

Ruby trunk - Bug #9558

Bus Error on Sparc in r45155 due to `__attribute__((packed))`

02/24/2014 10:26 AM - ngoto (Naohisa Goto)

Status: Closed	
Priority: Normal	
Assignee:	
Target version:	
ruby -v: ruby 2.2.0dev (2014-02-23) [sparc64-solaris2.10]	Backport: 1.9.3: UNKNOWN, 2.0.0: UNKNOWN, 2.1: UNKNOWN
Description Since r45155 , the following error occurs on Sparc Solaris. \$./ruby --disable-gems -e 'p Time.now.utc' -e:1: [BUG] Bus Error at 0xffffffff7ffe775 ruby 2.2.0dev (2014-02-23) [sparc64-solaris2.10] -- Control frame information ----- c:0003 p:---- s:0008 e:000007 CFUNC :utc c:0002 p:0014 s:0005 E:001fb8 EVAL -e:1 [FINISH] c:0001 p:0000 s:0002 E:0006e8 TOP [FINISH] -- Ruby level backtrace information ----- -e:1:in <main> -e:1:inutc' -- Other runtime information ----- <ul style="list-style-type: none">• Loaded script: -e• Loaded features: 0 enumerator.so [NOTE] You may have encountered a bug in the Ruby interpreter or extension libraries. Bug reports are welcome. For details: http://www.ruby-lang.org/bugreport.html Abort This is due to <code>__attribute__((packed))</code> added in r45155 . SPARC (and many other CPUs) can not access to unaligned memory address. Please revert r45155 .	

Associated revisions

Revision 131e4bec - 02/24/2014 11:10 AM - normal

time: only use packed struct on x86*

- `configure.in`: define `PACKED_STRUCT_UNALIGNED` for x86*
- `timev.h` (`struct vtm`): use `PACKED_STRUCT_UNALIGNED`
- `time.c` (`struct time_object`): ditto [Bug #9558] non-x86 cannot safely access unaligned addresses

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@45165 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 45165 - 02/24/2014 11:10 AM - normal

time: only use packed struct on x86*

- `configure.in`: define `PACKED_STRUCT_UNALIGNED` for x86*

- timev.h (struct vtm): use PACKED_STRUCT_UNALIGNED
- time.c (struct time_object): ditto [Bug #9558] non-x86 cannot safely access unaligned addresses

Revision 45165 - 02/24/2014 11:10 AM - normalperson (Eric Wong)

time: only use packed struct on x86*

- configure.in: define PACKED_STRUCT_UNALIGNED for x86*
- timev.h (struct vtm): use PACKED_STRUCT_UNALIGNED
- time.c (struct time_object): ditto [Bug #9558] non-x86 cannot safely access unaligned addresses

Revision 45165 - 02/24/2014 11:10 AM - normal

time: only use packed struct on x86*

- configure.in: define PACKED_STRUCT_UNALIGNED for x86*
- timev.h (struct vtm): use PACKED_STRUCT_UNALIGNED
- time.c (struct time_object): ditto [Bug #9558] non-x86 cannot safely access unaligned addresses

Revision 45165 - 02/24/2014 11:10 AM - normal

time: only use packed struct on x86*

- configure.in: define PACKED_STRUCT_UNALIGNED for x86*
- timev.h (struct vtm): use PACKED_STRUCT_UNALIGNED
- time.c (struct time_object): ditto [Bug #9558] non-x86 cannot safely access unaligned addresses

Revision 45165 - 02/24/2014 11:10 AM - normal

time: only use packed struct on x86*

- configure.in: define PACKED_STRUCT_UNALIGNED for x86*
- timev.h (struct vtm): use PACKED_STRUCT_UNALIGNED
- time.c (struct time_object): ditto [Bug #9558] non-x86 cannot safely access unaligned addresses

Revision 45165 - 02/24/2014 11:10 AM - normal

time: only use packed struct on x86*

- configure.in: define PACKED_STRUCT_UNALIGNED for x86*
- timev.h (struct vtm): use PACKED_STRUCT_UNALIGNED
- time.c (struct time_object): ditto [Bug #9558] non-x86 cannot safely access unaligned addresses

History

#1 - 02/24/2014 11:10 AM - Anonymous

- Status changed from Open to Closed

- % Done changed from 0 to 100

Applied in changeset [r45165](#).

time: only use packed struct on x86*

- configure.in: define PACKED_STRUCT_UNALIGNED for x86*
- timev.h (struct vtm): use PACKED_STRUCT_UNALIGNED
- time.c (struct time_object): ditto [Bug [#9558](#)] non-x86 cannot safely access unaligned addresses

#2 - 02/24/2014 11:19 AM - normalperson (Eric Wong)

ngotogenome@gmail.com wrote:

This is due to `attribute((packed))` added in [r45155](#).
SPARC (and many other CPUs) can not access to unaligned memory address.

Please revert [r45155](#).

Sorry about that, can you try [r45165](#)? Thanks.
I'm only packing the struct on x86* now.