

# Ruby master - Bug #4696

## thread.c#lock\_func() spurious wakeup unsafe

05/15/2011 04:22 PM - kosaki (Motohiro KOSAKI)

<b>Status:</b> Closed	
<b>Priority:</b> Normal	
<b>Assignee:</b> kosaki (Motohiro KOSAKI)	
<b>Target version:</b> 1.9.3	
<b>ruby -v:</b> -	<b>Backport:</b>

### Description

lock\_func (ie Mutex.lock) spurious wakeup unsafe

```
static int
lock_func(rb_thread_t *th, mutex_t *mutex, int timeout_ms)
{
  for (;;) {
    if (!mutex->th) {
      mutex->th = th;
      break;
    }

    mutex->cond_waiting++;
    if (timeout_ms) {
      ret = native_cond_timedwait(&mutex->cond, &mutex->lock, &timeout);
      if (ret == ETIMEDOUT) {
        interrupted = 2;
        mutex->cond_waiting--;
        break;
      }
    }
    else {
      native_cond_wait(&mutex->cond, &mutex->lock);
    }
    mutex->cond_notified--;

    if (RUBY_VM_INTERRUPTED(th)) {
      interrupted = 1;
      break;
    }
  }
  return interrupted;
}
```

- 1) native\_cond\_wait() spurious wakeup native\_cond\_signal() cond\_notified
- 2) pthread\_cond\_timedwait() pthread\_cond\_signal() ETIMEDOUT mutex->cond\_waiting cond wait predicate mutex->th RUBY\_VM\_INTERRUPTED ETIMEDOUT

```
(1) deadlock check YARV
mutex->cond_waiting
(2) pthread_cond_timedwait r31373

spurious wakeup native_cond_signal() wakeup
cond_signal+1 wakeup-1

deadlock
lock_func unlock
mutex->th NULL

Linux
pthread_cond_wait() glibc spurious wakeup
spurious wakeup linux thread EINTR
thread_pthread.c#native_cond_wait() (r31482)
pthread_cond_timedwait() glibc
clock_gettime()
ETIMEDOUT

NetBSD
pthread_cond_timedwait() ETIMEDOUT

Mac
cond_wait() 0
```

**Subtasks:**

Bug # 4723: check\_deadlock\_i transition\_for\_lock thread unsafe

Closed

**Associated revisions**

**Revision 31436 - 05/05/2011 06:32 AM - nagachika (Tomoyuki Chikanaga)**

- eval.c (frame\_func\_id): **method** return different name from methods defined by Module#define\_method with a same block. [ruby-core:35386] fixes #4606
  - eval (method\_entry\_of\_iseq): new helper function. search control frame stack for a method entry which has given iseq.
  - test/ruby/test\_method.rb: add tests for #4696

**Revision 31436 - 05/05/2011 06:32 AM - nagachika (Tomoyuki Chikanaga)**

- eval.c (frame\_func\_id): **method** return different name from methods defined by Module#define\_method with a same block. [ruby-core:35386] fixes #4606
  - eval (method\_entry\_of\_iseq): new helper function. search control frame stack for a method entry which has given iseq.
  - test/ruby/test\_method.rb: add tests for #4696

**Revision 31436 - 05/05/2011 06:32 AM - nagachika (Tomoyuki Chikanaga)**

- eval.c (frame\_func\_id): **method** return different name from methods defined by Module#define\_method with a same block. [ruby-core:35386] fixes #4606
  - eval (method\_entry\_of\_iseq): new helper function. search control frame stack for a method entry which has given iseq.
  - test/ruby/test\_method.rb: add tests for #4696

**Revision 31436 - 05/05/2011 06:32 AM - nagachika (Tomoyuki Chikanaga)**

- eval.c (frame\_func\_id): **method** return different name from methods defined by Module#define\_method with a same block. [ruby-core:35386] fixes #4606
  - eval (method\_entry\_of\_iseq): new helper function. search control frame stack for a method entry which has given iseq.
  - test/ruby/test\_method.rb: add tests for #4696

**Revision 31436 - 05/05/2011 06:32 AM - nagachika (Tomoyuki Chikanaga)**

- eval.c (frame\_func\_id): **method** return different name from methods defined by Module#define\_method with a same block. [ruby-core:35386] fixes #4606
  - eval (method\_entry\_of\_iseq): new helper function. search control frame stack for a method entry which has given iseq.
  - test/ruby/test\_method.rb: add tests for #4696

**Revision 31436 - 05/05/2011 06:32 AM - nagachika (Tomoyuki Chikanaga)**

- eval.c (frame\_func\_id): **method** return different name from methods defined by Module#define\_method with a same block. [ruby-core:35386] fixes #4606
  - eval (method\_entry\_of\_iseq): new helper function. search control frame stack for a method entry which has given iseq.
  - test/ruby/test\_method.rb: add tests for #4696

**Revision 614fcdf6 - 06/12/2011 06:55 AM - kosaki (Motohiro KOSAKI)**

- thread.c: introduce spurious wakeup safe deadlock check. [Bug #4696][ruby-dev:43554]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32021 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 32021 - 06/12/2011 06:55 AM - kosaki (Motohiro KOSAKI)**

- thread.c: introduce spurious wakeup safe deadlock check. [Bug #4696][ruby-dev:43554]

**Revision 32021 - 06/12/2011 06:55 AM - kosaki (Motohiro KOSAKI)**

- thread.c: introduce spurious wakeup safe deadlock check. [Bug #4696][ruby-dev:43554]

**Revision 32021 - 06/12/2011 06:55 AM - kosaki (Motohiro KOSAKI)**

- thread.c: introduce spurious wakeup safe deadlock check. [Bug #4696][ruby-dev:43554]

**Revision 32021 - 06/12/2011 06:55 AM - kosaki (Motohiro KOSAKI)**

- thread.c: introduce spurious wakeup safe deadlock check. [Bug #4696][ruby-dev:43554]

**Revision 32021 - 06/12/2011 06:55 AM - kosaki (Motohiro KOSAKI)**

- thread.c: introduce spurious wakeup safe deadlock check. [Bug #4696][ruby-dev:43554]

**Revision 32021 - 06/12/2011 06:55 AM - kosaki (Motohiro KOSAKI)**

- thread.c: introduce spurious wakeup safe deadlock check. [Bug #4696][ruby-dev:43554]

**Revision e3672626 - 06/13/2011 12:49 PM - kosaki (Motohiro KOSAKI)**

- thread\_win32.c (native\_cond\_signal): remove unnecessary rb\_bug(). It's additional fix for r32021. [Bug #4696]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@32059 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 32059 - 06/13/2011 12:49 PM - kosaki (Motohiro KOSAKI)**

- thread\_win32.c (native\_cond\_signal): remove unnecessary rb\_bug(). It's additional fix for r32021. [Bug #4696]

**Revision 32059 - 06/13/2011 12:49 PM - kosaki (Motohiro KOSAKI)**

- thread\_win32.c (native\_cond\_signal): remove unnecessary rb\_bug(). It's additional fix for r32021. [Bug #4696]

**Revision 32059 - 06/13/2011 12:49 PM - kosaki (Motohiro KOSAKI)**

- thread\_win32.c (native\_cond\_signal): remove unnecessary rb\_bug(). It's additional fix for r32021. [Bug #4696]

**Revision 32059 - 06/13/2011 12:49 PM - kosaki (Motohiro KOSAKI)**

- thread\_win32.c (native\_cond\_signal): remove unnecessary rb\_bug(). It's additional fix for r32021. [Bug #4696]

**Revision 32059 - 06/13/2011 12:49 PM - kosaki (Motohiro KOSAKI)**

- thread\_win32.c (native\_cond\_signal): remove unnecessary rb\_bug(). It's additional fix for r32021. [Bug #4696]

**Revision 32059 - 06/13/2011 12:49 PM - kosaki (Motohiro KOSAKI)**

- thread\_win32.c (native\_cond\_signal): remove unnecessary rb\_bug(). It's additional fix for r32021. [Bug #4696]

**History**

---

**#1 - 05/15/2011 04:23 PM - kosaki (Motohiro KOSAKI)**

- File 0001-new-deadlock-check.patch added

XXXXXXXXXXXXXXXXXXXX

**#2 - 05/15/2011 06:23 PM - mame (Yusuke Endoh)**

- ruby -v changed from ruby 1.9.3dev (2011-05-13 trunk 31548) [x86\_64-linux] to -

XXXXXX  
XXXXXXXXXXXXXXXXXXXX (XXXXXXXXXXXXXXXXXXXX)

2011051516:22 Motohiro KOSAKI [kosaki.motohiro@gmail.com](mailto:kosaki.motohiro@gmail.com):

XXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

XX  
XX  
XXXXXX

XX  
XXXXXX (XX  
XXXX) XXX

1.9.3 XXX  
XXXXXX revert XXXXXXXXXXXXXXXXXXXXXXX

# XXXtransition\_for\_lock XXX volatile XXXXXXXXXXXXXXXXXXXXXXX

--  
Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

**#3 - 05/15/2011 06:23 PM - mame (Yusuke Endoh)**

XXXXXX  
XXXXXXXXXXXXXXXXXXXX (XXXXXXXXXXXXXXXXXXXX)

2011051516:22 Motohiro KOSAKI [kosaki.motohiro@gmail.com](mailto:kosaki.motohiro@gmail.com):

XXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXX

XX  
XX  
XXXXXX

XX  
XXXXXX (XX  
XXXX) XXX

1.9.3  
revert

# transition\_for\_lock volatile

--  
Yusuke Endoh [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp)

**#4 - 05/15/2011 09:29 PM - kosaki (Motohiro KOSAKI)**

20110515 17:31 Yusuke ENDOH [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp):

thread.c  
(thread.c)

pthread  
pthread

Ruby::ConditionVariable  
Ruby  
python processing

**#5 - 05/15/2011 09:29 PM - kosaki (Motohiro KOSAKI)**

20110515 17:31 Yusuke ENDOH [mame@tsg.ne.jp](mailto:mame@tsg.ne.jp):

thread.c  
(thread.c)

pthread  
pthread

Ruby::ConditionVariable  
Ruby  
python processing

**#6 - 06/12/2011 03:55 PM - kosaki (Motohiro KOSAKI)**

- Status changed from Assigned to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r32021.  
Motohiro, thank you for reporting this issue.  
Your contribution to Ruby is greatly appreciated.  
May Ruby be with you.

- thread.c: introduce spurious wakeup safe deadlock check. [Bug #4696][ruby-dev:43554]

**Files**

0001-new-deadlock-check.patch	3.99 KB	05/15/2011	kosaki (Motohiro KOSAKI)
-------------------------------	---------	------------	--------------------------