

Backport191 - Backport #3336

Memory leak in IO.select() on Windows

05/25/2010 06:26 AM - hdm (HD Moore)

Status:	Assigned
Priority:	Normal
Assignee:	yugui (Yuki Sonoda)
Description	
<pre>=begin The IO.select() method leaks memory on the Windows platform (tested mingw32). This prevents any long-running service process from working properly on Windows and is a blocker for application development. Using LeakDiag, I tracked this down to an fdset not being freed properly, but I am not sure what the proper solution is yet, To quickly reproduce this issue, run the following snippet: require 'socket' a = TCPServer.new(8888) while true IO.select([a], nil, nil, 0.01) end Watch the memory usage spike up in the task manager or process explorer. =end</pre>	

History

#1 - 05/25/2010 07:16 AM - rogerdpack (Roger Pack)

```
=begin
Was able to reproduce with
ruby 1.9.3dev (2010-05-22) [i386-mingw32]
=end
```

#2 - 05/25/2010 09:42 AM - hdm (HD Moore)

```
=begin
The patch below fixes this memory leak
```

Index: io.c

```
=====
--- io.c (revision 27934)
+++ io.c (working copy)
@@ -7297,6 +7297,7 @@
rb_f_select(int argc, VALUE *argv, VALUE obj)
{
VALUE timeout;

• VALUE r; struct select_args args; struct timeval timerec; int i; @@ -7316,8 +7317,12 @@ #ifdef HAVE_RB_FD_INIT return
rb_ensure(select_call, (VALUE)&args, select_end, (VALUE)&args); #else
• return select_internal(args.read, args.write, args.except,
• args.timeout, args.fdsets);
• r = select_internal(args.read, args.write, args.except, args.timeout, args.fdsets); +
• for (i = 0; i < numberof(args.fdsets); ++i)
• rb_fd_term(&args.fdsets[i]); +
• return(r); #endif
}
=end
```

#3 - 05/25/2010 10:32 AM - usa (Usaku NAKAMURA)

- Status changed from Open to Assigned
- Assignee set to nobu (Nobuyoshi Nakada)

```
=begin
mingw only (VC++ version doesn't have this problem)
```

Nobu, I think that you should define HAVE_RB_FD_INIT in configure for mingw.
=end

#4 - 05/25/2010 12:15 PM - hdm (HD Moore)

=begin
Strangely, rb_fd_init() is being used before the #ifdef block; it does not make sense for this function to call the select_internal() unless the rb_fd_init()'s are similarly ifdef'd (any platform without HAVE_RB_FD_INIT will experience a memory leak).
=end

#5 - 05/25/2010 12:48 PM - usa (Usaku NAKAMURA)

=begin
Hello,

In message "[ruby-core:30407] [Bug #3336] Memory leak in IO.select() on Windows"
on May.25.2010 12:15:33, redmine@ruby-lang.org wrote:

Strangely, rb_fd_init() is being used before the #ifdef block; it does not make sense for this function to call the select_internal() unless the rb_fd_init()'s are similarly ifdef'd (any platform without HAVE_RB_FD_INIT will experience a memory leak).

I think that HAVE_RB_FD_INIT is not a good name.
It's used as the indicator of a necessity to call rb_fd_term().

Regards,
--
U.Nakamura usa@garbagecollect.jp

=end

#6 - 05/25/2010 01:27 PM - hdm (HD Moore)

=begin
Thank you for the explanation - that makes much more sense :)
=end

#7 - 05/26/2010 10:37 AM - nobu (Nobuyoshi Nakada)

- Status changed from Assigned to Closed
- % Done changed from 0 to 100

=begin
This issue was solved with changeset r28016.
HD, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

=end

#8 - 05/26/2010 11:40 AM - nobu (Nobuyoshi Nakada)

- Category set to core
- Status changed from Closed to Assigned
- Assignee changed from nobu (Nobuyoshi Nakada) to yugui (Yuki Sonoda)

=begin

=end

#9 - 05/27/2010 01:16 AM - rogerdpack (Roger Pack)

=begin
Seems fixed now in mingw. Thanks!
=end