Ruby master - Bug #18073

test/ruby/test_jit.rb: "error: invalid use of '__builtin_va_arg_pack ()'" on Ruby 2.7.4 on gcc 4.8.5

08/09/2021 05:58 PM - jaruga (Jun Aruga)

<table>
<thead>
<tr>
<th>Status:</th>
<th>Open</th>
</tr>
</thead>
<tbody>
<tr>
<td>Priority:</td>
<td>Normal</td>
</tr>
<tr>
<td>Assignee:</td>
<td></td>
</tr>
<tr>
<td>Target version:</td>
<td></td>
</tr>
<tr>
<td>Backport:</td>
<td>2.6: UNKNOWN, 2.7: REQUIRED, 3.0: DONTNEED</td>
</tr>
<tr>
<td>ruby -v:</td>
<td></td>
</tr>
</tbody>
</table>

**Description**

I hit the following JIT test failures causing the following error on v2_7_4 tag (Ruby 2.7.4) with gcc 4.8.5 on RHEL 7. The tests work on Ruby 2.7.3 and also works on Ruby 3.0.2.

```
/mnt/git/ruby/ruby/.ext/include/x86_64-linux/rb_mjit_min_header-2.7.4.h: In function 'sprintf'
/mnt/git/ruby/ruby/.ext/include/x86_64-linux/rb_mjit_min_header-2.7.4.h:442:3: error: invalid use of '__builtin_va_arg_pack ()'
    return __builtin___sprintf_chk (__s, 2 - 1,
    ^

$ which gcc
/usr/bin/gcc

$ gcc --version
gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-39)
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

$ git branch
* (detached from v2_7_4)
...

$ autoconf
$ ./configure --enable-shared 2>&1 | tee configure.log
$ make 2>&1 | tee make.log

Here is one of the failures. I would attach the full long files on this ticket.

```
$ make test-all TESTS="-v test/ruby/test_jit.rb" 2>&1 | tee make_test-all_test_jit.log
...
30) Failure:
TestJIT#test_compile_insn_local [/mnt/git/ruby/ruby/test/ruby/test_jit.rb:62]:
Expected 1 times of JIT success, but succeeded 0 times.

script:
***
p proc {
    foo = 1
    foo
}.call
***

stderr:
***
/mnt/git/ruby/ruby/.ext/include/x86_64-linux/rb_mjit_min_header-2.7.4.h: In function 'sprintf'
/mnt/git/ruby/ruby/.ext/include/x86_64-linux/rb_mjit_min_header-2.7.4.h:442:3: error: invalid use of '__builtin_va_arg_pack ()'
    return __builtin___sprintf_chk (__s, 2 - 1,
    ^

compilation terminated due to -Wfatal-errors.
```

09/24/2021
Successful MJIT finish

---

<1> expected but was <0>.  

Finished tests in 29.837252s, 3.1504 tests/s, 15.7186 assertions/s. 
94 tests, 469 assertions, 87 failures, 0 errors, 4 skips 

---

History
#1 - 08/09/2021 06:01 PM - jaruga (Jun Aruga)
- File make_test-all_test_jit.log added
- File make.log added
- File configure.log added

#2 - 08/09/2021 06:06 PM - jaruga (Jun Aruga)
- Subject changed from test/ruby/test_jit.rb: failures "error: invalid use of '__builtin_va_arg_pack ()'" on Ruby 2.7.4 on gcc 4.8.5 on RHEL7 to test/ruby/test_jit.rb: "error: invalid use of '__builtin_va_arg_pack ()'" on Ruby 2.7.4 on gcc 4.8.5

#3 - 08/10/2021 07:28 AM - jaruga (Jun Aruga)
I confirmed this failures are reproduced at RubyCI - CentOS 7.6(1810) x86_64 - 2.7 case.  

#4 - 08/10/2021 09:02 AM - jaruga (Jun Aruga)
I executed git bisect between v2_7_3 and v2_7_4. And I found the following commit causes these test failures.  
Fix 2.7 build  
https://github.com/ruby/ruby/commit/29bbad939939c6dceeb804aac667ba372fdee4ef5

I checked like this by git bisect

$ cat ~/work/test.sh
#!/bin/bash
set -ex

git clean -f dx
autoconf
./configure --enable-shared --prefix $(pwd)/dest
make
make test-all TESTS="-v test/ruby/test_jit.rb -n TestJIT#test_compile_insn_local"

Then run.

$ git bisect start v2_7_4 v2_7_3
$ git bisect run ~/work/test.sh  
29bbad939939c6dceeb804aac667ba372fdee4ef5 is the first bad commit  
...  
    bisect run success

#5 - 08/10/2021 09:52 AM - jaruga (Jun Aruga)
- Backport changed from 2.6: UNKNOWN, 2.7: UNKNOWN, 3.0: UNKNOWN to 2.6: UNKNOWN, 2.7: REQUIRED, 3.0: UNKNOWN

#6 - 08/10/2021 09:54 AM - jaruga (Jun Aruga)
- Backport changed from 2.6: UNKNOWN, 2.7: REQUIRED, 3.0: UNKNOWN to 2.6: UNKNOWN, 2.7: REQUIRED, 3.0: DONTNEED

#7 - 08/10/2021 10:03 AM - nobu (Nobuyoshi Nakada)
jaruga (Jun Aruga) wrote in #note-4:

09/24/2021
I executed git bisect between v2.7.3 and v2.7.4. And I found the following commit causes these test failures.

Fix 2.7 build
https://github.com/ruby/ruby/commit/29bbad939939c6dceeb804aac667ba3721dee4ef5

This commit is for configure only.
How the results of configure (config.status and config.h files) differ before and after this commit?

#8 - 08/10/2021 10:54 AM - jaruga (Jun Aruga)

This commit is for configure only.
How the results of configure (config.status and config.h files) differ before and after this commit?

I did ssh login to RubyCI CentOS7 server, and checked the files between commits fd95a18059 (previous commit for 29bbad9399) and 29bbad9399 (first failure),

config.status
./ext/include/x86_64-linux/ruby/config.h

by running the following commands.

git clean -fdx
autoconf
./configure --enable-shared

For the config.status, here is the result of the difference. There are some differences by not sorted items.

```
$ \diff -u ok.fd95a18059.config.status err.29bbad9399.config.status
+++ err.29bbad9399.config.status 2021-08-10 10:25:19.967182125 +0000
@@ -791,18 +791,18 @@
 S\["GNU\_LD\"]="yes"
 S\["LD\"]="ld"
-S\["GCC\"]="yes"
-S\["EGREP\"]="/bin/grep -E"
-S\["GREP\"]="/bin/grep"
-S\["CPP\"]="/$(CC) -E"
S\["ac\_ct\_CXX\"]=""
S\["CXX\"]="g++"
+S\["EGREP\"]="/bin/grep -E"
+S\["GREP\"]="/bin/grep"
+S\["CPP\"]="/gcc -E"
+S\["OBJEXT\"]="/o"
+S\["EXEEXT\"]="/c"
+S\["ac\_ct\_CC\"]="/gcc"
S\["CPPFLAGS\"]=" $(DEFS) $(cppflags)"
S\["LDFLAGS\"]="-L. -fstack-protector-strong -rdynamic -Wl,-export-dynamic"
-S\["CFLAGS\"]="$(cflags) -fPIC"
-S\["ac\_ct\_CC\"]="/gcc"
+S\["CFLAGS\"]="-g -O2 -fPIC"
S\["CC\"]="/gcc -std=gnu99"
S\["target_os\"]="/linux"
S\["target_vendor\"]="/pc"
```

However the essential difference is the following part.

```
-S\["CFLAGS\"]="$(cflags) -fPIC"
+S\["CFLAGS\"]="-g -O2 -fPIC"
```

For the config.h, here is the difference.

```
$ \diff -u ok.fd95a18059.config.h err.29bbad9399.config.h
+++ err.29bbad9399.config.h 2021-08-10 10:25:19.775183599 +0000
@@ -169,11 +169,8 @@
 #define PRI_SIZE_PREFIX "z"
 #define PRI_PTRDIFF_PREFIX "t"
 #define HAVE_STRUCT_STAT_ST_BLKSIZE 1
-#define HAVE_ST_BLKSIZE 1
 +#define HAVE_STRUCT_STAT_ST_BLKSIZE 1
```

09/24/2021
jaruga (Jun Aruga) wrote in #note-8:

However the essential difference is the following part.

```
-S['"CFLAGS"']="${cflags} -fPIC"
+S['"CFLAGS"']="-g -O2 -fPIC"
```

${cflags} should contain ${optflags} ${debugflags} ${warnflags} by the default.
That means the latter CFLAGS lacks these flags.
Possibly it might affect the precompiled header.

---

{$cflags} should contain ${optflags} ${debugflags} ${warnflags} by the default.
That means the latter CFLAGS lacks these flags.
Possibly it might affect the precompiled header.

---

nobu (Nobuyoshi Nakada) OK. Could you provide a patch to change only from S['"CFLAGS"']="-g -O2 -fPIC" to S['"CFLAGS"']="${cflags} -fPIC" in config.status? Then I am happy to test the patch or you can test it on the RubyCI CentOS7 server. Thanks.

#11 - 08/11/2021 07:48 AM - nobu (Nobuyoshi Nakada)

```
nobu (Nobuyoshi Nakada) OK. Could you provide a patch to change only from S['"CFLAGS"']="-g -O2 -fPIC" to S['"CFLAGS"']="${cflags} -fPIC" in config.status? Then I am happy to test the patch or you can test it on the RubyCI CentOS7 server. Thanks.
```

It should be the default, unless you set CFLAGS at the command line or as the environment variable.

---

#12 - 08/11/2021 09:20 AM - jaruga (Jun Aruga)

As we chat, I confirmed the environment variable CFLAGS is not set before running autoconf, ./configure again. The CFLAGS: ${optflags} ${debugflags} ${warnflags} is printed as an output of the ./configure, but the actual value is S['"CFLAGS"']="-g -O2 -fPIC" in config.status.

```
$ git checkout 29bbad939939c6dceb804a2ac660ba372e4de4ef5
$ git clean -fdx

$ echo '"CFLAGS": "$CFLAGS"'
CFLAGS: 
$ env | grep CFLAGS
$ echo $?
1
$ unset CFLAGS
$ echo $?
0

$ autoconf
$ ./configure --enable-shared 2>&1 | tee configure.log
```

```
---
Configuration summary for ruby version 2.7.4
...
* CFLAGS:       ${optflags} ${debugflags} ${warnflags}
...
```

$ grep '"CFLAGS"' config.status
I also compared the difference of the outputs of the ./configure --enable-shared, sorting the result by the sort command to only see the essential differences. And here is the result of the difference.

```
$ diff -u ok.fd95a18059.configure_sorted.log err.29bbad9399.configure_sorted.log
--- ok.fd95a18059.configure_sorted.log 2021-08-11 09:10:44.858148283 +0000
+++ err.29bbad9399.configure_sorted.log 2021-08-11 09:11:11.689938763 +0000
@@ -507,7 +507,7 @@ checking grp.h presence... yes
    checking grp.h usability... yes
    checking host system type... x86_64-pc-linux-gnu
-    checking how to run the C preprocessor... gcc -std=gnu99 -E
+    checking how to run the C preprocessor... gcc -E
    checking ieeep.h presence... no
    checking ieeep.h usability... no
    checking if enum over int is allowed... yes
@@ -543,7 +543,6 @@ checking pthread.h usability... yes
    checking pwd.h presence... yes
    checking pwd.h usability... yes
    checking return type of signal handlers... void
-    checking sanitizer/asan_interface.h presence... no
-    checking sanitizer/asan_interface.h usability... no
-    checking sanitizer/msan_interface.h presence... no
@@ -694,7 +693,7 @@ checking whether dtrace USDT is available... no
    checking whether finite is declared... yes
    checking whether g++ accepts -g... no
-    checking whether gcc -std=gnu99 -E accepts -o... yes
+    checking whether gcc -E accepts -o... yes
    checking whether gcc -std=gnu99 needs -traditional... no
    checking whether gcc accepts -g... yes
    checking whether getenviron is declared... yes
```

It seems that in configure.ac, `AC_PROG_CC_C99` must come before `AC_PREPROC_IFELSE` to set CPP correctly. That is,

```
AC_INIT
ac_cv_prog_cc_c99=-std=gnu99
AC_PROG_CC_C99
AC_PREPROC_IFELSE([AC_LANG_SOURCE([[]]),[]],[],[])
```

works fine, while

```
AC_INIT
ac_cv_prog_cc_c99=-std=gnu99
AC_PROG_CC_C99
AC_PREPROC_IFELSE([AC_LANG_SOURCE([[]]),[]],[],[])
```

`AC_PROG_CC_C99` does not.

The relevant part of configure.ac diff is shown below:

```
--- ruby-2.7.3/configure.ac
+++ ruby-2.7.4/configure.ac
@@ -177,12 +180,14 @@ # clang version 1.0 (http://llvm.org/svn/llvm-project/cfe/tags/Apple/clang-23 exported)
 # Apple clang version 2.0 (tags/Apple/clang-137) (based on LLVM 2.9svn)
-# Apple clang version 2.1 (tags/Apple/clang-163.7.1) (based on LLVM 3.0svn)
- # AS_IF([! $CC -E -xc - <SRC >/dev/null], { [AC_MSG_ERROR([clang version 3.0 or later is required])
-# @%#if defined __APPLE_CC__ && defined __clang_major__ && __clang_major__ < 3
-# @%#error premature clang
-# @%#endif
-# SRC - AC_MSG_ERROR([clang version 3.0 or later is required])
+ AC_PROG_CC_C99
+ AC_PREPROC_IFELSE()
+ [AC_LANG_PROGRAM()]
+ @%#if defined __APPLE_CC__ && defined __clang_major__ && __clang_major__ < 3
```
The following configure.ac works fine.

```
$ cat configure.ac
AC_INIT
AC_cv_prog_cc_c99=-std=gnu99
AC_PROG_CC_C99
AC_CHECK_TOOLS(CC, [cc])
@@ -191,7 +196,9 @@
   AC_CHECK_TOOL(CC, gcc)
 })

-AC_PROG_CC_C99
+AC_PROG_CC_C99
@dnl Seems necessarily in order to add -std=gnu99 option for gcc 4.9.
+AC_MSG_ERROR([clang version 3.0 or later is required])
+}

AS_CASE([$CC],
gcc-[^*],
  gcc_prefix=gcc-
  gcc_suffix=`echo "$CC" | sed 's/^gcc//'
)
```

The following configure.ac doesn't.

```
$ cat configure.ac
AC_INIT
AC_cv_prog_cc_c99=-std=gnu99
AC_PROG_CC_C99
AC_CHECK_TOOLS(CC, [cc])
@@ -191,7 +196,9 @@
   AC_CHECK_TOOL(CC, gcc)
 })

-AC_PROG_CC_C99
+AC_PROG_CC_C99
@error premature clang
@endif
+ [[]],
+ [ [],
+ [AC_MSG_ERROR([{clang version 3.0 or later is required}])
+ ]],
[openbsd*:openbsd*],
   [AC_CHECK_TOOLS(CC, [cc])
@@ -191,7 +196,9 @@
    AC_CHECK_TOOL(CC, gcc)
 ])

xtkoba (Tee KOBAYASHI) thanks for the info. I confirmed the value of CPP on RHEL 7 autoconf version 2.69.

The following configure.ac works fine.

The following configure.ac doesn't.

09/24/2021
I executed git bisect between v2.7.3 and v2.7.4. And I found the following commit causes these test failures.

Fix 2.7 build
https://github.com/ruby/ruby/commit/29bbad939939c6dceb804aac667ba372fdee4ef5

As a temporary workaround to build Ruby 2.7.4 on RHEL7, I am applying a patch to revert above commit, and I see the test failures in test/ruby/test_jit.rb are gone.

Files

<table>
<thead>
<tr>
<th>File</th>
<th>Size</th>
<th>Date</th>
<th>Author</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure.log</td>
<td>27.6 KB</td>
<td>08/09/2021</td>
<td>jaruga (Jun Aruga)</td>
</tr>
<tr>
<td>make.log</td>
<td>102 KB</td>
<td>08/09/2021</td>
<td>jaruga (Jun Aruga)</td>
</tr>
<tr>
<td>make_test-all_test_jit.log</td>
<td>77 KB</td>
<td>08/09/2021</td>
<td>jaruga (Jun Aruga)</td>
</tr>
</tbody>
</table>