Ruby trunk - Bug #13548

miniruby SEGV while building with non-default CFLAGS (caused by __builtin_setjmp)

05/08/2017 03:53 PM - vp (Vladimir Pavlov)

Status:	Feedback		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:		**Backport:**	2.2: UNKNOWN, 2.3: UNKNOWN, 2.4: UNKNOWN

Description

Trying to build ruby-2.4.1 using gcc-5.4.0 I get the attached error. Just-released gcc-7.1.0 causes the similar error. Old gcc-4.8.4 builds successfully.

The build command is:

```
CFLAGS="-O1 -fweb -g" LDFLAGS=-g ./configure --enable-shared --enable-load-relative && make clean && MAKE="make V=1" make -j1 V=1
```

Digging into ruby sources leaded me to conclusion the issue is caused by (probably incorrect) using __builtin_setjmp() .

But the links below
https://www.securecoding.cert.org/confluence/display/c/MSC22-C.+Use+the+setjmp%28%29%2C+longjmp%28%29+facility+securely
https://en.wikipedia.org/wiki/Setjmp.h
http://pubs.opengroup.org/onlinepubs/9699919799/functions/setjmp.html
get me to think you use setjmp in unportable way. Particularly, you assign the result of setjmp() call to a variable.

afaiu "library" versions of setjmp/longjmp don't require the code to be fully standard-compliant and might work even if the code is incorrect. But when using _builtin* versions the compiler hardly relies on standard compliance and treats ruby code as subjected to undefined behavior (gcc likes to behave like that last few years, even linux kernel used to have bugs appearing when building with newer gcc versions).

I would report the issue to gcc bugzilla too but they like "short piece of code to reproduce" that I failed to create. I don't know whether it's a bug in ruby or in gcc.

Please, try to fix the code if possible so it works with newer gcc version and _builtin-variants of setjmp/longjmp.

Right now adding --with-setjmp-type=setjmp to configure fixes the build.

P.S. From source code I understand it would require to rewrite huge parts of ruby to fix the issue, so I don't expect you fix that in the foreseeable future. Just decided to leave a note here.

History

#1 - 05/08/2017 04:07 PM - vp (Vladimir Pavlov)

Forgot to say, the same happens too with more popular

```
CFLAGS="-O3 -funroll-loops -g"
```

And the cause of the issue (in the backtrace attached) is not rb_str_buf_cat2+0x39, but is rb_require_internal+0x6ba (or even lower).

#2 - 05/19/2017 08:08 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Feedback

vp (Vladimir Pavlov) wrote:

> Particularly, you assign the result of setjmp() call to a variable.

It is not assigned since r43522.

Maybe newer __builtin_setjmp has changed the format of saved registers.

#3 - 06/02/2017 09:43 PM - vp (Vladimir Pavlov)

nobu (Nobuyoshi Nakada) wrote:

> It is not assigned since r43522.

I'm sorry. What I wrote was not what I meant.

The opengroup docs say setjmp should be used as "entire controlling expression" (with slight possible modifications), not a part of a controlling expression. The current implementation sometimes lead to expressions like

```
// original
if ((state = EXEC_TAG()) == 0)

// preprocessed
if ((state = setjmp(_th->tag->buf) ? ruby_threadptr_tag_state(_th) : 0) == 0)
```

where setjmp() is obviously not an entire controlling expression.

I tried to fix that (not so hard) and made all variables volatile in functions calling setjmp, but without a success.

So I guess it's a gcc bug (unless I failed again).

Files

builtin_setjmp.txt	8.98 KB	05/08/2017	vp (Vladimir Pavlov)