Ruby trunk - Bug #1388

cygwin-1.7, gcc4-4.3, and ruby-1.9. make btest #236 test_io.rb Segmentation fault

04/18/2009 04:56 AM - neomjp (neomjp neomjp)

Status:	Rejected		
Priority:	Normal		
Assignee:	nobu (Nobuyoshi Nakada)		
Target version:	2.6		
ruby -v:	ruby 1.9.2dev (2009-04-08 trunk 23198) [i386-cygwin]	**Backport:**	2.3: UNKNOWN, 2.4: UNKNOWN

Description

=begin
Cygwin 1.7 is currently under beta testing. It is currently at cygwin-1.7.0-46. If nothing goes overly wrong, the official 1.7.1 is planned to be released in June.
http://sourceware.org/ml/cygwin-announce/2009-04/msg00025.html

Two issues blocking the release are:

1) Stabilization of gcc-4.3; It is currently at gcc4-4.3.2-2, and several to-do's remain.
http://sourceware.org/ml/cygwin/2009-03/msg00378.html
http://sourceware.org/ml/cygwin/2009-03/msg00422.html
Hopefully it will get ready in gcc4-4.3.2-3.

2) Compilation of all packages using the stable gcc-4.3.

This bug report is about making ruby-1.9 ready for these new cygwin-1.7 and gcc-4.3. These are some of the patches required to make ruby trunk get compiled.

- eval_intern.h [CYGWIN]: Remove #ifdef **CYGWIN** for _setjmp() and _longjmp(). Cygwin-1.7 has its own definition in /usr/include/machine/setjmp.h . This is the minimally required patch to make the compilation go through to the end.

```
--- origsrc/ruby-1.9.2-r23198/eval_intern.h     2009-02-22 10:43:59.000000000 +0900
+++ src/ruby-1.9.2-r23198/eval_intern.h 2009-04-18 01:26:41.843750000 +0900
@@ -66,9 +66,6 @@ char *strrchr(const char *, const char);

#define ruby_setjmp(env) RUBY_SETJMP(env)
#define ruby_longjmp(env,val) RUBY_LONGJMP(env,val)
-#ifdef CYGWIN
-int _setjmp(), _longjmp();
-#endif

#include
#include
```

- ruby.c (push_include_cygwin): Use cygwin_conv_path instead of cygwin_conv_to_posix_path which is deprecated in cygwin-1.7.
- ruby.c (ruby_init_loadpath_safe): Use cygwin_conv_path instead of cygwin_conv_to_posix_path which is deprecated in cygwin-1.7.

```
--- origsrc/ruby-1.9.2-r23198/ruby.c    2009-03-17 10:29:17.000000000 +0900
+++ src/ruby-1.9.2-r23198/ruby.c        2009-04-18 01:26:41.859375000 +0900
@@ -257,7 +257,8 @@ push_include_cygwin(const char *path, VA
p = strncpy(RSTRING_PTR(buf), p, len);
}
}
```

- if (cygwin_conv_to_posix_path(p, rubylib) == 0)
- if (cygwin_conv_path(CCP_WIN_W_TO_POSIX | CCP_RELATIVE, p, rubylib, 1)
- == 0)   p = rubylib;   push_include(p, filter);   if (!*s) break; @@ -366,8 +367,10 @@ ruby_init_loadpath_safe(int safe_level) #elif defined **CYGWIN**   {   char rubylib[FILENAME_MAX];
- cygwin_conv_to_posix_path(libpath, rubylib);
- strncpy(libpath, rubylib, sizeof(libpath));

- if (cygwin_conv_path(CCP_WIN_W_TO_POSIX | CCP_RELATIVE,
- libpath, rubylib, 1)
- == 0)

- [redacted]

  ```
  }
  #endif
  p = strrchr(libpath, '/');
  ```

- strftime.c [CYGWIN]: Cygwin defines _timezone, _daylight, *_tzname[2], and tzname
  with dllimport attribute. But defines daylight and timezone without
  dllimport attribute.

```
--- origsrc/ruby-1.9.2-r23198/strftime.c        2009-03-17 10:29:17.000000000 +0
900
+++ src/ruby-1.9.2-r23198/strftime.c    2009-04-18 01:26:41.859375000 +0900
@@ -120,12 +120,16 @@ extern char *strchr();

#define range(low, item, hi)   max(low, min(item, hi))

-#if defined WIN32 || defined WIN32
+#if defined __CYGWIN_ || defined WIN32 || defined WIN32
#define DLL_IMPORT __declspec(dllimport)
#endif
#ifndef DLL_IMPORT
#define DLL_IMPORT
#endif
+#ifdef __CYGWIN_
+#define daylight _daylight
+#define timezone _timezone
+#endif
#if !defined(OS2) && defined(HAVE_TZNAME)
extern DLL_IMPORT char *tzname[2];
#ifdef HAVE_DAYLIGHT
```

With the above three patches, ruby-1.9.2-r23198 can get compiled with only one warning:

```
** PTHREAD SUPPORT MODE WARNING:
**
**   Ruby is compiled with --enable-pthread, but your Tcl/Tk library
**   seems to be compiled without pthread support. Although you can
...
```

This is expected because cygwin tcltk-20080420-1 is compiled without pthread support. But when I try to compile like

```
CC=gcc-4 configure --program-suffix="-19" --disable-pthread
make
```

compilation fails.

```
make: *** No rule to make target thread_.h', needed byminiprelude.o'.  Stop.
*** ERROR: make failed
```

This is because THREAD_MODEL is empty in Makefile. Looking into configure.in, I can see that when

```
if test "$rb_with_pthread" = "yes";
```

is false and

```
case "$target_os" in
when(cygwin*)
```

then THREAD_MODEL gets undefined. (when(mingw*) is true, THREAD_MODEL=win32.) If I compile like

```
CC=gcc-4 configure --program-suffix="-19" --disable-pthread
make THREAD_MODEL=w32
```

the compilation goes through to the end, and thread-win32.c seems to be used instead of thread-pthread.c. But the same warning persists.

```
** PTHREAD SUPPORT MODE WARNING:
**
**   Ruby is compiled with --enable-pthread, but your Tcl/Tk library
**   seems to be compiled without pthread support. Although you can
...
```

This is wrong because --disable-pthread is used. Looking into ext/tk/extconf.rb, I can see that this warning is emitted when

```
# check pthread mode
if (macro_defined?('HAVE_NATIVETHREAD', '#include "ruby.h"'))
# ruby -> enable
unless tcl_enable_thread
# ruby -> enable && tcl -> disable
```

But include/ruby/ruby.h has

```
#define HAVE_NATIVETHREAD
```

without any #ifdefs. So the pthread mode check in ext/tk/extconf.rb always evaluates to be true even when pthread support is disabled. This should be corrected. If these issues are corrected, then ruby-1.9 trunk can get compiled without warnings.

When I tried make run or make runruby, it failed.

- common.mk (TESTRUN_SCRIPT): Correct the path to test.rb

```
--- origsrc/ruby-1.9.2-r23198/common.mk    2009-04-10 11:32:15.000000000 +0900
+++ src/ruby-1.9.2-r23198/common.mk        2009-04-18 04:35:13.968750000 +0900
@@ -117,7 +117,7 @@
TESTSDIR      = $(srcdir)/test
TESTWORKDIR   = testwork

-TESTRUN_SCRIPT = $(srcdir)/test.rb
+TESTRUN_SCRIPT = $(srcdir)/sample/test.rb

BOOTSTRAPRUBY = $(BASERUBY)
```

With this patch, the results of make run or runruby are

```
make run
not ok/test: 900 failed 1
Fnot ok system 9 -- .../ruby-1.9.2-r23198/sample/test.rb:1948:in `'

make runruby
end of test(test: 900)
```

which is expected and good. miniruby.exe does not support euc-jp, shift_jis, windows-1251, cp932 in Encoding.name_list, so make run is expected to fail at that test. But the result of make btest is bad.

#236 test_io.rb:
at_exit { p :foo }

```
    megacontent = "abc" * 12345678
    #File.open("megasrc", "w") {|f| f << megacontent }

    Thread.new { sleep rand*0.2; Process.kill(:INT, $$) }

    r1, w1 = IO.pipe
    r2, w2 = IO.pipe
    t1 = Thread.new { w1 << megacontent; w1.close }
    t2 = Thread.new { r2.read }
    IO.copy_stream(r1, w2) rescue nil
    r2.close; w2.close
    r1.close; w1.close
    #=> killed by SIGABRT (signal 6)
```

```
| bootstraptest.tmp.rb:2: [BUG] Segmentation fault
| ruby 1.9.2dev (2009-04-15 trunk 23198) [i386-cygwin]
|
| -- control frame ----------
| c:0004 p:---- s:0010 b:0010 l:000009 d:000009 CFUNC  :p
| c:0003 p:0011 s:0006 b:0006 l:000aec d:000005 BLOCK  bootstraptest.tmp.rb:2
| c:0002 p:---- s:0004 b:0004 l:000003 d:000003 FINISH
| c:0001 p:0000 s:0002 b:0002 l:000aec d:000aec TOP    :19
| ---------------------------
| bootstraptest.tmp.rb:2:in block in <main>'
| bootstraptest.tmp.rb:2:inp'
|
| [NOTE]
| You may have encountered a bug in the Ruby interpreter or extension libraries.
| Bug reports are welcome.
| For details: http://www.ruby-lang.org/bugreport.html
|

FAIL 1/890 tests failed
make: *** [btest] Error 1
```

make btest-ruby also emits several errors, but I will submit it as another issue because this report is already too long...
=end

History

#1 - 05/01/2009 12:57 AM - neomjp (neomjp neomjp)

=begin
Thanks for the quick and thorough review. I am sorry that I could not
report back earlier.

On 2009/04/19 20:12, Nobuyoshi Nakada wrote:

> At Sat, 18 Apr 2009 04:56:10 +0900,
> neomjp neomjp wrote in [ruby-core:23241]:
>
>> -#ifdef **CYGWIN**
>> -int _setjmp(), _longjmp();
>> -#endif
>
> The definitions seem just with extern and arguments, and above
> declaration doesn't seem conflict with them, what error does
> occur?

In file included from .../ruby-1.9.2-r23198/eval.c:14:
.../ruby-1.9.2-r23198/eval_intern.h:70: error: conflicting types for '_longjmp'
/usr/include/machine/setjmp.h:318: error: previous declaration of '_longjmp' was here
make: *** [eval.o] Error 1

Conficting part is _longjmp. Here is the relevant part of setjmp.h from
cygwin-1.7 .

$ cygcheck -f /usr/include/machine/setjmp.h
cygwin-1.7.0-46

$ sed -n 317,323p /usr/include/machine/setjmp.h
#ifdef **CYGWIN**
extern void _longjmp(jmp_buf, int);
extern int _setjmp(jmp_buf);
#else
#define _setjmp(env)   sigsetjmp ((env), 0)
#define _longjmp(env, val)   siglongjmp ((env), (val))
#endif

In contrast, cygwin-1.5 did not have _setjmp or _longjmp

```
$ cygcheck -f /usr/include/machine/setjmp.h
cygwin-1.5.25-15

$ grep -Ecr --include=setjmp* "_longjmp|_setjmp" /usr/include/
/usr/include/machine/setjmp-dj.h:0
/usr/include/machine/setjmp.h:0
/usr/include/setjmp.h:0
```

- if (cygwin_conv_to_posix_path(p, rubylib) == 0)
- if (cygwin_conv_path(CCP_WIN_W_TO_POSIX | CCP_RELATIVE, p, rubylib, 1)
- == 0)

> I suspect it should use CCP_WIN_A_TO_POSIX and sizeof(rubylib)
> instead of 1, am I wrong?

```
You are totally right. Stupid me, I just read "If size is 0 ...
```

Otherwise, ...", and set it to a non-zero value.

> Previously, it couldn't work with THREAD_MODEL=win32, maybe
> something improved with cygwin 1.7?

```
I investigated this furthur, and found that it is probably not the
```

case. This Makefile variable THREAD_MODEL is used in two places in (un)common.mk, the variable VM_CORE_H_INCLUDES and the prerequisite for thread.o:

```
VM_CORE_H_INCLUDES = {$(VPATH)}vm_core.h {$(VPATH)}vm_opts.h \
{$(VPATH)}thread_$(THREAD_MODEL).h \
{$(VPATH)}node.h $(ID_H_INCLUDES)

thread.$(OBJEXT): {$(VPATH)}thread.c {$(VPATH)}eval_intern.h \
$(RUBY_H_INCLUDES) {$(VPATH)}gc.h $(VM_CORE_H_INCLUDES) \
{$(VPATH)}debug.h {$(VPATH)}thread_$(THREAD_MODEL).c
```

So, the variable THREAD_MODEL is not used in any rules. thread_$(THREAD_MODEL).c is #included from thread.c like this:

```
#if   defined(_WIN32)
#include "thread_win32.c"
...............................
#elif defined(HAVE_PTHREAD_H)
#include "thread_pthread.c"
...................................
#else
#error "unsupported thread type"
#endif
```

But in cygwin, _WIN32 is undefined, and HAVE_PTHREAD_H is defined. So thread_pthread.c is included. If I run the preprocessoer like

```
gcc-4 -v -E -O2 -pipe  -I. -I.ext/include/i386-cygwin
-I.../ruby-1.9.2-r23311/include -I.../ruby-1.9.2-r23311 -DRUBY_EXPORT
-o thread.o -c .../ruby-1.9.2-r23311/thread.c
```

This gives:

```
/usr/lib/gcc/i686-pc-cygwin/4.3.2/cc1.exe ... -D_CYGWIN32_
-D_CYGWIN_ -Dunix -D_unix_ -D__unix ...
```

and

```
static void timer_thread_function(void *);
# 182 ".../ruby-1.9.2-r23311/thread.c"
# 1 ".../ruby-1.9.2-r23311/thread_pthread.c" 1
# 17 ".../ruby-1.9.2-r23311/thread_pthread.c"
# 1 "/usr/include/sys/resource.h" 1 3 4
# 41 "/usr/include/sys/resource.h" 3 4
typedef unsigned long rlim_t;
....
```

Note that thread_pthread.c is #included instead of thread_win32.c.

So what happens with

CC=gcc-4 configure --program-suffix="-19" --disable-pthread
make THREAD_MODEL=w32

is

1. thread_pthread.c is #included from thread.c. (not thread_win32.c)
2. Objects are linked without -lpthread.

What kind of thread is working here? Anyway, both with/without
--disable-pthread passed test_thread.rb in make btest. .

> make run is supporsed to run your own script, so test.rb is a
> file which you should make. test-sample is what you want.

```
I see. "make test-sample" passes without errors, both with/without
```

--disable-pthread.

>> #236 test_io.rb:
>
> Segfaults in the at_exit block. I'll investigate it.

Thanks.

=end

#2 - 05/07/2009 02:03 PM - nobu (Nobuyoshi Nakada)

=begin
Hi,

At Fri, 1 May 2009 00:57:41 +0900,
neomjp neomjp wrote in [ruby-core:23340]:

> Conficting part is _longjmp. Here is the relevant part of setjmp.h from
> cygwin-1.7 .
>
> $ sed -n 317,323p /usr/include/machine/setjmp.h
> #ifdef **CYGWIN**
> extern void _longjmp(jmp_buf, int);
> extern int _setjmp(jmp_buf);

Yes of course, longjmp() never return and must not be int.

>> Previously, it couldn't work with THREAD_MODEL=win32, maybe
>> something improved with cygwin 1.7?
>
> I investigated this furthur, and found that it is probably not the
> case. This Makefile variable THREAD_MODEL is used in two places in
> (un)common.mk, the variable VM_CORE_H_INCLUDES and the prerequisite for
> thread.o:

I meant very early implementation, but not current one. It had
used Windows threads at first.

--
Nobu Nakada

=end

#3 - 05/12/2009 01:10 AM - neomjp (neomjp neomjp)

=begin
On 2009/05/01 0:57, neomjp neomjp wrote:

CC=gcc-4 configure --program-suffix="-19" --disable-pthread
make THREAD_MODEL=w32

1. Objects are linked without -lpthread.

```
It seems the miniruby was still using pthread even when linked
```

without -lpthread. The only difference in
"strings miniruby | grep -i pthread"
with/without --disable-pthread was the absence/presence of

pthread_attr_setinheritsched(&attr, PTHREAD_INHERIT_SCHED)

All other pthread functions were the same. miniruby was still
using pthread.

```
So, I tried forcing the compilation of thread_win32.c by replacing
```

#if   defined(WIN32)
with
#if   defined(_WIN32) || defined(CYGWIN_)
in thread.c:172 and vm_core.h:25 (r23390), and

CC=gcc-4 configure --program-suffix="-19" --disable-pthread
make THREAD_MODEL=w32

The compilation went through to the end (with some warnings), but
"make btest" failed miserably with numerous segfaults and four test
failures.

```
Hmm, now I understand that win32 thread does not work in cygwin.
```

I will take back my claims about the option to --disable-pthread in
cygwin-1.7. It was not the main topic of this bug, anyway.
Besides, it was a rather low-priority feature request in a non-default
setting.

Finally, an update:

- eval_intern.h: FIXED in r23317. Thanks.
- ruby.c: Nobu's fix in [ruby-core:23255] will be fine.
- strftime.c: A patch proposed in [ruby-core:23241].
- common.mk :INVALID, WONTFIX
- Segfault in #236 test_io.rb: This was what this bug was about.

--
neomjp
=end

#4 - 07/16/2009 04:32 PM - yugui (Yuki Sonoda)

- Assignee set to nobu (Nobuyoshi Nakada)

=begin

=end

#5 - 05/25/2010 12:51 AM - mame (Yusuke Endoh)

- Priority changed from Normal to 3

- Target version set to 2.0.0

=begin
Hi,

neomjp, we really appreciate your contribution for cygwin support, but
very sorry, we can't afford to review and test your patch because there
is no maintainer for cygwin.
Also, we have no enough time to test it for 1.9.2 release.
So I set this ticket to Low-priority.

A maintainer is required to add cygwin into "best effort" platform:

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

Are you interested?

--
Yusuke Endoh mame@tsg.ne.jp
=end

#6 - 06/03/2010 10:45 AM - usa (Usaku NAKAMURA)

- *Status changed from Open to Assigned*

=begin

=end

#7 - 07/31/2010 04:54 AM - neomjp (neomjp neomjp)

=begin
Hi,

After a long hiatus, I checked the status of this make btest, test_io.rb, segfault bug.

In trunk,

ruby-1.9.2-r23198 segfault (<- when this bug was reported.)
ruby-1.9.2-preview1 (r24184) segfault
ruby-1.9.2-preview2 (r24782) segfault
ruby-1.9.3-r27622 segfault
ruby-1.9.3-r27623 timeout or pass but no segfault (<- fix for test_io.rb
megacontent-copy_stream deadlock)
ruby-1.9.3-r28731 timeout or pass but no segfault

In ruby_1_9_2 branch,

ruby-1.9.2-preview3 (r28108) Too many "[BUG] pthread_mutex_unlock : Operation not permitted
(EPERM)" errors. Not sure if this segfault occurs.
ruby-1.9.2-r28508 timeout or pass but no segfault (<- fix for pthread bug)
ruby-1.9.2-rc1 (r28522) timeout or pass but no segfault
ruby-1.9.2-rc2 (r28613) timeout or pass but no segfault
ruby-1.9.2-r28724 timeout or pass but no segfault

In ruby_1_9_1 branch,

ruby-1.9.1-p429 (r28522) segfault
ruby-1.9.1-r28641 segfault

So, this segfault was seen only before the test was changed in r27623. After the
fix, the test will either pass, or timeout as show below:

#246 test_io.rb:
at_exit { p :foo }

```
    megacontent = "abc" * 12345678
    #File.open("megasrc", "w") {|f| f << megacontent }

    Thread.new { sleep rand*0.2; Process.kill(:INT, $$) }

    r1, w1 = IO.pipe
    r2, w2 = IO.pipe
    t1 = Thread.new { w1 << megacontent; w1.close }
    t2 = Thread.new { r2.read; r2.close }
    IO.copy_stream(r1, w2) rescue nil
    w2.close
    r1.close
    t1.join
    t2.join
    #=> killed by SIGKILL (signal 9) (timeout)  megacontent-copy_stream
```

FAIL 1/925 tests failed
make: *** [yes-btest] Error 1

What happens when it timeouts? When this test was isolated in a file and executed, it
sometimes showed a hang (or deadlock?). Maybe, the pipes were not properly killed?

1. I do not see a segfault any more. I see a pass or timeout (a hang or deadlock, meaning the pipes were not properly killed) instead.
2. r27623 may be ported also to ruby_1_9_1 branch. It would turn the second test failure reported in Bug #3292 [ruby-core:30238] from a segfault into a timeout.
3. The patch for ruby.c in [ruby-core:23255] was incorporated in r23468.
4. The declarations in strftime.c that the patch in [ruby-core:23241] [Bug #1388] tried to fix were removed in r28592. So, the patch is no more valid.
5. As for maintainership, I would be glad if I could be of some help, but I do not think I can promise to keep the 3 months rule in [ruby-core:25764]. Sometimes, I can compile ruby and run tests, but other times, my daily work will not allow me the time. I should better remain just another cygwin tester. =end

#8 - 02/18/2013 09:06 PM - mame (Yusuke Endoh)

- Description updated

- Target version changed from 2.0.0 to 2.6

#9 - 10/19/2017 02:25 PM - mame (Yusuke Endoh)

- Status changed from Assigned to Rejected

Sorry for leaving this ticket untouched, but Ruby 1.9 has been EOL status since a long time ago. If there is anyone who is willing to maintain cygwin ruby, please investigate the current status of the issue and write a patch.