

Ruby - Bug #20995

exception escapes block given to IO.popen("-") in child process

12/31/2024 11:55 PM - martin.dorey@hds.com (Martin Dorey)

|                 |        |  |
|-----------------|--------|--|
| Status:         | Closed |  |
| Priority:       | Normal |  |
| Assignee:       |        |  |
| Target version: |        |  |
| ruby -v:        |        | Backport: 3.1: REQUIRED, 3.2: DONE, 3.3: DONE, 3.4: DONE |

**Description**

I was surprised by the "ensure" being reached in the child process here:

```
martind@stormy:~/tmp/D161730$ cat repro.rb
#!/usr/bin/ruby -w
parent = Process.pid()
[false, true].each() {
  |nauseous|
  $stderr.puts("#{nauseous ? " " : "not "}raising exception from child:")
  begin
    IO.popen("-") {
      |io|
      unless io
        if nauseous
          raise("childish fit")
        end
      end
    }
  ensure
    $stderr.puts("in finalization block from #{Process.pid() == parent ? "parent" : "child"}")
    $stderr.puts()
  end
}
```

```
martind@stormy:~/tmp/D161730$ ~/download/ruby-3.3.4/ruby --version
ruby 3.3.4 (2024-07-09 revision be1089c8ec) [x86_64-linux]
martind@stormy:~/tmp/D161730$ ~/download/ruby-3.3.4/ruby -w repro.rb
not raising exception from child:
in finalization block from parent

raising exception from child:
in finalization block from child

repro.rb:11:in `block (2 levels) in <main>': childish fit (RuntimeError)
from repro.rb:7:in `popen'
from repro.rb:7:in `block in <main>'
from repro.rb:3:in `each'
from repro.rb:3:in `<main>'
in finalization block from parent

martind@stormy:~/tmp/D161730$
```

In the first iteration of the loop, we don't see the surprise - the child process exits when the block is completed. Only when an exception is thrown, in the second iteration, does the control flow escape from the block. It does so even for the SystemExit exception raised by exit(0) but not, as we see above, from just leaving the block (in which case perhaps we \_exit(0) at <https://github.com/ruby/blob/master/io.c#L8044>). My expectation was that the block would be executed in the child process much like the "main" part of a program, with the usual unhandled exception reporting, which we see demonstrated above, being invoked as soon as the exception propagated out the block, rather than first unwinding the callers.

I think the behavior is unchanged since the oldest version I was conveniently able to test, which was:

```
ruby 1.8.7 (2012-02-08 patchlevel 358) [x86_64-linux]
```

... so I didn't file this report as a Bug in case users have (accidentally?) come to depend on the behavior. I didn't find any mention of this at eg <https://ruby-doc.org/3.4.1/IO.html#method-c-popen> but I didn't file this report as a Feature request for the documentation in case the documentation deliberately leaves open such possibilities. I didn't even find anyone else puzzling over this behavior on eg StackOverflow, but I felt that I should report it for the benefit of the next poor sap to run into it, especially in case that's Future Me. Could it be a case where a warning would be worth more than the trouble it causes?

Associated revisions

Revision 8034e9c3d001ca3dff124ab42972684eac8af2ae - 01/02/2025 06:50 AM - nobu (Nobuyoshi Nakada)

[Bug #20995] Protect IO.popen block from exiting by exception

Revision 8034e9c3d001ca3dff124ab42972684eac8af2ae - 01/02/2025 06:50 AM - nobu (Nobuyoshi Nakada)

[Bug #20995] Protect IO.popen block from exiting by exception

Revision 8034e9c3 - 01/02/2025 06:50 AM - nobu (Nobuyoshi Nakada)

[Bug #20995] Protect IO.popen block from exiting by exception

Revision f150d67b7d389eb88e0cd13694d3529895d55579 - 01/11/2025 06:30 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 8034e9c3d001ca3dff124ab42972684eac8af2ae: [Backport #20995]

[Bug #20995] Protect `IO.popen` block from exiting by exception

Revision f150d67b7d389eb88e0cd13694d3529895d55579 - 01/11/2025 06:30 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 8034e9c3d001ca3dff124ab42972684eac8af2ae: [Backport #20995]

[Bug #20995] Protect `IO.popen` block from exiting by exception

Revision f150d67b - 01/11/2025 06:30 AM - nagachika (Tomoyuki Chikanaga)

merge revision(s) 8034e9c3d001ca3dff124ab42972684eac8af2ae: [Backport #20995]

[Bug #20995] Protect `IO.popen` block from exiting by exception

Revision 12a0807965624a0be37dc79371a69b5d787cc8d1 - 01/15/2025 01:59 AM - k0kubun (Takashi Kokubun)

merge revision(s) 8034e9c3d001ca3dff124ab42972684eac8af2ae: [Backport #20995]

[Bug #20995] Protect `IO.popen` block from exiting by exception

Revision 12a0807965624a0be37dc79371a69b5d787cc8d1 - 01/15/2025 01:59 AM - k0kubun (Takashi Kokubun)

merge revision(s) 8034e9c3d001ca3dff124ab42972684eac8af2ae: [Backport #20995]

[Bug #20995] Protect `IO.popen` block from exiting by exception

Revision 12a08079 - 01/15/2025 01:59 AM - k0kubun (Takashi Kokubun)

merge revision(s) 8034e9c3d001ca3dff124ab42972684eac8af2ae: [Backport #20995]

[Bug #20995] Protect `IO.popen` block from exiting by exception

Revision b65cea74295358265dfabc9e1f4d107b21e58e58 - 02/14/2025 02:01 AM - k0kubun (Takashi Kokubun)

merge revision(s) 8034e9c3d001ca3dff124ab42972684eac8af2ae: [Backport #20995]

[Bug #20995] Protect `IO.popen` block from exiting by exception

Revision b65cea74295358265dfabc9e1f4d107b21e58e58 - 02/14/2025 02:01 AM - k0kubun (Takashi Kokubun)

merge revision(s) 8034e9c3d001ca3dff124ab42972684eac8af2ae: [Backport #20995]

[Bug #20995] Protect `IO.popen` block from exiting by exception

Revision b65cea74 - 02/14/2025 02:01 AM - k0kubun (Takashi Kokubun)

merge revision(s) 8034e9c3d001ca3dff124ab42972684eac8af2ae: [Backport #20995]

[Bug #20995] Protect `IO.popen` block from exiting by exception

## History

### #1 - 01/02/2025 06:39 AM - nobu (Nobuyoshi Nakada)

- Tracker changed from Misc to Bug
- Backport set to 3.1: REQUIRED, 3.2: REQUIRED, 3.3: REQUIRED, 3.4: REQUIRED

It seems a bug.

### #2 - 01/02/2025 08:10 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

Applied in changeset [git|8034e9c3d001ca3dff124ab42972684eac8af2ae](https://github.com/ruby/ruby/commit/8034e9c3d001ca3dff124ab42972684eac8af2ae).

[Bug #20995] Protect IO.popen block from exiting by exception

### #3 - 01/03/2025 02:13 AM - martin.dorey@hds.com (Martin Dorey)

Domo arigato, Nobu-san, for agreeing that the previous behavior wasn't right and for such a quick fix. I really like the idea of expressing how the code should behave with a unit test. I was surprised by a couple of aspects of your test. This is how I expected it to behave:

```
martind@stormy:~/download/ruby$ git diff
...
diff --git a/test/ruby/test_process.rb b/test/ruby/test_process.rb
index 9c1836dffb..781fe77b7d 100644
--- a/test/ruby/test_process.rb
+++ b/test/ruby/test_process.rb
@@ -939,9 +939,10 @@ def test_popen_fork_ensure
     STDERR.reopen(STDOUT)
     raise "foo"
   else
-    assert_empty io.read
+    assert_include(io.read, "foo")
   end
 end
+  assert_not_predicate($?, :success?)
  rescue RuntimeError
    abort "[Bug #20995] should not reach here"
  end
end
martind@stormy:~/download/ruby$
```

I have very little idea whether this is safe but, with the following change, the above test passes:

```
martind@stormy:~/download/ruby$ git diff
diff --git a/io.c b/io.c
index a78936488f..612c9caa40 100644
--- a/io.c
+++ b/io.c
@@ -8038,10 +8038,11 @@ popen_finish(VALUE port, VALUE klass)
   if (NIL_P(port)) {
     /* child */
     if (rb_block_given_p()) {
-      rb_protect(rb_yield, Qnil, NULL);
+      int state;
+      rb_protect(rb_yield, Qnil, &state);
       rb_io_flush(rb_ractor_stdout());
       rb_io_flush(rb_ractor_stderr());
-      _exit(0);
+      _exit(ruby_cleanup(state));
     }
     return Qnil;
   }
}
```

With that change, these ancestral behaviors:

```
martind@stormy:~/download/ruby$ ../ruby-3.3.4/ruby --disable-gems -we 'IO.popen("-") { |io| raise("a fuss") unless io; }; puts($?.inspect())'
-e:1:in `block in <main>': a fuss (RuntimeError)
from -e:1:in `popen'
from -e:1:in `<main>'
#<Process::Status: pid 120178 exit 1>
martind@stormy:~/download/ruby$
```

... which I think have positive value, are preserved:

```
martind@stormy:~/download/ruby$ ./ruby --disable-gems -we 'IO.popen("-") { |io| raise("a fuss") unless io; }; puts($?.inspect())'
-e:1:in 'block in <main>': a fuss (RuntimeError)
  from -e:1:in 'IO.popen'
  from -e:1:in '<main>'
#<Process::Status: pid 120135 exit 1>
martind@stormy:~/download/ruby$
```

#### #4 - 01/03/2025 02:16 AM - martin.dorey@hds.com (Martin Dorey)

Sorry, I meant to add that, if I revert io.c to how it is in git, my one-liner says:

```
martind@stormy:~/download/ruby$ ./ruby --disable-gems -we 'IO.popen("-") { |io| raise("a fuss") unless io; }; puts($?.inspect())'
#<Process::Status: pid 126700 exit 0>
martind@stormy:~/download/ruby$
```

#### #5 - 01/11/2025 07:17 AM - nagachika (Tomoyuki Chikanaga)

- Backport changed from 3.1: REQUIRED, 3.2: REQUIRED, 3.3: REQUIRED, 3.4: REQUIRED to 3.1: REQUIRED, 3.2: DONE, 3.3: REQUIRED, 3.4: REQUIRED

ruby\_3\_2 [f150d67b7d389eb88e0cd13694d3529895d55579](#) merged revision(s) [8034e9c3d001ca3dff124ab42972684eac8af2ae](#).

#### #6 - 01/15/2025 01:59 AM - k0kubun (Takashi Kokubun)

- Backport changed from 3.1: REQUIRED, 3.2: DONE, 3.3: REQUIRED, 3.4: REQUIRED to 3.1: REQUIRED, 3.2: DONE, 3.3: DONE, 3.4: REQUIRED

ruby\_3\_3 [12a0807965624a0be37dc79371a69b5d787cc8d1](#) merged revision(s) [8034e9c3d001ca3dff124ab42972684eac8af2ae](#).

#### #7 - 02/14/2025 02:01 AM - k0kubun (Takashi Kokubun)

- Backport changed from 3.1: REQUIRED, 3.2: DONE, 3.3: DONE, 3.4: REQUIRED to 3.1: REQUIRED, 3.2: DONE, 3.3: DONE, 3.4: DONE

ruby\_3\_4 [b65cea74295358265dfabc9e1f4d107b21e58e58](#) merged revision(s) [8034e9c3d001ca3dff124ab42972684eac8af2ae](#).