

## Ruby trunk - Bug #9739

### TestException#test\_machine\_stackoverflow(\_by\_define\_method) failures on x64-mingw32

04/14/2014 05:18 AM - h.shirosaki (Hiroshi Shirosaki)

<b>Status:</b>	Closed	
<b>Priority:</b>	Normal	
<b>Assignee:</b>	nobu (Nobuyoshi Nakada)	
<b>Target version:</b>	2.2.0	
<b>ruby -v:</b>	ruby 2.2.0dev (2014-04-13 trunk 45580) [x64-mingw32]	<b>Backport:</b> 2.0.0: DONE, 2.1: DONE

#### Description

I have the following two test failures with x64-mingw32 on Windows 7.

```
$ gcc -v 2>&1 | tail -n1
gcc version 4.8.2 20130712 (prerelease) (Built by MinGW-builds project)
$ make test-all TESTS="-q ruby/test_exception.rb"
```

(snip)

```
[29/43] TestException#test_machine_stackoverflow = 0.05 s
```

1) Failure:

```
TestException#test_machine_stackoverflow [c:/Users/h.shirosaki/work/ruby/test/ruby/test_exception.
rb:537]:
pid 2228 killed by SIGSEGV (signal 11)
```

```
[30/43] TestException#test_machine_stackoverflow_by_define_method = 0.05 s
```

2) Failure:

```
TestException#test_machine_stackoverflow_by_define_method [c:/Users/h.shirosaki/work/ruby/test/rub
y/test_exception.rb:54
8]:
pid 8984 killed by SIGSEGV (signal 11)
```

```
Finished tests in 0.575033s, 74.7783 tests/s, 370.4135 assertions/s.
43 tests, 213 assertions, 2 failures, 0 errors, 0 skips
```

```
ruby -v: ruby 2.2.0dev (2014-04-13 trunk 45580) [x64-mingw32]
```

I'm not sure the reason, but setting larger stack size fixes SIGSEGV.

```
diff --git a/configure.in b/configure.in
index f52e0ba..078f9ba 100644
--- a/configure.in
+++ b/configure.in
@@ -2782,7 +2782,7 @@ if test "$with_dln_a_out" != yes; then
     rb_cv_dlopen=yes],
   [cygwin*|mingw*], [
     : ${LD_SHARED='${CC} -shared $(if $(filter-out -g -g0,$(debugflags)),,-s)'}
-   XLDFLAGS="${XLDFLAGS} -Wl,--stack,0x00200000,--enable-auto-import"
+   XLDFLAGS="${XLDFLAGS} -Wl,--stack,0x01200000,--enable-auto-import"
     DLDFLAGS="${DLDFLAGS} -Wl,--enable-auto-image-base,--enable-auto-import"
     : ${LIBPATHENV=""}
     rb_cv_dlopen=yes],
```

#### Associated revisions

Revision c22cbdc1 - 06/23/2014 03:08 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) r45947,r45951: [Backport #9739] [Backport #9844]

```
* thread_win32.c (rb_w32_stack_overflow_handler): use Structured
Exception Handling by Addvectoredexceptionhandler() for machine
stack overflow on mingw.
This would be equivalent to the handling using __try and __except
on mswin introduced by r43748.
```

Exception Handling by AddVectoredExceptionHandler() for machine  
This would be equivalent to the handling using \_\_try and \_\_except

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_1@46522 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 46522 - 06/23/2014 03:08 PM - nagachika (Tomoyuki Chikanaga)**

merge revision(s) r45947,r45951: [Backport #9739] [Backport #9844]

\* thread\_win32.c (rb\_w32\_stack\_overflow\_handler): use Structured Exception Handling by AddVectoredExceptionHandler() for machine stack overflow on mingw.  
This would be equivalent to the handling using \_\_try and \_\_except on mswin introduced by r43748.  
Exception Handling by AddVectoredExceptionHandler() for machine  
This would be equivalent to the handling using \_\_try and \_\_except

**Revision 2c1a9c3d - 06/27/2014 08:03 AM - usa (Usaku NAKAMURA)**

merge revision(s) 43748,45947,45951: [Backport #9739]

\* eval\_intern.h (SAVE\_ROOT\_JMPBUF): workaround for the failure of test/ruby/test\_exception.rb on Windows.  
wrap by \_\_try and \_\_exception statements on mswin to raise SIGSEGV when EXCEPTION\_STACK\_OVERFLOW is occurred, because MSVCRT doesn't handle the exception.  
however, (1) mingw-gcc doesn't support \_\_try and \_\_exception statements, and (2) we cannot retry SystemStackError after this change yet (maybe crashed) because SEH and longjmp() are too uncongential.

\* signal.c (check\_stack\_overflow, CHECK\_STACK\_OVERFLOW): now defined on Windows, too.

\* thread\_win32.c (ruby\_stack\_overflowed\_p): ditto.

\* thread\_win32.c (rb\_w32\_stack\_overflow\_handler): use Structured Exception Handling by AddVectoredExceptionHandler() for machine stack overflow on mingw.  
This would be equivalent to the handling using \_\_try and \_\_except on mswin introduced by r43748.  
Exception Handling by AddVectoredExceptionHandler() for machine  
This would be equivalent to the handling using \_\_try and \_\_except

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_2\_0\_0@46576 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

**Revision 46576 - 06/27/2014 08:03 AM - usa (Usaku NAKAMURA)**

merge revision(s) 43748,45947,45951: [Backport #9739]

\* eval\_intern.h (SAVE\_ROOT\_JMPBUF): workaround for the failure of test/ruby/test\_exception.rb on Windows.  
wrap by \_\_try and \_\_exception statements on mswin to raise SIGSEGV when EXCEPTION\_STACK\_OVERFLOW is occurred, because MSVCRT doesn't handle the exception.  
however, (1) mingw-gcc doesn't support \_\_try and \_\_exception statements, and (2) we cannot retry SystemStackError after this change yet (maybe crashed) because SEH and longjmp() are too uncongential.

\* signal.c (check\_stack\_overflow, CHECK\_STACK\_OVERFLOW): now defined on Windows, too.

\* thread\_win32.c (ruby\_stack\_overflowed\_p): ditto.

\* thread\_win32.c (rb\_w32\_stack\_overflow\_handler): use Structured Exception Handling by AddVectoredExceptionHandler() for machine stack overflow on mingw.  
This would be equivalent to the handling using \_\_try and \_\_except on mswin introduced by r43748.  
Exception Handling by AddVectoredExceptionHandler() for machine  
This would be equivalent to the handling using \_\_try and \_\_except

---

**History**

**#1 - 05/11/2014 10:59 PM - luislavena (Luis Lavena)**

- Category set to build
- Status changed from Open to Assigned
- Assignee set to nobu (Nobuyoshi Nakada)
- Target version set to 2.2.0

Nobu,

Can you review, reject or approve so we can resolve these test failures?

Thank you.

**#2 - 05/15/2014 03:13 AM - h.shirosaki (Hiroshi Shirosaki)**

- File 0001-Handle-machine-stack-overflow-on-mingw.patch added

I found this fix was provided on mswin at [r43748](#), but not on mingw.

We can use Structured Exception Handling by `Addvectoredexceptionhandler()` for machine stack overflow on mingw. This would be equivalent to the handling using `__try` and `__except` on mswin introduced by [r43748](#).

Attached a patch.

**#3 - 05/15/2014 12:43 PM - h.shirosaki (Hiroshi Shirosaki)**

- Category changed from build to platform/mingw
- Status changed from Assigned to Closed

Applied at [r45947](#). Thank you.

**#4 - 06/23/2014 03:08 PM - nagachika (Tomoyuki Chikanaga)**

- Backport changed from 2.0.0: UNKNOWN, 2.1: UNKNOWN to 2.0.0: UNKNOWN, 2.1: DONE

Backported into ruby\_2\_1 branch at r46522.

**#5 - 06/27/2014 08:04 AM - usa (Usaku NAKAMURA)**

- Backport changed from 2.0.0: UNKNOWN, 2.1: DONE to 2.0.0: DONE, 2.1: DONE

backported [r43748](#) (for mswin), [r45947](#) and [r45951](#) (for mingw) into ruby\_2\_0\_0 at r46576.

**#6 - 09/03/2014 08:55 AM - vo.x (Vit Ondruch)**

Interestingly, I met this test error recently on Fedora's builder:

```
3) Failure:
TestException#test_machine_stackoverflow_by_define_method [/build/buildd/build/BUILD/ruby-2.2.0-r47288/test/ruby/test_exception.rb:555]:
pid 19431 killed by SIGSEGV (signal 11)
15620 tests, 2470788 assertions, 1 failures, 0 errors, 31 skips
ruby -v: ruby 2.2.0dev (2014-08-26 trunk 47288) [i386-linux]
```

**Files**

---

0001-Handle-machine-stack-overflow-on-mingw.patch	1.89 KB	05/15/2014	h.shirosaki (Hiroshi Shirosaki)
---	---------	------------	---------------------------------