Ruby - Bug #1342

signal handling on HP-UX

04/01/2009 02:00 AM - graza (Graham Agnew)

Status:	Rejected		
Priority:	Normal		
Assignee:	naruse (Yui NARUSE)		
Target version:	2.6		
ruby -v:	ruby 1.9.1p0 (2009-01-30 revision 21907) [ia64-hpux11.23]	**Backport:**	

Description

=begin
Whenever I interrupt ruby on HP-UX 11i v2, I get a message from the operating system about an inability to establish context and a core dump. This is the messages:

> sendsig: useracc failed. 0x9fffffffbf7dae00 0x00000000005000
>
> Pid 3044 was killed due to failure in writing the signal context - possible stack overflow.
> Illegal instruction

Looking at the stack backtrace in the core file shows the following:

```
HP gdb 5.4.0 for HP Itanium (32 or 64 bit) and target HP-UX 11.2x.
Copyright 1986 - 2001 Free Software Foundation, Inc.
Hewlett-Packard Wildebeest 5.4.0 (based on GDB) is covered by the
GNU General Public License. Type "show copying" to see the conditions to
change it and/or distribute copies. Type "show warranty" for warranty/support.
..
Core was generated by `ruby'.
Program terminated with signal 4, Illegal instruction.
ILL_ILLOPC - Illegal Op-Code
#0  0xc00000000033a990:0 in __ksleep+0x30 () from /usr/lib/hpux64/libc.so.1
.gdbinit:2: Error in sourced command file:
No symbol "dummy_gdb_enums" in current context.
(gdb) ba
#0  0xc00000000033a990:0 in __ksleep+0x30 () from /usr/lib/hpux64/libc.so.1
#1  0xc0000000001280a0:0 in __mxn_sleep+0xae0 ()
from /usr/lib/hpux64/libpthread.so.1
#2  0xc0000000000c0f90:0 in <unknown_procedure> + 0xc50 ()
from /usr/lib/hpux64/libpthread.so.1
#3  0xc0000000000c1e30:0 in pthread_cond_timedwait+0x1d0 ()
from /usr/lib/hpux64/libpthread.so.1
warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance
```

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

warning: Cannot insert inlined instance

#4 0x40000000002f5db0:0 in native_cond_timedwait () at thread_pthread.c:123
#5 0x40000000002f7aa0:0 in thread_timer () at thread_pthread.c:756
#6 0xc0000000000cf3c0:0 in __pthread_bound_body+0x190 ()
from /usr/lib/hpux64/libpthread.so.1
(gdb)
=end

History

#1 - 04/01/2009 07:45 AM - graza (Graham Agnew)

=begin
I've been looking through the Ruby source code, specifically the Itanium specific code wrapped in "#ifdef __ia64" guards and within the assembly file ia64.s. While I can follow the references to the Intel documentation, it seems that the Itanium code is there to find the position of the register stack. There's also rb_ia64_flushrs in conjunction with setjmp() inside the function rb_gc_save_machine_context.

However looking at the HP documentation, it seems that setjmp an longjmp are not suitable for saving context. Instead getcontext and setontext should be used:

http://h21007.www2.hp.com/portal/site/dspp/menuitem.863c3e4cbcdc3f3515b49c108973a801?ciid=09083a7373f021103a7373f02110275d6e10RCRD

According to the referenced documents, this only applies when performing longjmp across threads, and I can't find any cases in the code where this is happening. At the same time, since setcontext and getcontext seem to be fairly widely available, shouldn't the source code be switched to use those? They seem to be more appropriate for managing context.
=end

#2 - 04/01/2009 08:14 AM - daz (Dave B)

=begin
"... since setcontext and getcontext seem to be fairly widely available, shouldn't the source code be switched to use those?"

http://en.wikipedia.org/wiki/Setcontext

Unknown to Windows (not found in SDK docs).

=end

#3 - 04/01/2009 08:47 AM - graza (Graham Agnew)

=begin
Hi Dave,

Granted this won't be available everywhere, however it remains that, setjmp and longjmp are not necessarily appropriate in HP-UX. The man page for setjmp/longjmp on HP-UX has the following:

> The effect of a call to longjmp() where the initialization of the jmp_buf argument was not performed in the calling thread is undefined.

So where available, shouldn't the getcontext / setcontext routines be used?

Cheers,
Gra.
=end

#4 - 04/02/2009 04:17 AM - nobu (Nobuyoshi Nakada)

=begin
Hi,

At Wed, 1 Apr 2009 07:45:15 +0900,
Graham Agnew wrote in [ruby-core:23086]:

> I've been looking through the Ruby source code, specifically
> the Itanium specific code wrapped in "#ifdef __ia64" guards
> and within the assembly file ia64.s. While I can follow the
> references to the Intel documentation, it seems that the
> Itanium code is there to find the position of the register
> stack. There's also rb_ia64_flushrs in conjunction with
> setjmp() inside the function rb_gc_save_machine_context.

Is it an ia64 specific issue?

> According to the referenced documents, this only applies when
> performing longjmp across threads, and I can't find any cases
> in the code where this is happening. At the same time, since
> setcontext and getcontext seem to be fairly widely available,
> shouldn't the source code be switched to use those? They
> seem to be more appropriate for managing context.

It shouldn't jump across threads. And getcontext/setcontext
has significant performance penalty than setjmp/longjmp.

If it is ia64 specific, getcontext/setcontext should be used on
such platforms.

--
Nobu Nakada

=end

#5 - 04/02/2009 08:15 AM - graza (Graham Agnew)

=begin
Hi Nakada-san,

The only other environment I've tried so far is AIX and I haven't seen this issue there at all. (But you probably knew that since you responded to my other issue on the Ruby forum. :)) This problem only happens on HP-UX/Itanium, not AIX.

Just as a bit of background, I am looking to integrate Ruby with a product sold by my company, so eventually I will also be compiling for HP-UX/PA-RISC and Solaris/SPARC. The product is 64-bit only on the Unix server side and 32-bit on the Windows client side. Hopefully I won't see this issue there.

In the HP-UX articles referenced above, The following comment is made in the second paper with regard to the assembly code included in the first paper:

> While the assembly code is useful for performance sensitive implementations, it is not portable to
> other architectures and requires a significant understanding of the Itanium calling conventions. This
> document extends the previous paper and the man pages by providing example HP-UX C-level source
> code to implement user level thread switching with the context library routines. These routines are
> more portable among releases of HP-UX and can be employed by software engineers.

If performance is an issue, then perhaps the assembly from the first paper is useful. Otherwise, getcontext/setcontext would seem more portable.

Cheers,
Gra.
=end

#6 - 04/02/2009 08:25 AM - graza (Graham Agnew)

=begin
Hi Nakada-san,

I should also say that this problem is worse on HP-UX 11i v3 (version 11.31). When running "make test" it doesn't even get past the sample/test.rb:signal tests; ruby core dumps with the same problem of establishing context.

Cheers,
Gra.
=end

#7 - 04/04/2009 11:15 AM - graza (Graham Agnew)

=begin
Hi Nakada-san,

I have modified my 1.9.1-p0 such that getcontext/setcontext would be used, but it hasn't helped. Basically this was done by running configure as normal and then changing the generated .ext/include/ia64-hpux11.23/ruby/config.h to have the following:

```
#define RUBY_SETJMP(env) ( env->value = 0, getcontext(&env->context), env->value )
#define RUBY_LONGJMP(env,val) ( env->value = val, setcontext(&env->context) )
typedef struct {
ucontext_t context;
int value;
} RUBY_JMP_BUF[1];
```

I had tp change one or two other places to get it to compile but after that everything compiles OK, and I think the context is being successfully saved and restored. However I'm still getting the same sort of errors as above. So it looks like this isn't the answer.

I've googled this error and the only other meaningful reference to this is that there was a bug in the Java VM for HP-UX. I don't know how to diagnose this problem further or what to try next.

Cheers,
Gra.
=end

#8 - 04/08/2009 08:59 PM - graza (Graham Agnew)

=begin
Some progress on this:

In the HP-UX documentation it says that on Itanium, PTHREAD_STACK_MIN is 256KB. But when I tracked the actual value doen in (limits.h), I found that it was only 4KB. Increasing this has solved the problem described in this ticket, however the test suite is now getting quite a few segmentation violation faults (SIGSEGV).

So it's not solved just yet.
=end

#9 - 04/08/2009 09:59 PM - graza (Graham Agnew)

=begin
OK, so the problem with Segmentation faults was related to the previous changes I had made to use getcontext/setcontext instead of setjmp/lonjmp; it was causing Fibers to fail for one thing, and who knows what else. Once I reverted back to setjmp/lonjmp all but one of the tests pass. The failing test is as per ticket #1341 - I haven't looked into that much just yet...

```
*** orig/ruby-1.9.1-p0/thread_pthread.c Tue Jan 20 09:53:14 2009
--- ruby-1.9.1-p0/thread_pthread.c      Wed Apr  8 13:53:08 2009
```

```
*** 17,22 ****
--- 17,27 ----
#include <sys/resource.h>
#endif
```

- #ifdef __hpux
- #undef PTHREAD_STACK_MIN
- #define PTHREAD_STACK_MIN 0x80000
- #endif
- static void native_mutex_lock(pthread_mutex_t *lock);
  static void native_mutex_unlock(pthread_mutex_t *lock);
  static int native_mutex_trylock(pthread_mutex_t *lock);

=end

#10 - 07/14/2009 12:25 AM - yugui (Yuki Sonoda)

- Priority changed from Normal to 3

=begin

=end

#11 - 10/21/2009 08:33 PM - naruse (Yui NARUSE)

- Status changed from Open to Assigned

- Assignee set to kanemoto (Yutaka Kanemoto)

=begin

=end

#12 - 10/21/2009 08:33 PM - naruse (Yui NARUSE)

- Status changed from Assigned to Open

- Assignee deleted (kanemoto (Yutaka Kanemoto))

=begin
Sorry wrong assignment.
=end

#13 - 05/25/2010 07:23 PM - naruse (Yui NARUSE)

- Target version changed from 1.9.1 to 2.0.0

=begin

=end

#14 - 06/26/2011 02:24 PM - naruse (Yui NARUSE)

- Status changed from Open to Feedback

- Assignee set to naruse (Yui NARUSE)

Graham, the patch is still available?
If so, I'll merge it.

#15 - 02/18/2013 09:06 PM - mame (Yusuke Endoh)

- Status changed from Feedback to Rejected

- Target version changed from 2.0.0 to 2.6

Marking as rejected due to no feedback from OP.

--
Yusuke Endoh mame@tsg.ne.jp