Ruby 1.8 - Bug #3106

Out-of-bounds on fd_set

04/07/2010 06:02 PM - candlerb (Brian Candler)

Status:	Open
Priority:	Normal
Assignee:	
Target version:	
ruby -v:	ruby 1.8.7 (2009-06-12 patchlevel 174) [x86_64-linux]

Description

=begin
Ruby uses select() to switch between active file descriptors, and has three fd_set's in struct thread_status_t. However, as far as I can see, no bounds checking is done to see if these are exceeded. Therefore, if you have more than FDSETSIZE files open, ruby can write beyond bounds and strange things happen.

Many Linux systems have FDSETSIZE 1024 and ulimit -n 1024 by default, and so the problem doesn't arise, but the ulimit can be raised by the sysadmin. At this point, ruby is happy to open more than 1024 files, and then crashes.

Should ruby not check for FDSETSIZE and refuse to use files whose fds are beyond this? (This limitation could be removed later if ruby ever moved to poll/epoll/kpoll)

The following code demonstrates the problem. It uses 'connect', because this causes ruby to do a non-blocking connect followed by a select.

```
require 'socket'
srv = []
cli = []
1024.times do
s = TCPServer.new('127.0.0.1',nil)
puts s.fileno
c = TCPSocket.new('127.0.0.1',s.addr[1])
puts c.fileno
srv << s
cli << c
end
```

Given 'ulimit -n 2048', for me it falls over like this:

```
...
1213
1214
1215
ert.rb:7:in initialize': Socket operation on non-socket - connect(2) (Errno::ENOTSOCK)
from ert.rb:7:innew'
from ert.rb:7
from ert.rb:4:in `times'
from ert.rb:4
```

More discussion at http://www.ruby-forum.com/topic/207462
=end