

## Ruby 1.8 - Bug #4132

### Socket.close attempting to close the socket twice

12/08/2010 07:57 AM - devmod (Claudio Villalobos)

<b>Status:</b>	Open
<b>Priority:</b>	Normal
<b>Assignee:</b>	
<b>Target version:</b>	
<b>ruby -v:</b>	ruby 1.8.7 (2010-01-10 patchlevel 249) [i486-linux]

---

**Description**

```
=begin
Socket.close calls shutdown twice on the same file descriptor

## socktest.rb ##
require 'socket'
include Socket::Constants
socket = Socket.new( AF_INET, SOCK_STREAM, 0 )
sockaddr = Socket.pack_sockaddr_in( 2200, 'localhost' )
socket.bind( sockaddr )
socket.listen( 5 )
puts "socket.to_i: #{socket.to_i}"
socket.close
#####

$ strace -e trace=open,close ruby1.8 socktest.rb
...
socket.to_i: 3
close(3)          = 0
close(3)          = -1 EBADF (Bad file descriptor)
```

---

```
socket.c
-
socks_s_close(sock)
...
1386  shutdown(fileno(fp->f), 2);
1387  shutdown(fileno(fp->f2), 2);
=end
```

### History

#### #1 - 12/08/2010 09:41 AM - matz (Yukihiro Matsumoto)

```
=begin
Hi,
```

In message "Re: [ruby-core:33628] [Ruby 1.8-Bug#4132][Open] Socket.close attempting to close the socket twice" on Wed, 8 Dec 2010 07:57:58 +0900, Claudio Villalobos [redmine@ruby-lang.org](mailto:redmine@ruby-lang.org) writes:

```
|Bug #4132: Socket.close attempting to close the socket twice
|http://redmine.ruby-lang.org/issues/show/4132
```

```
|
|Author: Claudio Villalobos
|Status: Open, Priority: Normal
|Category: ext
|ruby -v: ruby 1.8.7 (2010-01-10 patchlevel 249) [i486-linux]
```

```
|Socket.close calls shutdown twice on the same file descriptor
```

Ruby 1.8 creates two FILE\* streams on a socket to implement bidirectional communication. Two calls to close(2) is a side effect. Since it shouldn't do any harm, we don't consider it a bug. Tell us if it would cause any trouble (other than one useless system call, and a failure in strace log).

matz.

=end

**#2 - 12/08/2010 10:16 AM - devmod (Claudio Villalobos)**

=begin

Hi,

Since file descriptors are reused after closing, wouldn't there be an edge case where that file descriptor could have been assigned to another socket before the second close had been executed?

=end

**#3 - 12/08/2010 10:54 AM - matz (Yukihiro Matsumoto)**

=begin

Hi,

In message "Re: [ruby-core:33630] [Ruby 1.8-Bug#4132] Socket.close attempting to close the socket twice" on Wed, 8 Dec 2010 10:18:22 +0900, Claudio Villalobos [redmine@ruby-lang.org](mailto:redmine@ruby-lang.org) writes:

|

|Since file descriptors are reused after closing, wouldn't there be an edge case where that file descriptor could have been assigned to another socket before the second close had been executed?

On 1.8 (and mostly on 1.9 too), the execution is protected from multithreading, so that we don't have to worry about concurrency here.

matz.

=end

**#4 - 12/08/2010 10:58 AM - devmod (Claudio Villalobos)**

=begin

Understood. Thanks

=end

**#5 - 03/17/2011 03:35 AM - milocore (milocore milocore)**

=begin

There is should be a lot to discuss relevant to such an critical discussion, especially as <http://onlinemeetingnews.com> online meetings could be used for the purpose of significant change within the discussion.

=end