Ruby master - Feature #5097

Supported platforms of Ruby 1.9.3

07/26/2011 11:52 AM - naruse (Yui NARUSE)

Status:	Closed
Priority:	Normal
Assignee:	naruse (Yui NARUSE)
Target version:	1.9.3

Description

Let's decide the supported platforms.

== Background

http://redmine.ruby-lang.org/projects/ruby-19/wiki/SupportedPlatforms

== Process

If you want to support a platform, please declare.
But when a platform dependent bug is reported, it will be assigned to you.

== Current Maintainer

mswin32, mswin64 (Microsoft Windows):
NAKAMURA Usaku (usa)
mingw32 (Minimalist GNU for Windows):
Nobuyoshi Nakada (nobu)
IA-64 (Debian GNU/Linux):
TAKANO Mitsuhiro (takano32)
Symbian OS:
Alexander Zavorine (azov)
AIX:
Yutaka Kanemoto (kanemoto)
FreeBSD:
Akinori MUSHA (knu)
Solaris:
Naohisa Goto
RHEL, CentOS
KOSAKI Motohiro

Platforms which doesn't have a maintainer are following:

- Debian
- Ubuntu
- Mac OS X (LLVM related issues)
- cygwin (don't work)
- NetBSD (works)
- OpenBSD (it may not work)
- DragonFlyBSD (don't work)

Related issues:

Related to Ruby master - Bug #5094: Supported platforms of Ruby 1.9.3	**Closed**	**07/25/2011**

Associated revisions

Revision d7138d72 - 07/27/2011 03:30 PM - naruse (Yui NARUSE)

- test/fileutils/test_fileutils.rb: add OpenBSD case.
  patched by Jeremy Evans [ruby-core:38530] see #5097
- test/ruby/test_process.rb: ditto.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32707 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 32707 - 07/27/2011 03:30 PM - naruse (Yui NARUSE)

- test/fileutils/test_fileutils.rb: add OpenBSD case.
  patched by Jeremy Evans [ruby-core:38530] see #5097

- test/ruby/test_process.rb: ditto.

Revision 32707 - 07/27/2011 03:30 PM - naruse (Yui NARUSE)

- test/fileutils/test_fileutils.rb: add OpenBSD case.
  patched by Jeremy Evans [ruby-core:38530] see #5097

- test/ruby/test_process.rb: ditto.

Revision 32707 - 07/27/2011 03:30 PM - naruse (Yui NARUSE)

- test/fileutils/test_fileutils.rb: add OpenBSD case.
  patched by Jeremy Evans [ruby-core:38530] see #5097

- test/ruby/test_process.rb: ditto.

Revision 32707 - 07/27/2011 03:30 PM - naruse (Yui NARUSE)

- test/fileutils/test_fileutils.rb: add OpenBSD case.
  patched by Jeremy Evans [ruby-core:38530] see #5097

- test/ruby/test_process.rb: ditto.

Revision 32707 - 07/27/2011 03:30 PM - naruse (Yui NARUSE)

- test/fileutils/test_fileutils.rb: add OpenBSD case.
  patched by Jeremy Evans [ruby-core:38530] see #5097

- test/ruby/test_process.rb: ditto.

Revision 32707 - 07/27/2011 03:30 PM - naruse (Yui NARUSE)

- test/fileutils/test_fileutils.rb: add OpenBSD case.
  patched by Jeremy Evans [ruby-core:38530] see #5097

- test/ruby/test_process.rb: ditto.

Revision f39ed9dc - 07/27/2011 03:33 PM - naruse (Yui NARUSE)

merge revision(s) 32707:

```
    * test/fileutils/test_fileutils.rb: add OpenBSD case.
      patched by Jeremy Evans [ruby-core:38530] see #5097

    * test/ruby/test_process.rb: ditto.
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_9_3@32709 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

History

#1 - 07/26/2011 12:53 PM - normalperson (Eric Wong)

Yui NARUSE naruse@airemix.jp wrote:

> If you want to support a platform, please declare.
> But when a platform dependent bug is reported, it will be assigned to you.
>
> RHEL, CentOS
> KOSAKI Motohiro

5.x or 6.x? I still have to deal with CentOS 5.x machines sometimes,
so I can poke my head in if required.

> Platforms which doesn't have a maintainer are following:
>
> - Debian

I'll help with Debian Squeeze (x86/x86_64) and testing (Wheezy,
x86_64-only); but I can't commit to an official role.

--
Eric Wong

#2 - 07/26/2011 01:08 PM - jeremyevans0 (Jeremy Evans)

I volunteer to officially support Ruby on OpenBSD. I'm an OpenBSD developer and already maintain the OpenBSD port for Ruby (both 1.8.7 and 1.9.2) as well as the OpenBSD ports for Rubinius and JRuby.

#3 - 07/26/2011 03:53 PM - wyhaines (Kirk Haines)

On Mon, Jul 25, 2011 at 8:52 PM, Yui NARUSE naruse@airemix.jp wrote:

> - Ubuntu

I can volunteer to help with Ubuntu, if you need it.

Kirk Haines

#4 - 07/26/2011 07:53 PM - kosaki (Motohiro KOSAKI)

2011/7/26 Eric Wong normalperson@yhbt.net:

> Yui NARUSE naruse@airemix.jp wrote:
>
> > If you want to support a platform, please declare.
> > But when a platform dependent bug is reported, it will be assigned to you.
> >
> > RHEL, CentOS
> > Â KOSAKI Motohiro
>
> 5.x or 6.x? Â I still have to deal with CentOS 5.x machines sometimes,
> so I can poke my head in if required.

I only plan to maintain 6.x. So, your overture is really helpful to me.

Thanks!

#5 - 07/26/2011 07:59 PM - kosaki (Motohiro KOSAKI)

> I volunteer to officially support Ruby on OpenBSD. Â I'm an OpenBSD developer and already maintain the OpenBSD port for Ruby (both 1.8.7 and 1.9.2) as well as the OpenBSD ports for Rubinius and JRuby.

Great!

May I request one more thing? We who lead to release 1.9.3 developers
are worry about current 1.9.3 branch works on OpenBSD. We have no
OpenBSD test environment. Can you please try run `make check OPTS

#6 - 07/26/2011 10:19 PM - steveklabnik (Steve Klabnik)

I would like to help for OSX > 10.6, and maybe 10.5 if I can get a machine. I'm not exactly sure what 'maintainer's responsibility exactly entails, but I'd be happy to help organize, make a best effort at patching bugs, and provide builds relating to that platform.

#7 - 07/27/2011 12:29 AM - lucas (Lucas Nussbaum)

For Linux distributions, it would make sense to have a list both of distributions, and of architectures. There are problems that are distribution-specific, and problems that are architecture-specific.

I can help with Debian (and Ubuntu, since Ubuntu just imports the Debian package), since I'm one of the Ruby maintainers in Debian.

Regarding (Debian) architectures, the status is quite bad[1]:
armel: failing because of a GCC bug, see http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=634260
ia64: fails with:
compiling cont.c
cont.c: In function 'cont_save_thread':
cont.c:386: error: expected ';' before 'cont'
make[1]: *** [cont.o] Error 1

kfreebsd-amd64 and kfreebsd-i386: tests hang

sparc:
sample/test.rb:pack /build/buildd-ruby1.9.1_1.9.2.180+svn32566-1-sparc-Aby1Vq/ruby1.9.1-1.9.2.180+svn32566/sample/tes
t.rb:2002: [BUG] Bus Error

[1] https://buildd.debian.org/status/package.php?p=ruby1.9.1&suite=experimental

#8 - 07/27/2011 12:53 AM - luislavena (Luis Lavena)

On Mon, Jul 25, 2011 at 11:52 PM, Yui NARUSE naruse@airemix.jp wrote:

> If you want to support a platform, please declare.
> But when a platform dependent bug is reported, it will be assigned to you.
> == Current Maintainer
>
> mswin32, mswin64 (Microsoft Windows):
> NAKAMURA Usaku (usa)
> mingw32 (Minimalist GNU for Windows):
> Nobuyoshi Nakada (nobu)
> I can contribute with all available resources to test Ruby against
> mingw32 and mingw-w64 on both 32bits and upcoming 64bits work.

We continuously run builds against trunk and specific branchs for
regressions so ruby_1_9_3 builds cleanly.

In relation to tests, we never had a full zero errors/failures build,
which makes hard to verify these.

We at RubyInstaller project do our best and will continue to support it.

--
Luis Lavena

AREA 17

Perfection in design is achieved not when there is nothing more to add,
but rather when there is nothing more to take away.
Antoine de Saint-Exupéry

#9 - 07/27/2011 03:44 AM - jeremyevans0 (Jeremy Evans)

- File ruby193.log added

- File ruby193.diff added

Attached is the make check output on OpenBSD amd64. I'm also attaching a diff with the patches I used:

- bootstraptest/test_thread.rb: Skip 2 tests. The first one appears to hang, the second crashes with a sigaltstack error.
- configure: Use -pthread instead of -lpthread when linking. From OpenBSD man pages: "On OpenBSD, the -pthread option should be used to link threaded code, isolating the program from operating system details."

Use OpenBSD shared library naming convention (libruby19.so.1.0, as 1.9.2 used libruby19.so.0.0).

Don't include OpenBSD version number in $arch. OpenBSD version numbers do not have any relationship to API/ABI compatibility.

- ext/socket/lib/socket.rb: Don't check ipv6_recvpktinfo. This fixes a failure in one of the socket tests:

  test_udp_server(TestSocket):
  RuntimeError: no response from #

- test/fileutils/test_fileutils.rb: Add openbsd to the /freebsd|netbsd/ regexp. Fixes:

  test_chmod_symbol_mode(TestFileUtils):
  Errno::EFTYPE: Inappropriate file type or format - tmp/a

- test/ruby/test_process.rb: Add openbsd to the /freebsd/ regexp. Fixes:

  test_wait_and_sigchild(TestProcess) [/usr/obj/ports/ruby-1.9.3-p0/ruby-1.9.3-p0/test/ruby/test_process.rb:1205]:
  [ruby-core:19744].
  <[true]> expected but was
  <[true, true]>.

Skip test_rlimit_nofile as it appears to hang.

Even with the attached patch, there are 5 failures. 3 failures are weird linking errors in the rake tests I don't understand ("ruby lib version (1.9.3) doesn't match executable version (1.9.4)"). 2 failures appear to be caused by using GNU make syntax in the makefiles. On 1.9.2, BSD make was able to work with the makefiles, but starting on 1.9.3, it appears GNU make is required. Either makefiles should use not use GNU extensions or ruby should use gmake instead of make on BSD systems.

I used a modified version of the OpenBSD port for 1.9.2 to build 1.9.3, using a tarball created from b6fd481c of the git mirror (svn revision 32655). The following configure environment was used:

LIBruby19_VERSION=1.0 PREFIX="/usr/local" CPPFLAGS="-DOPENSSL_NO_STATIC_ENGINE -I/usr/local/include" LDFLAGS="-L/usr/local/lib" CONFIG_SITE='/usr/ports/infrastructure/db/config.site' MKDIR_P='mkdir -p' MAKE=gmake
PATH=/usr/obj/ports/ruby-1.9.3-p0/bin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/usr/local/bin:/usr/X11R6/bin

and the following configure arguments:

--program-suffix=19 --with-soname=ruby19 --enable-pthread --enable-ipv6 --disable-option-checking --with-tcl-include=/usr/local/include/tcl8.5 --with-tk-include=/usr/local/include/tk8.5 --with-X11-dir=/usr/X11R6 --enable-shared --prefix='/usr/local' --sysconfdir='/etc' --mandir='/usr/local/man' --infodir='/usr/local/info' --disable-silent-rules

#10 - 07/28/2011 12:32 AM - naruse (Yui NARUSE)

Jeremy Evans wrote:

> Attached is the make check output on OpenBSD amd64. I'm also attaching a diff with the patches I used:
>
> - bootstraptest/test_thread.rb: Skip 2 tests. The first one appears to hang, the second crashes with a sigaltstack error.

signalstack error seems because of the lack of OpenBSD specific setting.
see thread_pthread.c:506 and edit it.

> - configure: Use -pthread instead of -lpthread when linking. From OpenBSD man pages: "On OpenBSD, the -pthread option should be used to link threaded code, isolating the program from operating system details."

A patch should be for portable and for configure.in.

> Use OpenBSD shared library naming convention (libruby19.so.1.0, as 1.9.2 used libruby19.so.0.0).

What is LIBruby19_VERSION?

> Don't include OpenBSD version number in $arch. OpenBSD version numbers do not have any relationship to API/ABI compatibility.

On *BSD/Mac OS X includes version to arch.
It seems independent from API/ABI compatibility.

- ext/socket/lib/socket.rb: Don't check ipv6_recvpktinfo. This fixes a failure in one of the socket tests:

Why this fixes the problem?
We can't merge this without understanding.

- test/fileutils/test_fileutils.rb: Add openbsd to the /freebsd|netbsd/ regexp. Fixes:

- test/ruby/test_process.rb: Add openbsd to the /freebsd/ regexp. Fixes:

  test_wait_and_sigchild(TestProcess) [/usr/obj/ports/ruby-1.9.3-p0/ruby-1.9.3-p0/test/ruby/test_process.rb:1205]:
  [ruby-core:19744].
  <[true]> expected but was
  <[true, true]>.

> Skip test_rlimit_nofile as it appears to hang.

Merged in r32707.

> Even with the attached patch, there are 5 failures. 3 failures are weird linking errors in the rake tests I don't understand ("ruby lib version (1.9.3) doesn't match executable version (1.9.4)").

Hmm, I don't know about it.

> 2 failures appear to be caused by using GNU make syntax in the makefiles. On 1.9.2, BSD make was able to work with the makefiles, but starting on 1.9.3, it appears GNU make is required. Either makefiles should use not use GNU extensions or ruby should use gmake instead of make on BSD systems.

It's strange because on FreeBSD and NetBSD, BSD make works correctly.

#11 - 07/28/2011 09:50 AM - jeremyevans0 (Jeremy Evans)

- File ruby193.log added

- File ruby193.diff added

> Jeremy Evans wrote:
>
> > Attached is the make check output on OpenBSD amd64. I'm also attaching a diff with the patches I used:
> >
> > - bootstraptest/test_thread.rb: Skip 2 tests. The first one appears to hang, the second crashes with a sigaltstack error.
>
> signalstack error seems because of the lack of OpenBSD specific setting.
> see thread_pthread.c:506 and edit it.

I don't think the problem is there, as reading the documentation the call there (pthread_stackseg_np(pthread_self(), &stk)) appears to be correct. The error I get in the 2nd bootstrap test is "rb_register_sigaltstack. malloc error", which comes from signal.c:

```
439     newSS.ss_sp = th->altstack = malloc(ALT_STACK_SIZE);
440     if (newSS.ss_sp == NULL)
441         /* should handle error */
442         rb_bug("rb_register_sigaltstack. malloc error\n");
```

If you change the rb_bug call to a return, the 2nd bootstrap test passes. I'm not sure if that's a valid fix, as it probably affects signal handling for the thread. I first tried to change it to raise NoMemError, but that causes a segfault. Like the comment there states, the error should be handled, but I'm not sure how best to do that. I don't think this is an error caused by OpenBSD, just one that only manifests itself on OpenBSD.

The 1st bootstrap error appears to be the threads not exiting when the process exits, causing the threads to hang. It can be reproduced with a single thread (ruby19 -e "Thread.new{loop{Thread.pass}}"). The hang doesn't happen if you use while instead of loop (ruby19 -e 'Thread.new{Thread.pass while true}'). Any ideas?

- configure: Use -pthread instead of -lpthread when linking. From OpenBSD man pages: "On OpenBSD, the -pthread option should be used to link threaded code, isolating the program from operating system details."

A patch should be for portable and for configure.in.

I think I've got a working patch, it's attached.

Use OpenBSD shared library naming convention (libruby19.so.1.0, as 1.9.2 used libruby19.so.0.0).

What is LIBruby19_VERSION?

You can ignore that. On OpenBSD, the ports system sets the shared library numbers, because most software doesn't handle the library versioning correctly (bumping the minor for new functions, bumping the major for removed functions or changes in functions). But that's not something you are going to want pulled upstream.

Don't include OpenBSD version number in $arch. OpenBSD version numbers do not have any relationship to API/ABI compatibility.

On *BSD/Mac OS X includes version to arch.
It seems independent from API/ABI compatibility.

Including the version number means that every time the version number changes, all dependent ports that use sitearchdir need to be modified slightly. What's the rational behind including the operating system version if not for API/ABI compatibility?

- ext/socket/lib/socket.rb: Don't check ipv6_recvpktinfo. This fixes a failure in one of the socket tests:

Why this fixes the problem?
We can't merge this without understanding.

I think this is a general bug in the code. If ipv6_recvpktinfo is true (which it is on OpenBSD since OpenBSD supports IPV6_PKTINFO), the Addrinfo for IPV6 address "::" is not added as a socket to use. The code looks like this:

```
516     elsif ai.ipv6? && ai.ip_address == "::" && !ipv6_recvpktinfo
517       local_addrs.each {|a|
518         next if !a.ipv6?
519         ip_list << Addrinfo.new(a.to_sockaddr, :INET6, :DGRAM, 0);
520       }

527     sockets = ip_sockets_port0(ip_list, false)

544   pktinfo_sockets = {}
545   sockets.each {|s|
546     ai = s.local_address
547     if ipv6_recvpktinfo && ai.ipv6? && ai.ip_address == "::"
548       s.setsockopt(:IPV6, ipv6_recvpktinfo, 1)
549       pktinfo_sockets[s] = true
550     end
551   }
```

The pktinfo_sockets hash here is never used. And even if it was, because you are not adding the '::' address to ip_addrs if ipv6_recvpktinfo, no socket is being created for it, so you will never hit lines 548-549. The attached patch removes the "&& !ipv6_recvpktinfo" from 516 and the pktinfo_sockets hash handling.

Even with the attached patch, there are 5 failures. 3 failures are weird linking errors in the rake tests I don't understand ("ruby lib version (1.9.3) doesn't match executable version (1.9.4)").

Hmm, I don't know about it.

This appeared to be caused by my use of a non-pristine checkout. I did a pristine checkout and they went away.

2 failures appear to be caused by using GNU make syntax in the makefiles. On 1.9.2, BSD make was able to work with the makefiles, but starting on 1.9.3, it appears GNU make is required. Either makefiles should use not use GNU extensions or ruby should use gmake instead

of make on BSD systems.

It's strange because on FreeBSD and NetBSD, BSD make works correctly.

There is the same bug in about 4 files that causes this, the submitted patch includes the fixes and makes everything compile with OpenBSD make. This is because the old=new modifier must be the last modifier in a BSD make variable substituion, and both those files contained a trailing :, which I'm guessing indicates another modifier. I guess OpenBSD make is more strict than the other BSD makes, but the last modifier behavior is specified in the FreeBSD make man page as well.

I ran make check and found the following errors:

4 errors in test/unit's test::parallel tests caused by bugs, a patch for these is included.

3 errors in rake's tests that are caused by assuming that ruby is installed as "ruby". If you configure with --program-suffix, these fail. A patch for these is included.

The rlimit hang is caused because setting the RLIMIT_NOFILE to 0 causes ruby to hang on exit in OpenBSD. Setting it to 1 does not cause it to hang. I added a patch that modifies the test code to use 1 instead of 0, but if you'd like a patch to process.c to use 1 if 0 is given on OpenBSD, I can do that.

With the attached patch, there are no errors. There is a error in one of the timeout tests in the log, but that's spurious:

$ ruby19 -v test/test_timeout.rb
ruby 1.9.3dev (2011-07-28) [x86_64-openbsd5.0]
Run options:

Running tests:

...

Finished tests in 1.422562s, 2.1089 tests/s, 2.8118 assertions/s.

3 tests, 4 assertions, 0 failures, 0 errors, 0 skips

#12 - 07/28/2011 12:53 PM - akr (Akira Tanaka)

2011/7/28 Jeremy Evans merch-redmine@jeremyevans.net:

- ext/socket/lib/socket.rb: Don't check ipv6_recvpktinfo. This fixes a failure in one of the socket tests:

Why this fixes the problem?
We can't merge this without understanding.

I think this is a general bug in the code. If ipv6_recvpktinfo is true (which it is on OpenBSD since OpenBSD supports IPV6_PKTINFO), the Addrinfo for IPV6 address "::" is not added as a socket to use. The code looks like this:

```
516     elsif ai.ipv6? && ai.ip_address == "::" && !ipv6_recvpktinfo
517       local_addrs.each {|a|
518         next if !a.ipv6?
519         ip_list << Addrinfo.new(a.to_sockaddr, :INET6, :DGRAM, 0);
520       }

527     sockets = ip_sockets_port0(ip_list, false)

544   pktinfo_sockets = {}
545   sockets.each {|s|
546     ai = s.local_address
547     if ipv6_recvpktinfo && ai.ipv6? && ai.ip_address == "::"
548       s.setsockopt(:IPV6, ipv6_recvpktinfo, 1)
549       pktinfo_sockets[s] = true
550     end
551   }
```

The pktinfo_sockets hash here is never used. And even if it was, because you are not adding the '::' address to ip_addrs if ipv6_recvpktinfo, no socket is being created for it, so you will never hit lines 548-549. The attached patch removes the "&& !ipv6_recvpktinfo" from 516 and the pktinfo_sockets hash handling.

Thank you for notifying pktinfo_sockets is not used.

However I don't understand why ip_list doesn't contain '::' address on OpenBSD.

```
516       elsif ai.ipv6? && ai.ip_address == "::" && !ipv6_recvpktinfo
517         local_addrs.each {|a|
518           next if !a.ipv6?
519           ip_list << Addrinfo.new(a.to_sockaddr, :INET6, :DGRAM, 0);
520         }
521       else
522         ip_list << ai
523       end
```

The line 522 should add '::' address to ip_list on platforms which
supports IPV6_PKTINFO.

Your patch, removing "&& !ipv6_recvpktinfo", means we don't use IPV6_PKTINFO.
It is not my intent. (and it doesn't supports dynamic IP address change.)

#13 - 07/28/2011 03:02 PM - jeremyevans0 (Jeremy Evans)

=begin

> The pktinfo_sockets hash here is never used. And even if it was, because you are not adding the '::' address to ip_addrs if ipv6_recvpktinfo, no socket is being created for it, so you will +never hit lines 548-549. The attached patch removes the "&& !ipv6_recvpktinfo" from 516 and the pktinfo_sockets hash handling.

Thank you for notifying pktinfo_sockets is not used.

However I don't understand why ip_list doesn't contain '::' address on OpenBSD.

```
516       elsif ai.ipv6? && ai.ip_address == "::" && !ipv6_recvpktinfo
517         local_addrs.each {|a|
518           next if !a.ipv6?
519           ip_list << Addrinfo.new(a.to_sockaddr, :INET6, :DGRAM, 0);
520         }
521       else
522         ip_list << ai
523       end
```

The line 522 should add '::' address to ip_list on platforms which supports IPV6_PKTINFO.

The issue is that Socket.ip_address_list and Addrinfo.foreach return different things.

```
$ ruby19 -rsocket -e 'p Socket.ip_address_list'
[#<Addrinfo: ::1>, #<Addrinfo: fe80::1%lo0>, #<Addrinfo: 127.0.0.1>, #<Addrinfo: 192.168.1.4>, #<Addrinfo: fe8
0::92fb:a6ff:feed:afa1%re0>]
$ ruby19 -rsocket -e 'Addrinfo.foreach(nil, 0, nil, :DGRAM, nil, Socket::AI_PASSIVE) {|ai| p ai}'
#<Addrinfo: :: UDP>
#<Addrinfo: 0.0.0.0 UDP>
```

> Your patch, removing "&& !ipv6_recvpktinfo", means we don't use IPV6_PKTINFO. It is not my intent. (and it doesn't supports dynamic IP address change.)

I can see where it wouldn't support dynamic IP address change. But it also doesn't listen on all local addresses (# in this case), and that fails one of the tests. So either the test needs to be changed to not require that behavior or the code needs to be change to listen on all local addresses in addition to listening on '::' in the ipv6_recvpktinfo case.
=end

#14 - 07/28/2011 04:53 PM - drbrain (Eric Hodel)

On Jul 27, 2011, at 17:50, Jeremy Evans merch-redmine@jeremyevans.net wrote:

> Even with the attached patch, there are 5 failures. 3 failures are weird linking errors in the rake tests I don't understand ("ruby lib version (1.9.3) doesn't match executable version (1.9.4)").

Hmm, I don't know about it.

> This appeared to be caused by my use of a non-pristine checkout. I did a pristine checkout and they went away.

This may occur for certain tests if ./ruby is not the same version as the ruby in $PATH when a test invokes ruby in a forked process. It's a test bug I

can reproduce but haven't filed a ticket for. You can ignore it.

#15 - 07/28/2011 05:52 PM - znz (Kazuhiro NISHIYAMA)

I think
/#{RUBY} -e/ in ruby193.diff
should be
/#{Regexp.quote(RUBY)} -e/

#16 - 07/28/2011 10:23 PM - nobu (Nobuyoshi Nakada)

Hi,

At Wed, 27 Jul 2011 03:44:48 +0900,
Jeremy Evans wrote in [ruby-core:38530]:

> Attached is the make check output on OpenBSD amd64. I'm also attaching a diff with the patches I used:

What is the reason to change ECHO1 in common.mk and so on?
Does OpenBSD make fail to substitute with a text with a colon?
Anyway, It doesn't seem to work when V=1.

--
Nobu Nakada

#17 - 07/29/2011 12:23 AM - kosaki (Motohiro KOSAKI)

- bootstraptest/test_thread.rb: Skip 2 tests. The first one appears to hang, the second crashes with a sigaltstack error.

> signalstack error seems because of the lack of OpenBSD specific setting.
> see thread_pthread.c:506 and edit it.

I don't think the problem is there, as reading the documentation the call there (pthread_stackseg_np(pthread_self(), &stk)) appears to be correct.
Â The error I get in the 2nd bootstrap test is "rb_register_sigaltstack. malloc error", which comes from signal.c:

Â 439 Â Â newSS.ss_sp

#18 - 07/29/2011 12:36 AM - jeremyevans0 (Jeremy Evans)

Nobuyoshi Nakada wrote:

> Hi,
>
> At Wed, 27 Jul 2011 03:44:48 +0900,
> Jeremy Evans wrote in [ruby-core:38530]:
>
> > Attached is the make check output on OpenBSD amd64. I'm also attaching a diff with the patches I used:
>
> What is the reason to change ECHO1 in common.mk and so on?
> Does OpenBSD make fail to substitute with a text with a colon?
> Anyway, It doesn't seem to work when V=1.

It's invalid syntax in OpenBSD make, probably because the colon indicates another modifier and the old=new modifier must be the last modifier in a make variable substitution:

```
:old_string=new_string
         This is the AT&T System V UNIX style variable substitution.  It
         must be the last modifier specified.
```

#19 - 07/29/2011 12:53 AM - kosaki (Motohiro KOSAKI)

offtopic,

> ia64: fails with:
> compiling cont.c
> cont.c: In function 'cont_save_thread':

> cont.c:386: error: expected ';' before 'cont'
> make[1]: *** [cont.o] Error 1

This problem has been fixed by r32582.

Thanks.

#20 - 07/29/2011 03:46 AM - jeremyevans0 (Jeremy Evans)

Motohiro KOSAKI wrote:

> I wonder why OpenBSD can't allocate SIGSTKSZ size. Usually it's very small. Can you please tell us openbsd has which value of SIGSTKSZ and MINSIGSTKSZ?
> Also, can you please try following stack size reducing patch?

OpenBSD amd64
MINSIGSTKSZ = 8192
SIGSTKSZ = 40960 (8192 + 32768)

So your patch wouldn't work, as 4*1024 < 8192. I tried with MINSIGSTKSZ*2 and it still crashed, but it passed with just MINSIGSTKSZ.

#21 - 07/29/2011 05:17 AM - jeremyevans0 (Jeremy Evans)

Jeremy Evans wrote:

> Motohiro KOSAKI wrote:
>
> > I wonder why OpenBSD can't allocate SIGSTKSZ size. Usually it's very small. Can you please tell us openbsd has which value of SIGSTKSZ and MINSIGSTKSZ?
> > Also, can you please try following stack size reducing patch?
>
> OpenBSD amd64
> MINSIGSTKSZ = 8192
> SIGSTKSZ = 40960 (8192 + 32768)
>
> So your patch wouldn't work, as 4*1024 < 8192. I tried with MINSIGSTKSZ*2 and it still crashed, but it passed with just MINSIGSTKSZ.

Spoke to soon. It passed with just MINSIGSTKSZ if run in isolation, but not with the other tests:

make btest OPTS="--sets=thread" => works
make btest => works
make check OPTS="--sets=thread" => crashes
make check => crashes

This isn't that surprising as malloc can fail anytime there isn't enough memory (even if you are only requesting 1 byte). I think the only way to fix this is to either ignore the malloc error (risky) or safely raise a NoMethodError (safer, but I'm not sure how to do it).

#22 - 07/29/2011 05:18 AM - jeremyevans0 (Jeremy Evans)

Nobuyoshi Nakada wrote:

> Hi,
>
> At Wed, 27 Jul 2011 03:44:48 +0900,
> Jeremy Evans wrote in [ruby-core:38530]:
>
> > Attached is the make check output on OpenBSD amd64. I'm also attaching a diff with the patches I used:
>
> What is the reason to change ECHO1 in common.mk and so on?
> Does OpenBSD make fail to substitute with a text with a colon?
> Anyway, It doesn't seem to work when V=1.

I spoke with the OpenBSD make maintainer and he said it's a bug in OpenBSD make. So I guess there's no reason to change those files. Sorry for the false positive.

#23 - 07/29/2011 06:03 AM - normalperson (Eric Wong)

- File 0001-allocate-th-altstack-early-run-GC-on-allocation-fail.patch added

Jeremy: does my patch to allocate th->altstack in a place where xmalloc() is
possible help?

Since altstack is always initialized if Ruby is compiled to use it, I see
no reason to delay the allocation.

#24 - 07/29/2011 06:36 AM - jeremyevans0 (Jeremy Evans)

Eric Wong wrote:

> Jeremy: does my patch to allocate th->altstack in a place where xmalloc() is
> possible help?
>
> Since altstack is always initialized if Ruby is compiled to use it, I see
> no reason to delay the allocation.

Thanks Eric, that patch fixes the problem, and should be a better solution than the current code or my previous workaround.

#25 - 07/29/2011 01:41 PM - naruse (Yui NARUSE)

Jeremy Evans wrote:

> Nobuyoshi Nakada wrote:
>
> > At Wed, 27 Jul 2011 03:44:48 +0900,
> > Jeremy Evans wrote in [ruby-core:38530]:
> >
> > > Attached is the make check output on OpenBSD amd64. I'm also attaching a diff with the patches I used:
> >
> > What is the reason to change ECHO1 in common.mk and so on?
> > Does OpenBSD make fail to substitute with a text with a colon?
> > Anyway, It doesn't seem to work when V=1.
>
> I spoke with the OpenBSD make maintainer and he said it's a bug in OpenBSD make. So I guess there's no reason to change those files. Sorry for the false positive.

: is a null command of sh, so an workaround can be replace @: with @true.
But unfortunately it can't be accepted because Windows don't have true.
So please replace it when you port it until OpenBSD make is fixed.

#26 - 07/30/2011 12:53 AM - akr (Akira Tanaka)

2011/7/28 Jeremy Evans merch-redmine@jeremyevans.net:

> The issue is that Socket.ip_address_list and Addrinfo.foreach return different things.
>
> $ ruby19 -rsocket -e 'p Socket.ip_address_list'
> [#, #, #, #, #]
> $ ruby19 -rsocket -e 'Addrinfo.foreach(nil, 0, nil, :DGRAM, nil, Socket::AI_PASSIVE) {|ai| p ai}'
> #
> #

It is not a problem. GNU/Linux also behaves as so.

GNU/Linux% ./ruby -rsocket -e 'p Socket.ip_address_list'
[#, #, #, #]
GNU/Linux% ./ruby -rsocket -e 'Addrinfo.foreach(nil, 0, nil, :DGRAM,
nil, Socket::AI_PASSIVE) {|ai| p ai}'
#
#

> I can see where it wouldn't support dynamic IP address change. But it also doesn't listen on all local addresses (# in this case), and that fails one of the tests. So either the test needs to be changed to not require that behavior or the code needs to be change to listen on all local addresses in addition to listening on '::' in the ipv6_recvpktinfo case.

If IPV6_PKTINFO is usable, we don't need to listen all addresses
because we can detect the destination address of a incoming packet
using pktinfo.
(The address is used later to reply from the address.)

My intent is follows.

- If IPV6_PKTINFO is usable, listen only "::" (for IPv6). The destination address is detected by pktinfo. This way can handle dynamic IP change.
- If IPV6_PKTINFO is not usable, listen all local IPv6 addresses. The destination address is detected by the socket. This way doesn't support dynamic IP change, though.

I can't understand why OpenBSD fails the test.
--
Tanaka Akira

#27 - 07/30/2011 03:23 AM - jeremyevans0 (Jeremy Evans)

- File ruby193.diff added

> I can't understand why OpenBSD fails the test.

It looks like the socket failure on OpenBSD is because in Socket.udp_server_recv, you are adding the pktinfo in the sendmsg call, and OpenBSD appears not to like that in all cases, specifically when you are sending from a link-local IPv6 address to the same link-local IPv6 address on a different port. It causes the client to not receive the reply, which makes the test fail since the response is not received within 10 seconds. I think adding the pktinfo is unnecessary in the sendmsg call (the server needs pktinfo to know the address of the client, but the client must already know the address of the server in order to communicate), so I removed it. This causes the test to pass, and still should work with dynamic IPv6 address changes.

I've attached a smaller updated patch with four fixes:

1) Use "Thread.new{Thread.pass while true}" instead of "Thread.new{loop{Thread.pass}}" in the thread bootstrap tests, as otherwise they hang on OpenBSD. This works around the issue instead of fixing it, but it's not like "Thread.new{loop{Thread.pass}}" will be used in regular code.

2) Remove the unused pkinfo_sockets hash from Socket.udp_server_sockets.

3) Don't add the pktinfo to the sendmsg call in Socket.udp_server_recv.

4) Fix a couple of bugs in lib/test/unit.rb that assume @workers is an array when it can be nil.

The other patches I'm using are:

1) Eric Wong's thread patch

2) The patches for the rake issue that a separate ticket was created for (5114), with the Regexp.quote changes recommended by znz.

With these patches, make check has no failures on errors on OpenBSD.

#28 - 07/30/2011 03:36 AM - jeremyevans0 (Jeremy Evans)

I forgot to mention that r32730, r32731, r32734, and r32738 should be merged into the 1.9.3 branch as they are currently only on trunk. Maybe r32731, r32734, and r32738 should be combined into a single patch when merging into 1.9.3.

#29 - 07/30/2011 07:23 AM - akr (Akira Tanaka)

2011/7/30 Jeremy Evans merch-redmine@jeremyevans.net:

> It looks like the socket failure on OpenBSD is because in Socket.udp_server_recv, you are adding the pktinfo in the sendmsg call, and OpenBSD appears not to like that in all cases, specifically when you are sending from a link-local IPv6 address to the same link-local IPv6 address on a different port. It causes the client to not receive the reply, which makes the test fail since the response is not received within 10 seconds. I think adding the pktinfo is unnecessary in the sendmsg call (the server needs pktinfo to know the address of the client, but the client must already know the address of the server in order to communicate), so I removed it. This causes the test to pass, and still should work with dynamic IPv6 address changes.

The pktinfo for sendmsg is required to specify the source address of the
sending packet.

UDP server should reply from the address which client sent to.

If the server machine is multi-homed, the UDP server may reply from
different address without pktinfo for sendmsng.
I.e. if the server has addresses X and Y and the client send to a packet to X,
the server should reply from X.
But the server may reply from Y without pktinfo.

RFC 1123:
When the local host is multihomed, a UDP-based request/response
application SHOULD send the response with an IP source address

that is the same as the specific destination address of the UDP
request datagram. The "specific destination address" is defined
in the "IP Addressing" section of the companion RFC [INTRO:1].

RFC 3542:
(Note: The hop limit is not contained in the in6_pktinfo structure
for the following reason. Some UDP servers want to respond to client
requests by sending their reply out the same interface on which the
request was received and with the source IPv6 address of the reply
equal to the destination IPv6 address of the request. To do this the
application can enable just the IPV6_RECVPKTINFO socket option and
then use the received control information from recvmsg() as the
outgoing control information for sendmsg(). The application need not
examine or modify the in6_pktinfo structure at all. But if the hop
limit were contained in this structure, the application would have to
parse the received control information and change the hop limit
member, since the received hop limit is not the desired value for an
outgoing packet.)

If this doesn't work, I think it is a bug of OpenBSD.
--
Tanaka Akira

#30 - 07/30/2011 08:35 AM - jeremyevans0 (Jeremy Evans)

- File ipv6_recvpktinfo_test.rb added

=begin
It appears that OpenBSD has the correct behavior without including the pktinfo argument. I'm attaching a test program. Here's two examples of usage and the output of each.

Without including pktinfo in the server's sendmsg call in response to the recvmsg:

$ ruby19 ipv6_recvpktinfo_test.rb fe80::92fb:a6ff:feed:afa1%re0
[:server_listening_on, #]
[:server_recvmsg_nonblock_output, "foo", #, [#]]
[:server_response_sendmsg_args, ["oof", 0, #]]
[:response_from_server, "oof", #]

Note in the last line of the output how the response received from the server uses the correct IPv6 address (fe80::92fb:a6ff:feed:afa1%re0 even though the server is listening on ::).

When including the pktinfo in the send message call, things break:

$ ruby19 ipv6_recvpktinfo_test.rb fe80::92fb:a6ff:feed:afa1%re0 pktinfo
[:server_listening_on, #]
[:server_recvmsg_nonblock_output, "foo", #, [#]]
[:server_response_sendmsg_args, ["oof", 0, #, #]]
ipv6_recvpktinfo_test.rb:23:in block in <main>': no response from #<Addrinfo: [fe80::92fb:a6ff:feed:afa1%re0]:10000 UDP> (RuntimeError)
from /usr/local/lib/ruby/1.9.1/socket.rb:45:inconnect_internal'
from /usr/local/lib/ruby/1.9.1/socket.rb:92:in connect'
from ipv6_recvpktinfo_test.rb:21:in'

Running tcpdump without the pktinfo argument shows no output, while running tcpdump with the pktinfo argument shows the following output:

16:28:55.962400 fe80::92fb:a6ff:feed:afa1.10000 > fe80::92fb:a6ff:feed:afa1.23171: [udp sum ok] udp 3 (len 11, hlim 64)
tcpdump: WARNING: compensating for unaligned libpcap packets
16:28:55.962407 fe80::1 > fe80::92fb:a6ff:feed:afa1: icmp6: fe80::92fb:a6ff:feed:afa1 unreachable address (len 59, hlim 64)

Note how the response is not a UDP packet but an ICMP packet with an fe80::1 source address saying the response is unreachable. This is why the program times out, no UDP response is received.

So maybe there is a bug here, but OpenBSD supports the desired feature without the pktinfo argument in the sendmsg call. If the pktinfo argument is needed on other systems to work correctly, would you support a patch that adds the pktinfo argument to the sendmsg call unless /openbsd/ =~ RUBY_PLATFORM ?

=end

#31 - 07/30/2011 09:23 AM - akr (Akira Tanaka)

2011/7/30 Jeremy Evans :

> 16:28:55.962400 fe80::92fb:a6ff:feed:afa1.10000 > fe80::92fb:a6ff:feed:afa1.23171: [udp sum ok] udp 3 (len 11, hlim 64)
> tcpdump: WARNING: compensating for unaligned libpcap packets
> 16:28:55.962407 fe80::1 > fe80::92fb:a6ff:feed:afa1: icmp6: fe80::92fb:a6ff:feed:afa1 unreachable address (len 59, hlim 64)

> Note how the response is not a UDP packet but an ICMP packet with an fe80::1 source address saying the response is unreachable. This is why the program times out, no UDP response is received.
>
> So maybe there is a bug here, but OpenBSD supports the desired feature without the pktinfo argument in the sendmsg call. If the pktinfo argument iseeded on other systems to work correctly, would you support a patch that adds the pktinfo argument to the sendmsg call unless /openbsd/ =~ RUBY_PLATFORM ?

It is possible if OpenBSD doesn't support dynamic IP change here.
(pktinfo is required for dynamic IP change.)

However I'd like to confirm it is really a bug of OpenBSD.

I think the best way to do is a bug repot to OpenBSD.

We need a test program written in C for good bug report, though.

Tanaka Akira

#32 - 07/30/2011 11:23 AM - kosaki (Motohiro KOSAKI)

> Issue #5097 has been updated by Eric Wong.
>
> File 0001-allocate-th-altstack-early-run-GC-on-allocation-fail.patch added
>
> Jeremy: does my patch to allocate th->altstack in a place where xmalloc() is
> possible help?
>
> Since altstack is always initialized if Ruby is compiled to use it, I see
> no reason to delay the allocation.

Looks nice. I've commited it.
However, I dropped following hunk.

#if defined(MINSIGSTKSZ) && (ALT_STACK_SIZE < MINSIGSTKSZ)
#undef ALT_STACK_SIZE
#define ALT_STACK_SIZE MINSIGSTKSZ
#endif

Because it's only works if 1) SIGSTKSZ*2 < MINSIGSTKSZ or 2) SIGSTKSZ is
no defined, but MINSIGSTKSZ is defined. I haven't seen such platforms.

Frankly, current altstack size (i.e. SIGSTKSZ*2) is quite too large.
It was introduced
following commit. But, in fact, sigaltstack raise an error if an
argument is less than
MINSIGSTKSZ, not SIGSTKSZ. So, current one is overkill large.

I'd like to commit altstack reducing patch to trunk (of course, not
193) and wait a feedback.

commit 7c9a77f88031afaba8338cf7188bfa7391f8a06a
Author: yugui yugui@b2dd03c8-39d4-4d8f-98ff-823fe69b080e
Date: Sun Nov 23 04:17:52 2008 +0000

```
 * signal.c (ALT_STACK_SIZE): 4KB is not enough on Mac OS X.
   Uses SIGSTKSZ.  this fixes [ruby-core:20040].

 git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@20331 b2dd03c8-39d4-4d8f-9
```

ChangeLog | 5 +++++
signal.c | 4 ++++
2 files changed, 9 insertions(+), 0 deletions(-)

diff --git a/signal.c b/signal.c
index 38dd55f..0c98237 100644
--- a/signal.c
+++ b/signal.c
@@ -416,7 +416,11 @@ typedef RETSIGTYPE (*sighandler_t)(int);

#ifdef POSIX_SIGNAL

#ifdef USE_SIGALTSTACK
+#ifdef SIGSTKSZ
+#define ALT_STACK_SIZE SIGSTKSZ
+#else
#define ALT_STACK_SIZE (4*1024)
+#endif

#33 - 08/01/2011 03:29 AM - Anonymous

Dne 26.7.2011 12:45, KOSAKI Motohiro napsal(a):

> 2011/7/26 Eric Wongnormalperson@yhbt.net:
>
> > Yui NARUSEnaruse@airemix.jp wrote:
> >
> > > If you want to support a platform, please declare.
> > > But when a platform dependent bug is reported, it will be assigned to you.
> > >
> > > RHEL, CentOS
> > > KOSAKI Motohiro
> > > 5.x or 6.x? I still have to deal with CentOS 5.x machines sometimes,
> > > so I can poke my head in if required.
> > > I only plan to maintain 6.x. So, your overture is really helpful to me.
>
> Thanks!

As I am maintainer of official RHEL versions of Ruby, please let me know
if I might be of some help.

Vit

#34 - 08/01/2011 05:53 PM - naruse (Yui NARUSE)

2011/7/26 Steve Klabnik steve@steveklabnik.com:

> Issue #5097 has been updated by Steve Klabnik.
>
> I would like to help for OSX > 10.6, and maybe 10.5 if I can get a machine. I'm not exactly sure what 'maintainer's responsibility exactly entails, but I'd be happy to help organize, make a best effort at patching bugs, and provide builds relating to that platform.

Thank you for comment.
But mrkn, who has both 10.6 and 10.5 and is already a committer, comes
to be a maintainer.

FYI, maintainer's responsibility is

- build and test ruby periodically
- make a patch or tell suitable people to fix it if they find a bug
- handle a platform dependent ticket
- make a decision of the platform

Anyway we are welcome to your contribution.
First one of the responsibility is most important everybody who has
the machine can do it.

--
NARUSE, Yui Â naruse@airemix.jp

#35 - 08/01/2011 05:53 PM - naruse (Yui NARUSE)

2011/7/27 Luis Lavena luislavena@gmail.com:

> On Mon, Jul 25, 2011 at 11:52 PM, Yui NARUSE naruse@airemix.jp wrote:
>
> > If you want to support a platform, please declare.
> > But when a platform dependent bug is reported, it will be assigned to you.
> > == Current Maintainer
> >
> > mswin32, mswin64 (Microsoft Windows):
> > NAKAMURA Usaku (usa)
> > mingw32 (Minimalist GNU for Windows):

> Nobuyoshi Nakada (nobu)
> I can contribute with all available resources to test Ruby against
> mingw32 and mingw-w64 on both 32bits and upcoming 64bits work.

nobu says he is not the maintainer of mingw.
Will you be the maitainer of mingw32 and mingw-w64?

#36 - 08/02/2011 05:12 AM - steveklabnik (Steve Klabnik)

> But mrkn, who has both 10.6 and 10.5 and is already a committer, comes
> to be a maintainer.

Excellent! I thought it was kind of strange that we'd be looking for someone on OSX...

I'll give them what help I can. :)

#37 - 08/08/2011 04:22 AM - jeremyevans0 (Jeremy Evans)

Akira Tanaka wrote:

> 2011/7/30 Jeremy Evans merch-redmine@jeremyevans.net:
>
> > 16:28:55.962400 fe80::92fb:a6ff:feed:afa1.10000 > fe80::92fb:a6ff:feed:afa1.23171: [udp sum ok] udp 3 (len 11, hlim 64)
> > tcpdump: WARNING: compensating for unaligned libpcap packets
> > 16:28:55.962407 fe80::1 > fe80::92fb:a6ff:feed:afa1: icmp6: fe80::92fb:a6ff:feed:afa1 unreachable address (len 59, hlim 64)
> >
> > Note how the response is not a UDP packet but an ICMP packet with an fe80::1 source address saying the response is unreachable. This is why the program times out, no UDP response is received.
> >
> > So maybe there is a bug here, but OpenBSD supports the desired feature without the pktinfo argument in the sendmsg call. If the pktinfo argument is needed on other systems to work correctly, would you support a patch that adds the pktinfo argument to the sendmsg call unless /openbsd/

I ended up writing a test program in C that also displayed the issue, and it turns out it was a bug in OpenBSD. A fix for the issue was just committed today.

Jeremy

#38 - 08/08/2011 08:23 AM - akr (Akira Tanaka)

2011/8/8 Jeremy Evans merch-redmine@jeremyevans.net:

> I ended up writing a test program in C that also displayed the issue, and it turns out it was a bug in OpenBSD. A fix for the issue was just committed today.

Great.

What's the URL of the bug report?
What's the OpenBSD release which will have the fix?

We have choices in Ruby now:

- disable ipv6_recvpktinfo for all OpenBSD.
- disable ipv6_recvpktinfo for OpenBSD older than the fix.
- enable ipv6_recvpktinfo for OpenBSD as now. -- Tanaka Akira

#39 - 08/08/2011 09:35 AM - jeremyevans0 (Jeremy Evans)

Akira Tanaka wrote:

> 2011/8/8 Jeremy Evans merch-redmine@jeremyevans.net:
>
> > I ended up writing a test program in C that also displayed the issue, and it turns out it was a bug in OpenBSD. A fix for the issue was just committed today.
>
> Great.

What's the URL of the bug report?
What's the OpenBSD release which will have the fix?

We have choices in Ruby now:

- disable ipv6_recvpktinfo for all OpenBSD.
- disable ipv6_recvpktinfo for OpenBSD older than the fix.
- enable ipv6_recvpktinfo for OpenBSD as now. -- Tanaka Akira

OpenBSD doesn't have a bug tracker, I posted to an internal mailing list. The fix will be in OpenBSD 5.0, released in November.

I recommend not making any changes. Now that the code works on OpenBSD, I don't see the point of adding an exception to support older versions. OpenBSD users are strongly encouraged to upgrade every 6 months when a new version is released, so there won't be that many people running < 5.0 a year from now.

Jeremy

#40 - 03/22/2012 09:22 AM - naruse (Yui NARUSE)

- Status changed from Assigned to Closed

Files

ruby193.log	529 KB	07/27/2011	jeremyevans0 (Jeremy Evans)
ruby193.diff	4.53 KB	07/27/2011	jeremyevans0 (Jeremy Evans)
ruby193.log	525 KB	07/28/2011	jeremyevans0 (Jeremy Evans)
ruby193.diff	7.21 KB	07/28/2011	jeremyevans0 (Jeremy Evans)
0001-allocate-th-altstack-early-run-GC-on-allocation-fail.patch	2.43 KB	07/29/2011	normalperson (Eric Wong)
ruby193.diff	2.23 KB	07/30/2011	jeremyevans0 (Jeremy Evans)
ipv6_recvpktinfo_test.rb	925 Bytes	07/30/2011	jeremyevans0 (Jeremy Evans)