Ruby master - Bug #9008

TestProcess#test_clock_getres_constants and TestProcess#test_clock_gettime_constants fails on ARM

10/09/2013 10:14 PM - vo.x (Vit Ondruch)

Status:	Rejected		
Priority:	Normal		
Assignee:	akr (Akira Tanaka)		
Target version:			
ruby -v:	ruby -v: ruby 2.1.0dev (2013-09-22 trunk 43011) [armv7hl-linux]	**Backport:**	1.9.3: UNKNOWN, 2.0.0: UNKNOWN

Description

=begin
I observe following two errors on ARM Building Ruby for Fedora Rawhide. It seems that (({:CLOCK_REALTIME_ALARM})) and (({:CLOCK_BOOTTIME_ALARM})) are not supported there. The error message is confusing, though :/

```
3) Error:
TestProcess#test_clock_getres_constants:
Errno::E524: Unknown error 524 - clock_getres
/builddir/build/BUILD/ruby-2.1.0-preview1/test/ruby/test_process.rb:1752:in clock_getres'
/builddir/build/BUILD/ruby-2.1.0-preview1/test/ruby/test_process.rb:1752:inblock in test_clock_getres_constants'
/builddir/build/BUILD/ruby-2.1.0-preview1/test/ruby/test_process.rb:1749:in each'
/builddir/build/BUILD/ruby-2.1.0-preview1/test/ruby/test_process.rb:1749:intest_clock_getres_constants'
4) Error:
TestProcess#test_clock_gettime_constants:
Errno::E524: Unknown error 524 - clock_gettime
/builddir/build/BUILD/ruby-2.1.0-preview1/test/ruby/test_process.rb:1676:in clock_gettime'
/builddir/build/BUILD/ruby-2.1.0-preview1/test/ruby/test_process.rb:1676:inblock in test_clock_gettime_constants'
/builddir/build/BUILD/ruby-2.1.0-preview1/test/ruby/test_process.rb:1673:in each'
/builddir/build/BUILD/ruby-2.1.0-preview1/test/ruby/test_process.rb:1673:intest_clock_gettime_constants'
```

I am going to resolve the issue temporary by applying

```
sed -i '/Process.constants.grep(\/\ACLOCK_\/).each {|n|/ s/$/\n      next if [:CLOCK_REALTIME_ALARM,
:CLOCK_BOOTTIME_ALARM].include? n/' \
test/ruby/test_process.rb
```

on ARM platform, but is there some better solution?
=end

History

#1 - 10/09/2013 10:23 PM - akr (Akira Tanaka)

2013/10/9 vo.x (Vit Ondruch) v.ondruch@tiscali.cz:

> Bug #9008: TestProcess#test_clock_getres_constants and TestProcess#test_clock_gettime_constants fails on ARM
> https://bugs.ruby-lang.org/issues/9008
>
> I observe following two errors on ARM Building Ruby for Fedora Rawhide. It seems that (({:CLOCK_REALTIME_ALARM})) and (({:CLOCK_BOOTTIME_ALARM})) are not supported there. The error message is confusing, though :/
>
> Errno::E524: Unknown error 524 - clock_gettime

This means clock_gettime() causes an error and errno is 524.

Why clock_gettime on Fedora Rawhide (ARM) doesn't cause EINVAL for
such constants?

http://pubs.opengroup.org/onlinepubs/9699919799/functions/clock_getres.html

```
|
|   [EINVAL]
|       The clock_id argument does not specify a known clock.
--
```

Tanaka Akira

#2 - 10/10/2013 01:29 AM - vo.x (Vit Ondruch)

> Why clock_gettime on Fedora Rawhide (ARM) doesn't cause EINVAL for
> such constants?

That is good question. I was hoping that you might know better ... So to say, I have no idea. I tried to identify the place where the error is risen, but I failed. I might try to report this issue to our Kernel maintainers.

#3 - 10/15/2013 03:52 AM - kosaki (Motohiro KOSAKI)

ENOTSUPP
Note:

524 mean ENOSUPP.
CLOCK_REALTIME_ALARM and CLOCK_BOOTTIME_ALARM uses following code. That says, it's a kernel bug.

linux/kernel/time/alarmtimer.c

/**

- alarm_clock_getres - posix getres interface
- @which_clock: clockid
- @tp: timespec to fill *

- Returns the granularity of underlying alarm base clock
  */
  static int alarm_clock_getres(const clockid_t which_clock, struct timespec *tp)
  {
  clockid_t baseid = alarm_bases[clock2alarm(which_clock)].base_clockid;

  ```
  if (!alarmtimer_get_rtcdev())
          return -ENOTSUPP;
  ```

  ```
  return hrtimer_get_res(baseid, tp);
  ```

  }

Any question?

#4 - 10/15/2013 03:57 AM - kosaki (Motohiro KOSAKI)

One more.

Linux man page agree with OpenGroup spec. see

http://man7.org/linux/man-pages/man2/clock_getres.2.html

> EINVAL The clk_id specified is not supported on this system.

#5 - 10/15/2013 04:53 PM - akr (Akira Tanaka)

2013/10/15 kosaki (Motohiro KOSAKI) kosaki.motohiro@gmail.com:

> Issue #9008 has been updated by kosaki (Motohiro KOSAKI).
>
> ENOTSUPP
> Note:
>
> 524 mean ENOSUPP.
>
> Any question?

Is ENOTSUPP related with ENOTSUP defined by POSIX?
--

Tanaka Akira

#6 - 10/15/2013 11:23 PM - kosaki (Motohiro KOSAKI)

(10/15/13 3:30 AM), Tanaka Akira wrote:

> 2013/10/15 kosaki (Motohiro KOSAKI) kosaki.motohiro@gmail.com:
>
> > Issue #9008 has been updated by kosaki (Motohiro KOSAKI).
> >
> > ENOTSUPP
> > Note:
> >
> > 524 mean ENOSUPP.
> >
> > Any question?
>
> Is ENOTSUPP related with ENOTSUP defined by POSIX?

dunno.

1. Linux doesn't have ENOTSUP
2. Linux define ENOTSUPP as NFS specific and in-kernel only definition
3. ENOTSUPP is not exported to userland
4. but timer_create and other various syscall uses ENOTSUPP as ENOTSUP

I guess this is historical mess. And I guess 4 is a bug. But I'm not sure.

#7 - 10/16/2013 12:22 AM - kosaki (Motohiro KOSAKI)

Oops. error.

Current glibc has following definition.

/usr/include/bits/errno.h

define ENOTSUP EOPNOTSUPP

So, my previous mail is wrong. ENOTSUPP is equal to ENOTSUP now.

#8 - 10/16/2013 07:53 AM - akr (Akira Tanaka)

2013/10/16 kosaki (Motohiro KOSAKI) kosaki.motohiro@gmail.com:

> Issue #9008 has been updated by kosaki (Motohiro KOSAKI).
>
> Current glibc has following definition.
>
> # /usr/include/bits/errno.h
>
> # define ENOTSUP EOPNOTSUPP

EOPNOTSUPP is not ENOTSUPP.

If errno 524 is defined as an error symbol Ruby knows,
Errno::XXX should be defined.
(Ruby knows ENOTSUP and EOPNOTSUPP.)

Anyway I understand ENOTSUP is not intended to used in userland.
So Ruby don't need to know it.
--
Tanaka Akira

#9 - 10/16/2013 10:53 AM - kosaki (Motohiro KOSAKI)

On Tue, Oct 15, 2013 at 6:48 PM, Tanaka Akira akr@fsij.org wrote:

> 2013/10/16 kosaki (Motohiro KOSAKI) kosaki.motohiro@gmail.com:
>
> > Issue #9008 has been updated by kosaki (Motohiro KOSAKI).

> > Current glibc has following definition.
> >
> > # /usr/include/bits/errno.h
> >
> > # define ENOTSUP EOPNOTSUPP
>
> EOPNOTSUPP is not ENOTSUPP.

Right. sorry for noise.

> If errno 524 is defined as an error symbol Ruby knows,
> Errno::XXX should be defined.
> (Ruby knows ENOTSUP and EOPNOTSUPP.)
>
> Anyway I understand ENOTSUP is not intended to used in userland.
>
> # So Ruby don't need to know it.
>
> Tanaka Akira

#10 - 10/16/2013 05:30 PM - vo.x (Vit Ondruch)

Just for the record, this is discussed on Fedora's ARM list as well: https://lists.fedoraproject.org/pipermail/arm/2013-October/006965.html

#11 - 10/19/2013 09:00 AM - kosaki (Motohiro KOSAKI)

- *Status changed from Open to Rejected*

I and John Stulz agreed we need Linux kernel and now my patch is queued for linux 3.13.
It mean Linux 3.13 and later return EINVAL instead 524.

Closed then.

#12 - 10/20/2013 07:08 AM - vo.x (Vit Ondruch)

Thank you for sorting this out!

#13 - 10/20/2013 10:53 AM - kosaki (Motohiro KOSAKI)

> I and John Stulz agreed we need Linux kernel

s/need Linux kernel/need to fix Linux kernel/, of course.

#14 - 02/06/2014 07:56 AM - vo.x (Vit Ondruch)

Just FYI, the test suite passes on Fedora Rawhide with 3.12.9-300.fc20.armv7hl Kernel. Thanks for pushing the fix upstream.