.
Akira, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- include/ruby/intern.h (rb_fd_set_cloexec): declared.
- io.c (rb_fd_set_cloexec): new function.
  (ruby_dup): call rb_fd_set_cloexec to set close-on-exec flag.
  (rb_sysopen_internal): ditto.
  (rb_pipe): ditto.
  (io_reopen): ditto.
  (io_cntl): ditto.
- process.c (rb_f_exec): change the default :close_others option to true.
  (rb_f_system): ditto.
  (move_fds_to_avoid_crash): call rb_fd_set_cloexec to set close-on-exec flag.
  (ruby_setsid): ditto.
  (rb_daemon): ditto.
- thread_pthread.c (rb_thread_create_timer_thread): call rb_fd_set_cloexec to set close-on-exec flag.
- ruby.c (load_file_internal): ditto.
- file.c (rb_file_s_truncate): ditto.
  (file_load_ok): ditto.
- random.c (fill_random_seed): ditto.
- ext/pty/pty.c (chfunc): ditto.
  (get_device_once): ditto.
- ext/openssl/ossl_bio.c (ossl_obj2bio): ditto.
- ext/socket/init.c (rsock_socket): ditto.
  (rsock_s_accept_nonblock): ditto.
  (rsock_s_accept): ditto.
- ext/socket/socket.c (rsock_sock_s_socketpair): ditto.
- ext/socket/ancdata.c (discard_cmsg): ditto.
  (make_io_for_unix_rights): ditto.
- ext/socket/unixsocket.c (unix_recv_io): ditto.
- ext/io/console/console.c (console_dev): ditto.

[ruby-core:38140] [Feature #5041]

Files

close-on-exec-by-default.patch	18.6 KB	08/11/2011	akr (Akira Tanaka)