

Ruby 1.8 - Bug #118

SortedSet#inspect does not work with nested SortedSets

06/09/2008 02:56 AM - NoKarma (Arthur Schreiber)

Status:	Closed
Priority:	Normal
Assignee:	knu (Akinori MUSHHA)
Target version:	
ruby -v:	
Description	
=begin SortedSet["a", "b", SortedSet["c"]].inspect # => raises an ArgumentError: comparison of String with SortedSet failed =end	

Associated revisions

Revision 9169 - 09/15/2005 02:33 PM - nahi (Hiroshi Nakamura)

- lib/{soap,wsdl,xsd}, test/{soap,wsdl,xsd}: imported soap4r/1.5.5.

```
#nnn is a ticket number at http://dev.ctor.org/soap4r
```

```
* SOAP
```

```
* allow to configure an envelope namespace of SOAP request. (#124)
  TemporaryNamespace = 'http://www.w3.org/2003/05/soap-envelope'
  @client.options["soap.envelope.requestnamespace"] =
    TemporaryNamespace
  @client.options["soap.envelope.responsenamespace"] =
    TemporaryNamespace
  @client.do_proc(...)
```

```
* let SOAP request XML indent space configurable. see
  "soap.envelope.no_indent" option. (#130)
```

```
* let external CES configurable.
  ex. client["soap.mapping.external_ces"] = 'SJIS'. $KCODE is used
  by default. (#133)
  external CES ::= CES used in Ruby object of client and server
  internal CES ::= CES used in SOAP/OM
```

```
* add iso-8859-1 external CES support. (#106)
```

```
* fixed illegal 'qualified' handling of elements. it caused
  ASP.NET inteoperability problem. (#144)
```

```
* added 'soap.envelope.use_numeric_character_reference' (boolean)
  option to let query XML use numeric character reference in XML,
  not plain UTF-8 character. !GoogleSearch server seems to not
  allow plain UTF-8 character since 2005-08-15 update. (#147)
```

```
* SOAP::Header::SimpleHeader (de)serialization throws an exception
  on !SimpleHeader.on_(in|out)bound when header is a String. so we
  could not use a simple single element headerItem. fixed. thanks
  to emil. (#129)
```

```
* out parameter of rpc operation did not work. (#132)
```

```
* follow HTTP redirect only if using http-access2. (#125) (#145)
```

```
* add a workaround for importing an WSDL whose path begins with
  drive letter. (#115)
```

```
* WSDL
```

- * SOAP Data which is defined as a simpletype was not mapped correctly to Ruby obj when using wsdl2ruby.rb generated classdef file. (#123)
- * rpc/literal support. (#118)
- * re-implemented local element qualify/unqualify control. handles elementFormDefault and form in WSDL. (#119)
- * Array of an element which has simpleType causes a crash. (#128)
- * parameterOrder may not contain return part so it can be shorter than parts size. Thanks to Hugh. (#139)
- * Samples
- * added !BasicAuth client sample. (#117)
- * added Base64 client/server sample.
- * added Flickr SOAP interface client sample. (#122)
- * added !SalesForce client sample. (#135)
- * updated Thawte CA certificate for !GoogleAdWords sample.
- * updated a client script with the newer version made by Johan. thanks!
- * shortened long file names. (#120)
- * fixed typo in authheader sample. (#129)
- * updated deprecated method usage. (#138)

Revision 9169 - 09/15/2005 02:33 PM - nahi (Hiroshi Nakamura)

- lib/{soap,wsdl,xsd}, test/{soap,wsdl,xsd}: imported soap4r/1.5.5.

#nnn is a ticket number at <http://dev.ctor.org/soap4r>

* SOAP

- * allow to configure an envelope namespace of SOAP request. (#124)
TemporaryNamespace = 'http://www.w3.org/2003/05/soap-envelope'
@client.options["soap.envelope.requestnamespace"] =
TemporaryNamespace
@client.options["soap.envelope.responsenamespace"] =
TemporaryNamespace
@client.do_proc(...)
- * let SOAP request XML indent space configurable. see
"soap.envelope.no_indent" option. (#130)
- * let external CES configurable.
ex. client["soap.mapping.external_ces"] = 'SJIS'. \$KCODE is used
by default. (#133)
external CES ::= CES used in Ruby object of client and server
internal CES ::= CES used in SOAP/OM
- * add iso-8859-1 external CES support. (#106)
- * fixed illegal 'qualified' handling of elements. it caused
ASP.NET inteoperability problem. (#144)
- * added 'soap.envelope.use_numeric_character_reference' (boolean)
option to let query XML use numeric character reference in XML,
not plain UTF-8 character. !GoogleSearch server seems to not
allow plain UTF-8 character since 2005-08-15 update. (#147)

- * SOAP::Header::SimpleHeader (de)serialization throws an exception on !SimpleHeader.on_(in|out)bound when header is a String. so we could not use a simple single element headerItem. fixed. thanks to emil. (#129)
- * out parameter of rpc operation did not work. (#132)
- * follow HTTP redirect only if using http-access2. (#125) (#145)
- * add a workaround for importing an WSDL whose path begins with drive letter. (#115)
- * WSDL
- * SOAP Data which is defined as a simpletype was not mapped correctly to Ruby obj when using wsdl2ruby.rb generated classdef file. (#123)
- * rpc/literal support. (#118)
- * re-implemented local element qualify/unqualify control. handles elementFormDefault and form in WSDL. (#119)
- * Array of an element which has simpleType causes a crash. (#128)
- * parameterOrder may not contain return part so it can be shorter than parts size. Thanks to Hugh. (#139)
- * Samples
- * added !BasicAuth client sample. (#117)
- * added Base64 client/server sample.
- * added Flickr SOAP interface client sample. (#122)
- * added !SalesForce client sample. (#135)
- * updated Thawte CA certificate for !GoogleAdWords sample.
- * updated a client script with the newer version made by Johan. thanks!
- * shortened long file names. (#120)
- * fixed typo in authheader sample. (#129)
- * updated deprecated method usage. (#138)

Revision 9169 - 09/15/2005 02:33 PM - nahi (Hiroshi Nakamura)

- lib/{soap,wsdl,xsd}, test/{soap,wsdl,xsd}: imported soap4r/1.5.5.

#nnn is a ticket number at <http://dev.ctor.org/soap4r>

* SOAP

- * allow to configure an envelope namespace of SOAP request. (#124)


```
TemporaryNamespace = 'http://www.w3.org/2003/05/soap-envelope'
@client.options["soap.envelope.requestnamespace"] =
  TemporaryNamespace
@client.options["soap.envelope.responsenspace"] =
  TemporaryNamespace
@client.do_proc(...)
```
- * let SOAP request XML indent space configurable. see "soap.envelope.no_indent" option. (#130)
- * let external CES configurable.


```
ex. client["soap.mapping.external_ces"] = 'SJIS'. $KCODE is used
```

by default. (#133)
external CES ::= CES used in Ruby object of client and server
internal CES ::= CES used in SOAP/OM

* add iso-8859-1 external CES support. (#106)

* fixed illegal 'qualified' handling of elements. it caused ASP.NET inteoperability problem. (#144)

* added 'soap.envelope.use_numeric_character_reference' (boolean) option to let query XML use numeric character reference in XML, not plain UTF-8 character. !GoogleSearch server seems to not allow plain UTF-8 character since 2005-08-15 update. (#147)

* SOAP::Header::SimpleHeader (de)serialization throws an exception on !SimpleHeader.on_(in|out)bound when header is a String. so we could not use a simple single element headerItem. fixed. thanks to emil. (#129)

* out parameter of rpc operation did not work. (#132)

* follow HTTP redirect only if using http-access2. (#125) (#145)

* add a workaround for importing an WSDL whose path begins with drive letter. (#115)

* WSDL

* SOAP Data which is defined as a simpletype was not mapped correctly to Ruby obj when using wsdl2ruby.rb generated clasdef file. (#123)

* rpc/literal support. (#118)

* re-implemented local element qualify/unqualify control. handles elementFormDefault and form in WSDL. (#119)

* Array of an element which has simpleType causes a crash. (#128)

* prarmeterOrder may not contain return part so it can be shorter than parts size. Thanks to Hugh. (#139)

* Samples

* added !BasicAuth client sample. (#117)

* added Base64 client/server sample.

* added Flickr SOAP interface client sample. (#122)

* added !SalesForce client sample. (#135)

* updated Thawte CA certificate for !GoogleAdWords sample.

* updated a client script with the newer version made by Johan. thanks!

* shortened long file names. (#120)

* fixed typo in authheader sample. (#129)

* updated deprecated method usage. (#138)

Revision 9169 - 09/15/2005 02:33 PM - nahi (Hiroshi Nakamura)

- lib/{soap,wsdl,xsd}, test/{soap,wsdl,xsd}: imported soap4r/1.5.5.

#nnn is a ticket number at <http://dev.ctor.org/soap4r>

* SOAP

- * allow to configure an envelope namespace of SOAP request. (#124)
TemporaryNamespace = 'http://www.w3.org/2003/05/soap-envelope'
@client.options["soap.envelope.requestnamespace"] =
TemporaryNamespace
@client.options["soap.envelope.responsenamespace"] =
TemporaryNamespace
@client.do_proc(...)
- * let SOAP request XML indent space configurable. see
"soap.envelope.no_indent" option. (#130)
- * let external CES configurable.
ex. client["soap.mapping.external_ces"] = 'SJIS'. \$KCODE is used
by default. (#133)
external CES ::= CES used in Ruby object of client and server
internal CES ::= CES used in SOAP/OM
- * add iso-8859-1 external CES support. (#106)
- * fixed illegal 'qualified' handling of elements. it caused
ASP.NET inteoperability problem. (#144)
- * added 'soap.envelope.use_numeric_character_reference' (boolean)
option to let query XML use numeric character reference in XML,
not plain UTF-8 character. !GoogleSearch server seems to not
allow plain UTF-8 character since 2005-08-15 update. (#147)
- * SOAP::Header::SimpleHeader (de)serialization throws an exception
on !SimpleHeader.on_(in/out)bound when header is a String. so we
could not use a simple single element headerItem. fixed. thanks
to emil. (#129)
- * out parameter of rpc operation did not work. (#132)
- * follow HTTP redirect only if using http-access2. (#125) (#145)
- * add a workaround for importing an WSDL whose path begins with
drive letter. (#115)
- * WSDL
- * SOAP Data which is defined as a simpletype was not mapped
correctly to Ruby obj when using wsdl2ruby.rb generated classdef
file. (#123)
- * rpc/literal support. (#118)
- * re-implemented local element qualify/unqualify control. handles
elementFormDefault and form in WSDL. (#119)
- * Array of an element which has simpleType causes a crash. (#128)
- * prarmeterOrder may not contain return part so it can be shorter
than parts size. Thanks to Hugh. (#139)
- * Samples
- * added !BasicAuth client sample. (#117)
- * added Base64 client/server sample.
- * added Flickr SOAP interface client sample. (#122)
- * added !SalesForce client sample. (#135)
- * updated Thawte CA certificate for !GoogleAdWords sample.
- * updated a client script with the newer version made by Johan.
thanks!
- * shortened long file names. (#120)
- * fixed typo in authheader sample. (#129)

* updated deprecated method usage. (#138)

Revision 9169 - 09/15/2005 02:33 PM - nahi (Hiroshi Nakamura)

- lib/{soap,wsdl,xsd}, test/{soap,wsdl,xsd}: imported soap4r/1.5.5.

#nnn is a ticket number at <http://dev.ctor.org/soap4r>

* SOAP

* allow to configure an envelope namespace of SOAP request. (#124)
TemporaryNamespace = 'http://www.w3.org/2003/05/soap-envelope'
@client.options["soap.envelope.requestnamespace"] =
TemporaryNamespace
@client.options["soap.envelope.responsenamespace"] =
TemporaryNamespace
@client.do_proc(...)

* let SOAP request XML indent space configurable. see
"soap.envelope.no_indent" option. (#130)

* let external CES configurable.
ex. client["soap.mapping.external_ces"] = 'SJIS'. \$KCODE is used
by default. (#133)
external CES ::= CES used in Ruby object of client and server
internal CES ::= CES used in SOAP/OM

* add iso-8859-1 external CES support. (#106)

* fixed illegal 'qualified' handling of elements. it caused
ASP.NET inteoperability problem. (#144)

* added 'soap.envelope.use_numeric_character_reference' (boolean)
option to let query XML use numeric character reference in XML,
not plain UTF-8 character. !GoogleSearch server seems to not
allow plain UTF-8 character since 2005-08-15 update. (#147)

* SOAP::Header::SimpleHeader (de)serialization throws an exception
on !SimpleHeader.on_(in|out)bound when header is a String. so we
could not use a simple single element headerItem. fixed. thanks
to emil. (#129)

* out parameter of rpc operation did not work. (#132)

* follow HTTP redirect only if using http-access2. (#125) (#145)

* add a workaround for importing an WSDL whose path begins with
drive letter. (#115)

* WSDL

* SOAP Data which is defined as a simpletype was not mapped
correctly to Ruby obj when using wsdl2ruby.rb generated classdef
file. (#123)

* rpc/literal support. (#118)

* re-implemented local element qualify/unqualify control. handles
elementFormDefault and form in WSDL. (#119)

* Array of an element which has simpleType causes a crash. (#128)

* parameterOrder may not contain return part so it can be shorter
than parts size. Thanks to Hugh. (#139)

* Samples

* added !BasicAuth client sample. (#117)

* added Base64 client/server sample.

- * added Flickr SOAP interface client sample. (#122)
- * added !SalesForce client sample. (#135)
- * updated Thawte CA certificate for !GoogleAdWords sample.
- * updated a client script with the newer version made by Johan. thanks!
- * shortened long file names. (#120)
- * fixed typo in authheader sample. (#129)
- * updated deprecated method usage. (#138)

Revision 9169 - 09/15/2005 02:33 PM - nahi (Hiroshi Nakamura)

- lib/{soap,wsdl,xsd}, test/{soap,wsdl,xsd}: imported soap4r/1.5.5.

#nnn is a ticket number at <http://dev.ctor.org/soap4r>

* SOAP

- * allow to configure an envelope namespace of SOAP request. (#124)


```
TemporaryNamespace = 'http://www.w3.org/2003/05/soap-envelope'
@client.options["soap.envelope.requestnamespace"] =
  TemporaryNamespace
@client.options["soap.envelope.responsensamespace"] =
  TemporaryNamespace
@client.do_proc(...)
```
- * let SOAP request XML indent space configurable. see


```
"soap.envelope.no_indent" option. (#130)
```
- * let external CES configurable.


```
ex. client["soap.mapping.external_ces"] = 'SJIS'. $KCODE is used
by default. (#133)
external CES ::= CES used in Ruby object of client and server
internal CES ::= CES used in SOAP/OM
```
- * add iso-8859-1 external CES support. (#106)
- * fixed illegal 'qualified' handling of elements. it caused
 ASP.NET inteoperability problem. (#144)
- * added 'soap.envelope.use_numeric_character_reference' (boolean)
 option to let query XML use numeric character reference in XML,
 not plain UTF-8 character. !GoogleSearch server seems to not
 allow plain UTF-8 character since 2005-08-15 update. (#147)
- * SOAP::Header::SimpleHeader (de)serialization throws an exception
 on !SimpleHeader.on_(in|out)bound when header is a String. so we
 could not use a simple single element headerItem. fixed. thanks
 to emil. (#129)
- * out parameter of rpc operation did not work. (#132)
- * follow HTTP redirect only if using http-access2. (#125) (#145)
- * add a workaround for importing an WSDL whose path begins with
 drive letter. (#115)
- * WSDL
 - * SOAP Data which is defined as a simpletype was not mapped
 correctly to Ruby obj when using wsdl2ruby.rb generated classdef
 file. (#123)
 - * rpc/literal support. (#118)

- * re-implemented local element qualify/unqualify control. handles elementFormDefault and form in WSDL. (#119)
- * Array of an element which has simpleType causes a crash. (#128)
- * parameterOrder may not contain return part so it can be shorter than parts size. Thanks to Hugh. (#139)
- * Samples
- * added !BasicAuth client sample. (#117)
- * added Base64 client/server sample.
- * added Flickr SOAP interface client sample. (#122)
- * added !SalesForce client sample. (#135)
- * updated Thawte CA certificate for !GoogleAdWords sample.
- * updated a client script with the newer version made by Johan. thanks!
- * shortened long file names. (#120)
- * fixed typo in authheader sample. (#129)
- * updated deprecated method usage. (#138)

Revision 29c5ab0b - 09/15/2005 02:33 PM - nahi (Hiroshi Nakamura)

- lib/{soap,wsdl,xsd}, test/{soap,wsdl,xsd}: imported soap4r/1.5.5.

#nnn is a ticket number at <http://dev.ctor.org/soap4r>

* SOAP

- * allow to configure an envelope namespace of SOAP request. (#124)


```
TemporaryNamespace = 'http://www.w3.org/2003/05/soap-envelope'
@client.options["soap.envelope.requestnamespace"] =
  TemporaryNamespace
@client.options["soap.envelope.responsenamespace"] =
  TemporaryNamespace
@client.do_proc(...)
```
- * let SOAP request XML indent space configurable. see "soap.envelope.no_indent" option. (#130)
- * let external CES configurable.


```
ex. client["soap.mapping.external_ces"] = 'SJIS'. $KCODE is used
by default. (#133)
  external CES ::= CES used in Ruby object of client and server
  internal CES ::= CES used in SOAP/OM
```
- * add iso-8859-1 external CES support. (#106)
- * fixed illegal 'qualified' handling of elements. it caused ASP.NET inteoperability problem. (#144)
- * added 'soap.envelope.use_numeric_character_reference' (boolean) option to let query XML use numeric character reference in XML, not plain UTF-8 character. !GoogleSearch server seems to not allow plain UTF-8 character since 2005-08-15 update. (#147)
- * SOAP::Header::SimpleHeader (de)serialization throws an exception on !SimpleHeader.on_(in|out)bound when header is a String. so we could not use a simple single element headerItem. fixed. thanks to emil. (#129)

- * out parameter of rpc operation did not work. (#132)
- * follow HTTP redirect only if using http-access2. (#125) (#145)
- * add a workaround for importing an WSDL whose path begins with drive letter. (#115)
- * WSDL
- * SOAP Data which is defined as a simpletype was not mapped correctly to Ruby obj when using wsdl2ruby.rb generated classdef file. (#123)
- * rpc/literal support. (#118)
- * re-implemented local element qualify/unqualify control. handles elementFormDefault and form in WSDL. (#119)
- * Array of an element which has simpleType causes a crash. (#128)
- * parameterOrder may not contain return part so it can be shorter than parts size. Thanks to Hugh. (#139)
- * Samples
- * added !BasicAuth client sample. (#117)
- * added Base64 client/server sample.
- * added Flickr SOAP interface client sample. (#122)
- * added !SalesForce client sample. (#135)
- * updated Thawte CA certificate for !GoogleAdWords sample.
- * updated a client script with the newer version made by Johan. thanks!
- * shortened long file names. (#120)
- * fixed typo in authheader sample. (#129)
- * updated deprecated method usage. (#138)

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@9169 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 9171 - 09/15/2005 02:47 PM - nahi (Hiroshi Nakamura)

- lib/{soap,wsdl,xsd}, test/{soap,wsdl,xsd}: imported soap4r/1.5.5.

#nnn is a ticket number at <http://dev.ctor.org/soap4r>

* SOAP

```
* allow to configure an envelope namespace of SOAP request. (#124)
  TemporaryNamespace = 'http://www.w3.org/2003/05/soap-envelope'
  @client.options["soap.envelope.requestnamespace"] =
    TemporaryNamespace
  @client.options["soap.envelope.responsenamespace"] =
    TemporaryNamespace
  @client.do_proc(...)
```

```
* let SOAP request XML indent space configurable. see
  "soap.envelope.no_indent" option. (#130)
```

```
* let external CES configurable.
  ex. client["soap.mapping.external_ces"] = 'SJIS'. $KCODE is used
  by default. (#133)
  external CES ::= CES used in Ruby object of client and server
  internal CES ::= CES used in SOAP/OM
```

- * add iso-8859-1 external CES support. (#106)
- * fixed illegal 'qualified' handling of elements. it caused ASP.NET inteoperability problem. (#144)
- * added 'soap.envelope.use_numeric_character_reference' (boolean) option to let query XML use numeric character reference in XML, not plain UTF-8 character. !GoogleSearch server seems to not allow plain UTF-8 character since 2005-08-15 update. (#147)
- * SOAP::Header::SimpleHeader (de)serialization throws an exception on !SimpleHeader.on_(in|out)bound when header is a String. so we could not use a simple single element headerItem. fixed. thanks to emil. (#129)
- * out parameter of rpc operation did not work. (#132)
- * follow HTTP redirect only if using http-access2. (#125) (#145)
- * add a workaround for importing an WSDL whose path begins with drive letter. (#115)
- * WSDL
- * SOAP Data which is defined as a simpletype was not mapped correctly to Ruby obj when using wsdl2ruby.rb generated classdef file. (#123)
- * rpc/literal support. (#118)
- * re-implemented local element qualify/unqualify control. handles elementFormDefault and form in WSDL. (#119)
- * Array of an element which has simpleType causes a crash. (#128)
- * prarmeterOrder may not contain return part so it can be shorter than parts size. Thanks to Hugh. (#139)
- * Samples
- * added !BasicAuth client sample. (#117)
- * added Base64 client/server sample.
- * added Flickr SOAP interface client sample. (#122)
- * added !SalesForce client sample. (#135)
- * updated Thawte CA certificate for !GoogleAdWords sample.
- * updated a client script with the newer version made by Johan. thanks!
- * shortened long file names. (#120)
- * fixed typo in authheader sample. (#129)
- * updated deprecated method usage. (#138)

Revision 533c2426 - 09/15/2005 02:47 PM - nahi (Hiroshi Nakamura)

- lib/{soap,wsdl,xsd}, test/{soap,wsdl,xsd}: imported soap4r/1.5.5.

#nnn is a ticket number at <http://dev.ctor.org/soap4r>

* SOAP

- * allow to configure an envelope namespace of SOAP request. (#124)
TemporaryNamespace = 'http://www.w3.org/2003/05/soap-envelope'
@client.options["soap.envelope.requestnamespace"] =

```
    TemporaryNamespace
    @client.options["soap.envelope.responsenamespace"] =
    TemporaryNamespace
    @client.do_proc(...)
```

- * let SOAP request XML indent space configurable. see "soap.envelope.no_indent" option. (#130)
- * let external CES configurable.
ex. client["soap.mapping.external_ces"] = 'SJIS'. \$KCODE is used by default. (#133)
external CES ::= CES used in Ruby object of client and server
internal CES ::= CES used in SOAP/OM
- * add iso-8859-1 external CES support. (#106)
- * fixed illegal 'qualified' handling of elements. it caused ASP.NET inteoperability problem. (#144)
- * added 'soap.envelope.use_numeric_character_reference' (boolean) option to let query XML use numeric character reference in XML, not plain UTF-8 character. !GoogleSearch server seems to not allow plain UTF-8 character since 2005-08-15 update. (#147)
- * SOAP::Header::SimpleHeader (de)serialization throws an exception on !SimpleHeader.on_(in|out)bound when header is a String. so we could not use a simple single element headerItem. fixed. thanks to emil. (#129)
- * out parameter of rpc operation did not work. (#132)
- * follow HTTP redirect only if using http-access2. (#125) (#145)
- * add a workaround for importing an WSDL whose path begins with drive letter. (#115)
- * WSDL
- * SOAP Data which is defined as a simpletype was not mapped correctly to Ruby obj when using wsdl2ruby.rb generated classdef file. (#123)
- * rpc/literal support. (#118)
- * re-implemented local element qualify/unqualify control. handles elementFormDefault and form in WSDL. (#119)
- * Array of an element which has simpleType causes a crash. (#128)
- * prarmeterOrder may not contain return part so it can be shorter than parts size. Thanks to Hugh. (#139)
- * Samples
- * added !BasicAuth client sample. (#117)
- * added Base64 client/server sample.
- * added Flickr SOAP interface client sample. (#122)
- * added !SalesForce client sample. (#135)
- * updated Thawte CA certificate for !GoogleAdWords sample.
- * updated a client script with the newer version made by Johan. thanks!
- * shortened long file names. (#120)
- * fixed typo in authheader sample. (#129)
- * updated deprecated method usage. (#138)

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_8@9171 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 8dded0725b1ed215c8da1131a132b4ca6eb2f434 - 05/01/2009 07:46 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@23320 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 8dded072 - 05/01/2009 07:46 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@23320 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 23320 - 05/01/2009 07:46 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]

Revision 23320 - 05/01/2009 07:46 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]

Revision 23320 - 05/01/2009 07:46 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]

Revision 23320 - 05/01/2009 07:46 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]

Revision 23320 - 05/01/2009 07:46 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]

Revision 23320 - 05/01/2009 07:46 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]

Revision f7b562ef0114d8421d7d1cc6595e0c056feada8f - 05/01/2009 07:52 AM - knu (Akinori MUSHHA)

- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@23321 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision f7b562ef - 05/01/2009 07:52 AM - knu (Akinori MUSHHA)

- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@23321 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 23321 - 05/01/2009 07:52 AM - knu (Akinori MUSHHA)

- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

Revision 23321 - 05/01/2009 07:52 AM - knu (Akinori MUSHHA)

- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

Revision 23321 - 05/01/2009 07:52 AM - knu (Akinori MUSHHA)

- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

class. [Bug #118]

Revision 23321 - 05/01/2009 07:52 AM - knu (Akinori MUSHHA)

- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

Revision 23321 - 05/01/2009 07:52 AM - knu (Akinori MUSHHA)

- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

Revision 23321 - 05/01/2009 07:52 AM - knu (Akinori MUSHHA)

- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

Revision e872e1dbacf4571b72c25d140a37e683873efa02 - 05/01/2009 07:54 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]
- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_8@23322 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision e872e1db - 05/01/2009 07:54 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]
- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_1_8@23322 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 23322 - 05/01/2009 07:54 AM - knu (Akinori MUSHHA)

- lib/set.rb (SortedSet#add): Do not let an uncomparable object in. [Bug #118]
- lib/set.rb (Set#merge): Only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class. [Bug #118]

Revision e3e8af93 - 02/16/2018 02:45 PM - k0kubun (Takashi Kokubun)

mjit.c: fix deadlock on class serial increment

This is reported by @hasimo. Fixing a case like this:

```
#0 Ill_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
#1 0x00007fc7bd824dbd in __GI_pthread_mutex_lock (mutex=mutex@entry=0x55946d294440 ) at ../nptl/pthread_mutex_lock.c:80
#2 0x000055946cec54d9 in rb_native_mutex_lock (lock=lock@entry=0x55946d294440 ) at thread_pthread.c:211
#3 0x000055946cde10ca in CRITICAL_SECTION_START (msg=0x55946cfb5423 "mjit_gc_start_hook", level=4) at mjit.c:392
#4 mjit_gc_start_hook () at mjit.c:412
```

#5 0x000055946cda0dfe in gc_enter (event=0x55946cfaf91e "gc_rest", objspace=0x55946da51760) at gc.c:6623
#6 gc_rest (objspace=objspace@entry=0x55946da51760) at gc.c:6515
#7 0x000055946cd9f1cf in gc_rest (objspace=0x55946da51760) at gc.c:7841
#8 objspace_malloc_increase (objspace=objspace@entry=0x55946da51760, new_size=, old_size=old_size@entry=0, type=type@entry=MEMOP_TYPE_MALLOC, mem=0x7fc7a4439010) at gc.c:7842
#9 0x000055946cda1706 in objspace_malloc_fixup (size=, mem=0x7fc7a4439010, objspace=0x55946da51760) at gc.c:7910
#10 objspace_xmalloc0 (objspace=0x55946da51760, size=, size@entry=3145728) at gc.c:7939
#11 0x000055946cda3620 in ruby_xmalloc0 (size=3145728) at gc.c:8006
#12 ruby_xmalloc (size=size@entry=3145728) at gc.c:8015
#13 0x000055946ce93f4c in st_init_table_with_size (type=0x55946d28da30, size=) at st.c:602
#14 0x000055946ce94287 in rebuild_table (tab=tab@entry=0x55946db669f0) at st.c:777
#15 0x000055946ce963f7 in rebuild_table_if_necessary (tab=0x55946db669f0) at st.c:1139
#16 st_add_direct_with_hash (hash=8577035585096733536, value=20, key=808451, tab=0x55946db669f0) at st.c:1207
#17 st_update (tab=0x55946db669f0, key=key@entry=808451, func=, arg=140726472841392) at st.c:1512
#18 0x000055946cda9e27 in tbl_update (optional_arg=, func=, key=, hash=) at hash.c:561
#19 rb_hash_aset (hash=94095983218480, key=key@entry=808451, val=val@entry=20) at hash.c:1654
#20 0x000055946cde243a in mjit_add_class_serial (class_serial=class_serial@entry=404225) at mjit.c:1414 3
#21 0x000055946cefcfab in rb_next_class_serial () at vm.c:321
#22 0x000055946cf48324 in class_alloc (klass=, flags=28) at class.c:178
#23 rb_include_class_new (module=module@entry=94096115733840, super=0) at class.c:820
#24 0x000055946cf487ac in include_modules_at (klass=klass@entry=94096135960920, c=, module=, module@entry=94096115734160, search_super=search_super@entry=1) at class.c:913
#25 0x000055946cf48ac8 in rb_include_module (klass=94096135960920, module=module@entry=94096115734160) at class.c:870
#26 0x000055946cd84993 in rb_mod_append_features (module=94096115734160, include=) at eval.c:1178
#27 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6c9a20, cc=0x7ffd6f6c9ba0, argv=0x7ffd6f6c9ba0, calling=0x7ffd6f6c9a30, ec=0x55946da519c8) at vm_eval.c:87
#28 vm_call0_cfunc (argv=0x7ffd6f6c9ba0, cc=0x7ffd6f6c9ba0, ci=0x7ffd6f6c9a20, calling=0x7ffd6f6c9a30, ec=0x55946da519c8) at vm_eval.c:102
#29 vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6c9ae0, ci=ci@entry=0x7ffd6f6c9ad0, cc=cc@entry=0x7ffd6f6c9b00, argv=argv@entry=0x7ffd6f6c9ba0) at vm_eval.c:133
#30 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6c9ba0, argc=1, id=4849, recv=94096115734160, ec=0x55946da519c8) at vm_eval.c:60
#31 rb_call0 (ec=0x55946da519c8, recv=94096115734160, mid=4849, mid@entry=94096135960920, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6c9ba0, scope=scope@entry=CALL_FCALL, self=94096135960920) at vm_eval.c:302
#32 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x7ffd6f6c9ba0, argc=1, mid=94096135960920, recv=) at vm_eval.c:595
#33 rb_funcallv (recv=, mid=mid@entry=4849, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6c9ba0) at vm_eval.c:825
#34 0x000055946cd848a7 in rb_mod_include (argc=0, argv=0x7fc7bdb4fce8, module=94096135960920) at eval.c:1203
#35 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594710c06b0, cc=, calling=, reg_cfp=0x7fc7bdc4df00, ec=0x55946da519c8) at vm_inshelper.c:1928
#36 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4df00, calling=, ci=0x5594710c06b0, cc=) at vm_inshelper.c:1944
#37 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#38 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#39 0x000055946cf035fc in invoke_block (captured=0x7ffd6f6ca0a0, opt_pc=, type=, cref=0x559476c23930, self=94096135960920, iseq=0x559471335ee0, ec=0x55946da519c8) at vm.c:1005
#40 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7ffd6f6ca0a0, self=94096135960920, argc=, argv=, passed_block_handler=0, cref=0x559476c23930, is_lambda=0) at vm.c:1057
#41 0x000055946cf04520 in invoke_block_from_c_bh (ec=ec@entry=0x55946da519c8, block_handler=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108, cref=, is_lambda=, is_lambda@entry=0, force_blockarg=0, passed_block_handler=0) at vm.c:1075
#42 0x000055946cf04958 in vm_yield_with_cref (is_lambda=0, cref=, argv=0x7ffd6f6ca108, argc=1, ec=0x55946da519c8) at vm.c:1112
#43 yield_under (under=94096135960920, self=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108) at vm_eval.c:1572
#44 0x000055946cf04b12 in rb_mod_module_exec (argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108, mod=) at vm_eval.c:1770
#45 0x000055946ce00fc6 in rb_mod_initialize (module=94096135960920) at object.c:1978
#46 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6ca130, cc=0x7fc7bdb4fc98, argv=0x7fc7bdb4fc98, calling=0x7ffd6f6ca140, ec=0x55946da519c8) at vm_eval.c:87
#47 vm_call0_cfunc (argv=0x7fc7bdb4fc98, cc=0x7fc7bdb4fc98, ci=0x7ffd6f6ca130, calling=0x7ffd6f6ca140, ec=0x55946da519c8) at vm_eval.c:102
#48 vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6ca1f0, ci=ci@entry=0x7ffd6f6ca1e0, cc=cc@entry=0x7ffd6f6ca210, argv=argv@entry=0x7fc7bdb4fc98) at vm_eval.c:133
#49 0x000055946cf074b2 in vm_call0 (me=, argv=0x7fc7bdb4fc98, argc=0, id=3057, recv=94096135960920, ec=0x55946da519c8) at vm_eval.c:60
#50 rb_call0 (ec=0x55946da519c8, recv=recv@entry=94096135960920, mid=mid@entry=3057, argc=argc@entry=3057, argv=argv@entry=0x0, scope=scope@entry=CALL_FCALL, self=94095983466120) at vm_eval.c:302
#51 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x0, argc=3057, mid=3057, recv=94096135960920) at vm_eval.c:595
#52 rb_funcallv (recv=recv@entry=94096135960920, mid=mid@entry=3057, argc=argc@entry=0, argv=argv@entry=0x7fc7bdb4fc98) at vm_eval.c:825
#53 0x000055946cd89673 in rb_obj_call_init (obj=obj@entry=94096135960920, argc=argc@entry=0, argv=argv@entry=0x7fc7bdb4fc98) at eval.c:1590
#54 0x000055946ce048a1 in rb_class_s_new (argc=0, argv=0x7fc7bdb4fc98, klass=) at object.c:2153
#55 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x559471339c80, cc=, calling=, reg_cfp=0x7fc7bdc4dfa8, ec=0x55946da519c8) at vm_inshelper.c:1928
#56 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4dfa8, calling=, ci=0x559471339c80, cc=) at vm_inshelper.c:1944
#57 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#58 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#59 0x000055946cf03274 in invoke_bmethod (ec=ec@entry=0x55946da519c8, iseq=iseq@entry=0x55946f15cb0c0, self=self@entry=94096115734640, me=me@entry=0x559475664e28, type=type@entry=572653825, opt_pc=0, captured=0x55947158e8a0) at vm.c:1026
#60 0x000055946cf03534 in invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x55947158e8a0, self=94096115734640, argc=, argv=,

passed_block_handler=0, cref=0x0, is_lambda=1) at vm.c:1060
#61 0x000055946cf036ce in invoke_block_from_c_proc (ec=, proc=, self=, argc=1, argv=, passed_block_handler=, is_lambda=) at vm.c:1150
#62 0x000055946cf03811 in vm_invoke_bmethod (block_handler=, argv=, argc=1, self=, proc=, ec=0x55946da519c8) at vm.c:1175
#63 vm_call_bmethod_body (ci=, cc=0x55946f625960, argv=, calling=0x7ffd6f6ca9b0, ec=0x55946da519c8) at vm_inshelper.c:1971
#64 vm_call_bmethod (ec=0x55946da519c8, cfp=0x7fc7bdc4e0c0, calling=0x7ffd6f6ca9b0, ci=, cc=0x55946f625960) at vm_inshelper.c:1988
#65 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e0c0, calling=, ci=, cc=) at vm_inshelper.c:2417
#66 0x000055946cf0e63e in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:797
#67 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#68 0x000055946cf06767 in vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6cad00, ci=ci@entry=0x7ffd6f6cacf0, cc=cc@entry=0x7ffd6f6cad20, argv=argv@entry=0x7ffd6f6cadb0) at vm_eval.c:129
#69 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6cadb0, argc=1, id=3681, recv=94096115734640, ec=0x55946da519c8) at vm_eval.c:60
#70 rb_call0 (ec=0x55946da519c8, recv=recv@entry=94096115734640, mid=3681, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cad90, scope=scope@entry=CALL_FCALL, self=94095983452880) at vm_eval.c:302
#71 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x7ffd6f6cad90, argc=1, mid=, recv=94096115734640) at vm_eval.c:595
#72 rb_funcallv (recv=recv@entry=94096115734640, mid=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cadb0) at vm_eval.c:825
#73 0x000055946cf477d2 in rb_class_inherited (super=super@entry=94096115734640, klass=klass@entry=94096135961440) at class.c:625
#74 0x000055946cf0f796 in vm_declare_class (super=, cbase=94096094868200, flags=, id=847387) at vm_inshelper.c:3134
#75 vm_define_class (super=, cbase=, flags=, id=847387) at vm_inshelper.c:3167
#76 vm_find_or_create_class_by_id (super=, cbase=, flags=, id=847387) at vm_inshelper.c:3196
#77 vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:685
#78 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#79 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x559471f1b548) at vm.c:2046
#80 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94096135872160, wrap=wrap@entry=0) at load.c:611
#81 0x000055946cdd41f1 in rb_require_internal (fname=94096135872400, fname@entry=94096135872440, safe=0) at load.c:992
#82 0x000055946cdd4493 in rb_require_safe (safe=, fname=94096135872440) at load.c:1038
#83 rb_f_require (obj=, fname=94096135872440) at load.c:820
#84 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594708700a0, cc=, calling=, reg_cfp=0x7fc7bdc4e168, ec=0x55946da519c8) at vm_inshelper.c:1928
#85 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e168, calling=, ci=0x5594708700a0, cc=) at vm_inshelper.c:1944
#86 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e168, calling=, ci=, cc=) at vm_inshelper.c:2417
#87 0x000055946cf0e63e in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:797
#88 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#89 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4e490, opt_pc=, type=, cref=0x0, self=94096096426480, iseq=0x55946e49f0b8, ec=0x55946da519c8) at vm.c:1005
#90 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4e490, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#91 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#92 vm_yield (argc=1, argv=0x7ffd6f6cb938, ec=) at vm.c:1120
#93 rb_yield_0 (argv=0x7ffd6f6cb938, argc=1) at vm_eval.c:980
#94 rb_yield_1 (val=94096130169040) at vm_eval.c:986
#95 rb_yield (val=) at vm_eval.c:996
#96 0x000055946cf2113d in rb_ary_each (ary=94096076222560) at array.c:1820
#97 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946ed0d090, cc=, calling=, reg_cfp=0x7fc7bdc4e478, ec=0x55946da519c8) at vm_inshelper.c:1928
#98 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e478, calling=, ci=0x55946ed0d090, cc=) at vm_inshelper.c:1944
#99 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#100 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#101 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4e500, opt_pc=, type=, cref=0x0, self=94096096426480, iseq=0x55946e49f298, ec=0x55946da519c8) at vm.c:1005
#102 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4e500, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#103 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#104 vm_yield (argc=1, argv=0x7ffd6f6cbdf8, ec=) at vm.c:1120
#105 rb_yield_0 (argv=0x7ffd6f6cbdf8, argc=1) at vm_eval.c:980
#106 rb_yield_1 (val=94096095502480) at vm_eval.c:986
#107 rb_yield (val=) at vm_eval.c:996
#108 0x000055946cf2113d in rb_ary_each (ary=94096095328480) at array.c:1820
#109 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e8552a0, cc=, calling=, reg_cfp=0x7fc7bdc4e4e8, ec=0x55946da519c8) at vm_inshelper.c:1928
#110 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e4e8, calling=, ci=0x55946e8552a0, cc=) at vm_inshelper.c:1944
#111 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#112 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#113 0x000055946cf06767 in vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6cc2d0, ci=ci@entry=0x7ffd6f6cc2c0, cc=cc@entry=0x7ffd6f6cc2f0, argv=argv@entry=0x7ffd6f6cc390) at vm_eval.c:129
#114 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6cc390, argc=0, id=135807, recv=94096096186440, ec=0x55946da519c8) at vm_eval.c:60
#115 rb_call0 (ec=0x55946da519c8, recv=94096096186440, mid=135807, argc=, argv=argv@entry=0x8, scope=scope@entry=CALL_PUBLIC, self=94095993048320) at vm_eval.c:302

#116 0x000055946cf0a31a in rb_call (scope=CALL_PUBLIC, argv=0x8, argc=, mid=, recv=) at vm_eval.c:595
#117 rb_funcall_with_block (recv=, mid=, argc=argc@entry=0, argv=argv@entry=0x7ffd6f6cc390, passed_procval=passed_procval@entry=8) at vm_eval.c:857
#118 0x000055946ceb319c in rb_sym_proc_call (mid=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cc388, passed_proc=passed_proc@entry=8) at string.c:10480
#119 0x000055946cf0477c in vm_yield_with_symbol (block_handler=0, argv=0x7ffd6f6cc388, argc=1, symbol=, ec=) at vm_inshelper.c:2573
#120 invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1084
#121 vm_yield (argc=1, argv=0x7ffd6f6cc388, ec=) at vm.c:1120
#122 rb_yield_0 (argv=0x7ffd6f6cc388, argc=1) at vm_eval.c:980
#123 rb_yield_1 (val=94096096186440) at vm_eval.c:986
#124 rb_yield (val=) at vm_eval.c:996
#125 0x000055946cf2113d in rb_ary_each (ary=94095993048320) at array.c:1820
#126 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594744d8280, cc=, calling=, reg_cfp=0x7fc7bdc4e590, ec=0x55946da519c8) at vm_inshelper.c:1928
#127 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e590, calling=, ci=0x5594744d8280, cc=) at vm_inshelper.c:1944
#128 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e590, calling=, ci=, cc=) at vm_inshelper.c:2417
#129 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#130 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#131 0x000055946cf035fc in invoke_block (captured=0x7ffd6f6cc8e0, opt_pc=, type=, cref=0x559474604128, self=94096096426480, iseq=0x5594745027e8, ec=0x55946da519c8) at vm.c:1005
#132 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7ffd6f6cc8e0, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x559474604128, is_lambda=0) at vm.c:1057
#133 0x000055946cf04520 in invoke_block_from_c_bh (ec=ec@entry=0x55946da519c8, block_handler=, argc=argc@entry=1, argv=argv@entry=0x7fc7bdb4f7f8, cref=, is_lambda=, is_lambda@entry=0, force_blockarg=0, passed_block_handler=0) at vm.c:1075
#134 0x000055946cf04958 in vm_yield_with_cref (is_lambda=0, cref=, argv=0x7fc7bdb4f7f8, argc=1, ec=0x55946da519c8) at vm.c:1112
#135 yield_under (under=94096093646040, self=, argc=1, argv=0x7fc7bdb4f7f8) at vm_eval.c:1572
#136 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e5cd230, cc=, calling=, reg_cfp=0x7fc7bdc4e600, ec=0x55946da519c8) at vm_inshelper.c:1928
#137 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e600, calling=, ci=0x55946e5cd230, cc=) at vm_inshelper.c:1944
#138 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#139 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#140 0x000055946cf035fc in invoke_block (captured=0x5594744c4fb0, opt_pc=, type=, cref=0x0, self=94095990659320, iseq=0x55946e192aa0, ec=0x55946da519c8) at vm.c:1005
#141 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x5594744c4fb0, self=94095990659320, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#142 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#143 vm_yield (argc=1, argv=0x7ffd6f6ccdc8, ec=) at vm.c:1120
#144 rb_yield_0 (argv=0x7ffd6f6ccdc8, argc=1) at vm_eval.c:980
#145 rb_yield_1 (val=94096094867920) at vm_eval.c:986
#146 rb_yield (val=) at vm_eval.c:996
#147 0x000055946cf2113d in rb_ary_each (ary=94096094867160) at array.c:1820
#148 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6cce00, cc=0x7ffd6f6cce70, argv=0x7fc7bdb4f6b8, calling=0x7ffd6f6cce50, ec=0x55946da519c8) at vm_eval.c:87
#149 vm_call0_cfunc (argv=0x7fc7bdb4f6b8, cc=0x7ffd6f6cce70, ci=0x7ffd6f6cce00, calling=0x7ffd6f6cce50, ec=0x55946da519c8) at vm_eval.c:102
#150 vm_call0_body (ec=0x55946da519c8, calling=calling@entry=0x7ffd6f6cce00, ci=ci@entry=0x7ffd6f6cceb0, cc=cc@entry=0x7ffd6f6cce00, argv=0x7fc7bdb4f6b8) at vm_eval.c:133
#151 0x000055946cf06c50 in vm_call0 (me=, argv=, argc=, id=, recv=, ec=) at vm_eval.c:60
#152 rb_vm_call (ec=, recv=, id=, argc=, argv=, me=) at vm_eval.c:209
#153 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946dba2780, cc=, calling=, reg_cfp=0x7fc7bdc4e7c0, ec=0x55946da519c8) at vm_inshelper.c:1928
#154 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e7c0, calling=, ci=0x55946dba2780, cc=) at vm_inshelper.c:1944
#155 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e7c0, calling=, ci=, cc=) at vm_inshelper.c:2417
#156 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#157 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#158 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946dfed3a8) at vm.c:2046
#159 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095988939080, wrap=wrap@entry=0) at load.c:611
#160 0x000055946cdd41f1 in rb_require_internal (fname=94095988939160, fname@entry=94095988939200, safe=0) at load.c:992
#161 0x000055946cdd4493 in rb_require_safe (safe=, fname=94095988939200) at load.c:1038
#162 rb_f_require (obj=, fname=94095988939200) at load.c:820
#163 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946dffa920, cc=, calling=, reg_cfp=0x7fc7bdc4e948, ec=0x55946da519c8) at vm_inshelper.c:1928
#164 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e948, calling=, ci=0x55946dffa920, cc=) at vm_inshelper.c:1944
#165 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e948, calling=, ci=, cc=) at vm_inshelper.c:2417
#166 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#167 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#168 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946dfef438) at vm.c:2046
#169 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993039280, wrap=wrap@entry=0) at load.c:611
#170 0x000055946cdd41f1 in rb_require_internal (fname=fname@entry=94095993043840, safe=0) at load.c:992

#171 0x000055946cdd4493 in rb_require_safe (safe=, fname=94095993043840) at load.c:1038
#172 rb_f_require (obj=, fname=94095993043840) at load.c:820
#173 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946ea12160, cc=, calling=, reg_cfp=0x7fc7bdc4e9b8, ec=0x55946da519c8) at vm_inshelper.c:1928
#174 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e9b8, calling=, ci=0x55946ea12160, cc=) at vm_inshelper.c:1944
#175 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e9b8, calling=, ci=, cc=) at vm_inshelper.c:2417
#176 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#177 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#178 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e3d7c48) at vm.c:2046
#179 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095988920840, wrap=) at load.c:611
#180 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095988920840) at load.c:642
#181 rb_f_load (argc=, argv=) at load.c:710
#182 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e7c96e0, cc=, calling=, reg_cfp=0x7fc7bdc4ea28, ec=0x55946da519c8) at vm_inshelper.c:1928
#183 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ea28, calling=, ci=0x55946e7c96e0, cc=) at vm_inshelper.c:1944
#184 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ea28, calling=, ci=, cc=) at vm_inshelper.c:2417
#185 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#186 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#187 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4eae8, opt_pc=, type=, cref=0x0, self=9409598891400, iseq=0x55946e4dae10, ec=0x55946da519c8) at vm.c:1005
#188 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4eae8, self=9409598891400, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#189 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#190 vm_yield (argc=1, argv=0x7ffd6f6ce8e8, ec=) at vm.c:1120
#191 rb_yield_0 (argv=0x7ffd6f6ce8e8, argc=1) at vm_eval.c:980
#192 rb_yield_1 (val=94095988924840) at vm_eval.c:986
#193 rb_yield (val=) at vm_eval.c:996
#194 0x000055946cf2113d in rb_ary_each (ary=94095988920960) at array.c:1820
#195 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e421000, cc=, calling=, reg_cfp=0x7fc7bdc4ead0, ec=0x55946da519c8) at vm_inshelper.c:1928
#196 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ead0, calling=, ci=0x55946e421000, cc=) at vm_inshelper.c:1944
#197 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ead0, calling=, ci=, cc=) at vm_inshelper.c:2417
#198 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#199 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#200 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e35f2c0) at vm.c:2046
#201 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095992643280, wrap=) at load.c:611
#202 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095992643280) at load.c:642
#203 rb_f_load (argc=, argv=) at load.c:710
#204 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e24bb20, cc=, calling=, reg_cfp=0x7fc7bdc4ec58, ec=0x55946da519c8) at vm_inshelper.c:1928
#205 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ec58, calling=, ci=0x55946e24bb20, cc=) at vm_inshelper.c:1944
#206 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ec58, calling=, ci=, cc=) at vm_inshelper.c:2417
#207 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#208 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#209 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e3a7390) at vm.c:2046
#210 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993616120, wrap=) at load.c:611
#211 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095993616120) at load.c:642
#212 rb_f_load (argc=, argv=) at load.c:710
#213 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e6a00d0, cc=, calling=, reg_cfp=0x7fc7bdc4ecc8, ec=0x55946da519c8) at vm_inshelper.c:1928
#214 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ecc8, calling=, ci=0x55946e6a00d0, cc=) at vm_inshelper.c:1944
#215 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ecc8, calling=, ci=, cc=) at vm_inshelper.c:2417
#216 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#217 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#218 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e470100) at vm.c:2046
#219 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993672200, wrap=) at load.c:611
#220 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095993672200) at load.c:642
#221 rb_f_load (argc=, argv=) at load.c:710
#222 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e3306a0, cc=, calling=, reg_cfp=0x7fc7bdc4efa0, ec=0x55946da519c8) at vm_inshelper.c:1928
#223 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4efa0, calling=, ci=0x55946e3306a0, cc=) at vm_inshelper.c:1944
#224 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4efa0, calling=, ci=, cc=) at vm_inshelper.c:2417
#225 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#226 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#227 0x000055946cf119d5 in rb_iseq_eval_main (iseq=iseq@entry=0x55946e4bbba0) at vm.c:2057
#228 0x000055946cd83d54 in ruby_exec_internal (n=0x55946e4bbba0) at eval.c:247
#229 0x000055946cd87fdf in ruby_exec_node (n=0x55946e4bbba0) at eval.c:311
#230 ruby_run_node (n=) at eval.c:303

#231 0x000055946cd831bf in main (argc=22, argv=0x7ffd6f6d0148) at ./main.c:42

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@62433 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 62433 - 02/16/2018 02:45 PM - k0kubun (Takashi Kokubun)

mjit.c: fix deadlock on class serial increment

This is reported by @hasimo. Fixing a case like this:

```
#0 Ill_lock_wait () at ./sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
#1 0x00007fc7bd824dbd in __GI_pthread_mutex_lock (mutex=mutex@entry=0x55946d294440 ) at ./nptl/pthread_mutex_lock.c:80 4
#2 0x000055946cec54d9 in rb_native_mutex_lock (lock=lock@entry=0x55946d294440 ) at thread_pthread.c:211
#3 0x000055946cde10ca in CRITICAL_SECTION_START (msg=0x55946cfb5423 "mjit_gc_start_hook", level=4) at mjit.c:392
#4 mjit_gc_start_hook () at mjit.c:412
#5 0x000055946cda0dfe in gc_enter (event=0x55946cfaf91e "gc_rest", objspace=0x55946da51760) at gc.c:6623
#6 gc_rest (objspace=objspace@entry=0x55946da51760) at gc.c:6515
#7 0x000055946cd9f1cf in gc_rest (objspace=0x55946da51760) at gc.c:7841
#8 objspace_malloc_increase (objspace=objspace@entry=0x55946da51760, new_size=, old_size=old_size@entry=0,
type=type@entry=MEMOP_TYPE_MALLOC, mem=0x7fc7a4439010) at gc.c:7842
#9 0x000055946cda1706 in objspace_malloc_fixup (size=, mem=0x7fc7a4439010, objspace=0x55946da51760) at gc.c:7910
#10 objspace_xmalloc0 (objspace=0x55946da51760, size=, size@entry=3145728) at gc.c:7939
#11 0x000055946cda3620 in ruby_xmalloc0 (size=3145728) at gc.c:8006
#12 ruby_xmalloc (size=size@entry=3145728) at gc.c:8015
#13 0x000055946ce93f4c in st_init_table_with_size (type=0x55946d28da30 , size=) at st.c:602
#14 0x000055946ce94287 in rebuild_table (tab=tab@entry=0x55946db669f0) at st.c:777
#15 0x000055946ce963f7 in rebuild_table_if_necessary (tab=0x55946db669f0) at st.c:1139
#16 st_add_direct_with_hash (hash=8577035585096733536, value=20, key=808451, tab=0x55946db669f0) at st.c:1207
#17 st_update (tab=0x55946db669f0, key=key@entry=808451, func=, arg=140726472841392) at st.c:1512
#18 0x000055946cda9e27 in tbl_update (optional_arg=, func=, key=, hash=) at hash.c:561
#19 rb_hash_aset (hash=94095983218480, key=key@entry=808451, val=val@entry=20) at hash.c:1654
#20 0x000055946cde243a in mjit_add_class_serial (class_serial=class_serial@entry=404225) at mjit.c:1414 3
#21 0x000055946cefcfab in rb_next_class_serial () at vm.c:321
#22 0x000055946cf48324 in class_alloc (klass=, flags=28) at class.c:178
#23 rb_include_class_new (module=module@entry=94096115733840, super=0) at class.c:820
#24 0x000055946cf487ac in include_modules_at (klass=class@entry=94096135960920, c=, module=, module@entry=94096115734160,
search_super=search_super@entry=1) at class.c:913
#25 0x000055946cf48ac8 in rb_include_module (klass=94096135960920, module=module@entry=94096115734160) at class.c:870
#26 0x000055946cd84993 in rb_mod_append_features (module=94096115734160, include=) at eval.c:1178
#27 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6c9a20, cc=0x7ffd6f6c9ba0, argv=0x7ffd6f6c9ba0, calling=0x7ffd6f6c9a30,
ec=0x55946da519c8) at vm_eval.c:87
#28 vm_call0_cfunc (argv=0x7ffd6f6c9ba0, cc=0x7ffd6f6c9ba0, ci=0x7ffd6f6c9a20, calling=0x7ffd6f6c9a30, ec=0x55946da519c8) at vm_eval.c:102
#29 vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6c9ae0, ci=ci@entry=0x7ffd6f6c9ad0,
cc=cc@entry=0x7ffd6f6c9b00, argv=argv@entry=0x7ffd6f6c9ba0) at vm_eval.c:133
#30 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6c9ba0, argc=1, id=4849, recv=94096115734160, ec=0x55946da519c8) at vm_eval.c:60
#31 rb_call0 (ec=0x55946da519c8, recv=94096115734160, mid=4849, mid@entry=94096135960920, argc=argc@entry=1,
argv=argv@entry=0x7ffd6f6c9ba0, scope=scope@entry=CALL_FCALL, self=94096135960920) at vm_eval.c:302
#32 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x7ffd6f6c9ba0, argc=1, mid=94096135960920, recv=) at vm_eval.c:595
#33 rb_funcallv (recv=, mid=mid@entry=4849, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6c9ba0) at vm_eval.c:825
#34 0x000055946cd848a7 in rb_mod_include (argc=0, argv=0x7fc7bdb4fce8, module=94096135960920) at eval.c:1203
#35 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594710c06b0, cc=, calling=, reg_cfp=0x7fc7bdc4df00, ec=0x55946da519c8) at
vm_inshelper.c:1928
#36 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4df00, calling=, ci=0x5594710c06b0, cc=) at vm_inshelper.c:1944
#37 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at
/tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#38 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#39 0x000055946cf035fc in invoke_block (captured=0x7ffd6f6ca0a0, opt_pc=, type=, cref=0x559476c23930, self=94096135960920,
iseq=0x559471335ee0, ec=0x55946da519c8) at vm.c:1005
#40 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7ffd6f6ca0a0, self=94096135960920, argc=, argv=, passed_block_handler=0,
cref=0x559476c23930, is_lambda=0) at vm.c:1057
#41 0x000055946cf04520 in invoke_block_from_c_bh (ec=ec@entry=0x55946da519c8, block_handler=, argc=argc@entry=1,
argv=argv@entry=0x7ffd6f6ca108, cref=, is_lambda=, is_lambda@entry=0, force_blockarg=0, passed_block_handler=0) at vm.c:1075
#42 0x000055946cf04958 in vm_yield_with_cref (is_lambda=0, cref=, argv=0x7ffd6f6ca108, argc=1, ec=0x55946da519c8) at vm.c:1112
#43 yield_under (under=94096135960920, self=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108) at vm_eval.c:1572
#44 0x000055946cf04b12 in rb_mod_module_exec (argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108, mod=) at vm_eval.c:1770
#45 0x000055946ce00fc6 in rb_mod_initialize (module=94096135960920) at object.c:1978
#46 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6ca130, cc=0x7fc7bdb4fc98, argv=0x7fc7bdb4fc98, calling=0x7ffd6f6ca140,
ec=0x55946da519c8) at vm_eval.c:87
#47 vm_call0_cfunc (argv=0x7fc7bdb4fc98, cc=0x7fc7bdb4fc98, ci=0x7ffd6f6ca130, calling=0x7ffd6f6ca140, ec=0x55946da519c8) at vm_eval.c:102
#48 vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6ca1f0, ci=ci@entry=0x7ffd6f6ca1e0,
cc=cc@entry=0x7ffd6f6ca210, argv=argv@entry=0x7fc7bdb4fc98) at vm_eval.c:133
#49 0x000055946cf074b2 in vm_call0 (me=, argv=0x7fc7bdb4fc98, argc=0, id=3057, recv=94096135960920, ec=0x55946da519c8) at vm_eval.c:60
#50 rb_call0 (ec=0x55946da519c8, recv=recv@entry=94096135960920, mid=mid@entry=3057, argc=argc@entry=3057, argv=argv@entry=0x0,
scope=scope@entry=CALL_FCALL, self=94095983466120) at vm_eval.c:302
#51 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x0, argc=3057, mid=3057, recv=94096135960920) at vm_eval.c:595
```

#52 rb_funcallv (recv=recv@entry=94096135960920, mid=mid@entry=3057, argc=argc@entry=0, argv=argv@entry=0x7fc7bdb4fc98) at vm_eval.c:825
#53 0x000055946cd89673 in rb_obj_call_init (obj=obj@entry=94096135960920, argc=argc@entry=0, argv=argv@entry=0x7fc7bdb4fc98) at eval.c:1590
#54 0x000055946ce048a1 in rb_class_s_new (argc=0, argv=0x7fc7bdb4fc98, klass=) at object.c:2153
#55 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x559471339c80, cc=, calling=, reg_cfp=0x7fc7bdc4dfa8, ec=0x55946da519c8) at vm_inshelper.c:1928
#56 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4dfa8, calling=, ci=0x559471339c80, cc=) at vm_inshelper.c:1944
#57 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#58 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#59 0x000055946cf03274 in invoke_bmethod (ec=ec@entry=0x55946da519c8, iseq=iseq@entry=0x55946f15cbc0, self=self@entry=94096115734640, me=me@entry=0x559475664e28, type=type@entry=572653825, opt_pc=0, captured=0x55947158e8a0) at vm.c:1026
#60 0x000055946cf03534 in invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x55947158e8a0, self=94096115734640, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=1) at vm.c:1060
#61 0x000055946cf036ce in invoke_block_from_c_proc (ec=, proc=, self=, argc=1, argv=, passed_block_handler=, is_lambda=) at vm.c:1150
#62 0x000055946cf03811 in vm_invoke_bmethod (block_handler=, argv=, argc=1, self=, proc=, ec=0x55946da519c8) at vm.c:1175
#63 vm_call_bmethod_body (ci=, cc=0x55946f625960, argv=, calling=0x7ffd6f6ca9b0, ec=0x55946da519c8) at vm_inshelper.c:1971
#64 vm_call_bmethod (ec=0x55946da519c8, cfp=0x7fc7bdc4e0c0, calling=0x7ffd6f6ca9b0, ci=, cc=0x55946f625960) at vm_inshelper.c:1988
#65 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e0c0, calling=, ci=, cc=) at vm_inshelper.c:2417
#66 0x000055946cf0e63e in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:797
#67 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#68 0x000055946cf06767 in vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6cad00, ci=ci@entry=0x7ffd6f6cacf0, cc=cc@entry=0x7ffd6f6cad20, argv=argv@entry=0x7ffd6f6cadb0) at vm_eval.c:129
#69 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6cadb0, argc=1, id=3681, recv=94096115734640, ec=0x55946da519c8) at vm_eval.c:60
#70 rb_call0 (ec=0x55946da519c8, recv=recv@entry=94096115734640, mid=3681, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cad90, scope=scope@entry=CALL_FCALL, self=94095983452880) at vm_eval.c:302
#71 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x7ffd6f6cad90, argc=1, mid=, recv=94096115734640) at vm_eval.c:595
#72 rb_funcallv (recv=recv@entry=94096115734640, mid=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cadb0) at vm_eval.c:825
#73 0x000055946cf477d2 in rb_class_inherited (super=super@entry=94096115734640, klass=klass@entry=94096135961440) at class.c:625
#74 0x000055946cf0f796 in vm_declare_class (super=, cbase=94096094868200, flags=, id=847387) at vm_inshelper.c:3134
#75 vm_define_class (super=, cbase=, flags=, id=847387) at vm_inshelper.c:3167
#76 vm_find_or_create_class_by_id (super=, cbase=, flags=, id=847387) at vm_inshelper.c:3196
#77 vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:685
#78 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#79 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x559471f1b548) at vm.c:2046
#80 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94096135872160, wrap=wrap@entry=0) at load.c:611
#81 0x000055946cdd41f1 in rb_require_internal (fname=94096135872400, fname@entry=94096135872440, safe=0) at load.c:992
#82 0x000055946cdd4493 in rb_require_safe (safe=, fname=94096135872440) at load.c:1038
#83 rb_f_require (obj=, fname=94096135872440) at load.c:820
#84 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594708700a0, cc=, calling=, reg_cfp=0x7fc7bdc4e168, ec=0x55946da519c8) at vm_inshelper.c:1928
#85 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e168, calling=, ci=0x5594708700a0, cc=) at vm_inshelper.c:1944
#86 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e168, calling=, ci=, cc=) at vm_inshelper.c:2417
#87 0x000055946cf0e63e in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:797
#88 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#89 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4e490, opt_pc=, type=, cref=0x0, self=94096096426480, iseq=0x55946e49f0b8, ec=0x55946da519c8) at vm.c:1005
#90 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4e490, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#91 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#92 vm_yield (argc=1, argv=0x7ffd6f6cb938, ec=) at vm.c:1120
#93 rb_yield_0 (argv=0x7ffd6f6cb938, argc=1) at vm_eval.c:980
#94 rb_yield_1 (val=94096130169040) at vm_eval.c:986
#95 rb_yield (val=) at vm_eval.c:996
#96 0x000055946cf2113d in rb_ary_each (ary=94096076222560) at array.c:1820
#97 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946ed0d090, cc=, calling=, reg_cfp=0x7fc7bdc4e478, ec=0x55946da519c8) at vm_inshelper.c:1928
#98 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e478, calling=, ci=0x55946ed0d090, cc=) at vm_inshelper.c:1944
#99 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#100 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#101 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4e500, opt_pc=, type=, cref=0x0, self=94096096426480, iseq=0x55946e49f298, ec=0x55946da519c8) at vm.c:1005
#102 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4e500, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#103 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#104 vm_yield (argc=1, argv=0x7ffd6f6cbdf8, ec=) at vm.c:1120
#105 rb_yield_0 (argv=0x7ffd6f6cbdf8, argc=1) at vm_eval.c:980

#106 rb_yield_1 (val=94096095502480) at vm_eval.c:986
#107 rb_yield (val=) at vm_eval.c:996
#108 0x000055946cf2113d in rb_ary_each (ary=94096095328480) at array.c:1820
#109 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946e8552a0, cc=, calling=, reg_cfp=0x7fc7bdc4e4e8, ec=0x55946da519c8) at vm_inshelper.c:1928
#110 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e4e8, calling=, ci=0x55946e8552a0, cc=) at vm_inshelper.c:1944
#111 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#112 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#113 0x000055946cf06767 in vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7fd6f6cc2d0, ci=ci@entry=0x7fd6f6cc2c0, cc=cc@entry=0x7fd6f6cc2f0, argv=argv@entry=0x7fd6f6cc390) at vm_eval.c:129
#114 0x000055946cf074b2 in vm_call0 (me=, argv=0x7fd6f6cc390, argc=0, id=135807, recv=94096096186440, ec=0x55946da519c8) at vm_eval.c:60
#115 rb_call0 (ec=0x55946da519c8, recv=94096096186440, mid=135807, argc=, argv=argv@entry=0x8, scope=scope@entry=CALL_PUBLIC, self=94095993048320) at vm_eval.c:302
#116 0x000055946cf0a31a in rb_call (scope=CALL_PUBLIC, argv=0x8, argc=, mid=, recv=) at vm_eval.c:595
#117 rb_funcall_with_block (recv=, mid=, argc=argc@entry=0, argv=argv@entry=0x7fd6f6cc390, passed_procval=passed_procval@entry=8) at vm_eval.c:857
#118 0x000055946ceb319c in rb_sym_proc_call (mid=, argc=argc@entry=1, argv=argv@entry=0x7fd6f6cc388, passed_proc=passed_proc@entry=8) at string.c:10480
#119 0x000055946cf0477c in vm_yield_with_symbol (block_handler=0, argv=0x7fd6f6cc388, argc=1, symbol=, ec=) at vm_inshelper.c:2573
#120 invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1084
#121 vm_yield (argc=1, argv=0x7fd6f6cc388, ec=) at vm.c:1120
#122 rb_yield_0 (argv=0x7fd6f6cc388, argc=1) at vm_eval.c:980
#123 rb_yield_1 (val=94096096186440) at vm_eval.c:986
#124 rb_yield (val=) at vm_eval.c:996
#125 0x000055946cf2113d in rb_ary_each (ary=94095993048320) at array.c:1820
#126 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x5594744d8280, cc=, calling=, reg_cfp=0x7fc7bdc4e590, ec=0x55946da519c8) at vm_inshelper.c:1928
#127 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e590, calling=, ci=0x5594744d8280, cc=) at vm_inshelper.c:1944
#128 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e590, calling=, ci=, cc=) at vm_inshelper.c:2417
#129 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#130 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#131 0x000055946cf035fc in invoke_block (captured=0x7fd6f6cc8e0, opt_pc=, type=, cref=0x559474604128, self=94096096426480, iseq=0x5594745027e8, ec=0x55946da519c8) at vm.c:1005
#132 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fd6f6cc8e0, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x559474604128, is_lambda=0) at vm.c:1057
#133 0x000055946cf04520 in invoke_block_from_c_bh (ec=ec@entry=0x55946da519c8, block_handler=, argc=argc@entry=1, argv=argv@entry=0x7fc7bdb4f7f8, cref=, is_lambda=, is_lambda@entry=0, force_blockarg=0, passed_block_handler=0) at vm.c:1075
#134 0x000055946cf04958 in vm_yield_with_cref (is_lambda=0, cref=, argv=0x7fc7bdb4f7f8, argc=1, ec=0x55946da519c8) at vm.c:1112
#135 yield_under (under=94096093646040, self=, argc=1, argv=0x7fc7bdb4f7f8) at vm_eval.c:1572
#136 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946e5cd230, cc=, calling=, reg_cfp=0x7fc7bdc4e600, ec=0x55946da519c8) at vm_inshelper.c:1928
#137 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e600, calling=, ci=0x55946e5cd230, cc=) at vm_inshelper.c:1944
#138 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#139 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#140 0x000055946cf035fc in invoke_block (captured=0x5594744c4fb0, opt_pc=, type=, cref=0x0, self=94095990659320, iseq=0x55946e192aa0, ec=0x55946da519c8) at vm.c:1005
#141 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x5594744c4fb0, self=94095990659320, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#142 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#143 vm_yield (argc=1, argv=0x7fd6f6ccdc8, ec=) at vm.c:1120
#144 rb_yield_0 (argv=0x7fd6f6ccdc8, argc=1) at vm_eval.c:980
#145 rb_yield_1 (val=94096094867920) at vm_eval.c:986
#146 rb_yield (val=) at vm_eval.c:996
#147 0x000055946cf2113d in rb_ary_each (ary=94096094867160) at array.c:1820
#148 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7fd6f6cce00, cc=0x7fd6f6cce70, argv=0x7fc7bdb4f6b8, calling=0x7fd6f6cce50, ec=0x55946da519c8) at vm_eval.c:87
#149 vm_call0_cfunc (argv=0x7fc7bdb4f6b8, cc=0x7fd6f6cce70, ci=0x7fd6f6cce00, calling=0x7fd6f6cce50, ec=0x55946da519c8) at vm_eval.c:102
#150 vm_call0_body (ec=0x55946da519c8, calling=calling@entry=0x7fd6f6cce00, ci=ci@entry=0x7fd6f6cceb0, cc=cc@entry=0x7fd6f6cccee0, argv=0x7fc7bdb4f6b8) at vm_eval.c:133
#151 0x000055946cf06c50 in vm_call0 (me=, argv=, argc=, id=, recv=, ec=) at vm_eval.c:60
#152 rb_vm_call (ec=, recv=, id=, argc=, argv=, me=) at vm_eval.c:209
#153 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946dba2780, cc=, calling=, reg_cfp=0x7fc7bdc4e7c0, ec=0x55946da519c8) at vm_inshelper.c:1928
#154 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e7c0, calling=, ci=0x55946dba2780, cc=) at vm_inshelper.c:1944
#155 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e7c0, calling=, ci=, cc=) at vm_inshelper.c:2417
#156 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#157 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#158 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946dfed3a8) at vm.c:2046
#159 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095988939080, wrap=wrap@entry=0) at

load.c:611
#160 0x000055946cdd41f1 in rb_require_internal (fname=94095988939160, fname@entry=94095988939200, safe=0) at load.c:992
#161 0x000055946cdd4493 in rb_require_safe (safe=, fname=94095988939200) at load.c:1038
#162 rb_f_require (obj=, fname=94095988939200) at load.c:820
#163 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946dffa920, cc=, calling=, reg_cfp=0x7fc7bdc4e948, ec=0x55946da519c8) at vm_inshelper.c:1928
#164 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e948, calling=, ci=0x55946dffa920, cc=) at vm_inshelper.c:1944
#165 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e948, calling=, ci=, cc=) at vm_inshelper.c:2417
#166 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#167 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#168 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946dfee438) at vm.c:2046
#169 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993039280, wrap=wrap@entry=0) at load.c:611
#170 0x000055946cdd41f1 in rb_require_internal (fname=fname@entry=94095993043840, safe=0) at load.c:992
#171 0x000055946cdd4493 in rb_require_safe (safe=, fname=94095993043840) at load.c:1038
#172 rb_f_require (obj=, fname=94095993043840) at load.c:820
#173 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946ea12160, cc=, calling=, reg_cfp=0x7fc7bdc4e9b8, ec=0x55946da519c8) at vm_inshelper.c:1928
#174 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e9b8, calling=, ci=0x55946ea12160, cc=) at vm_inshelper.c:1944
#175 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e9b8, calling=, ci=, cc=) at vm_inshelper.c:2417
#176 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#177 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#178 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e3d7c48) at vm.c:2046
#179 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095988920840, wrap=) at load.c:611
#180 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095988920840) at load.c:642
#181 rb_f_load (argc=, argv=) at load.c:710
#182 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e7c96e0, cc=, calling=, reg_cfp=0x7fc7bdc4ea28, ec=0x55946da519c8) at vm_inshelper.c:1928
#183 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ea28, calling=, ci=0x55946e7c96e0, cc=) at vm_inshelper.c:1944
#184 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ea28, calling=, ci=, cc=) at vm_inshelper.c:2417
#185 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#186 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#187 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4eae8, opt_pc=, type=, cref=0x0, self=9409598891400, iseq=0x55946e4dae10, ec=0x55946da519c8) at vm.c:1005
#188 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4eae8, self=9409598891400, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#189 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#190 vm_yield (argc=1, argv=0x7fd6f6ce8e8, ec=) at vm.c:1120
#191 rb_yield_0 (argv=0x7fd6f6ce8e8, argc=1) at vm_eval.c:980
#192 rb_yield_1 (val=94095988924840) at vm_eval.c:986
#193 rb_yield (val=) at vm_eval.c:996
#194 0x000055946cf2113d in rb_ary_each (ary=94095988920960) at array.c:1820
#195 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e421000, cc=, calling=, reg_cfp=0x7fc7bdc4ead0, ec=0x55946da519c8) at vm_inshelper.c:1928
#196 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ead0, calling=, ci=0x55946e421000, cc=) at vm_inshelper.c:1944
#197 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ead0, calling=, ci=, cc=) at vm_inshelper.c:2417
#198 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#199 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#200 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e35f2c0) at vm.c:2046
#201 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095992643280, wrap=) at load.c:611
#202 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095992643280) at load.c:642
#203 rb_f_load (argc=, argv=) at load.c:710
#204 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e24bb20, cc=, calling=, reg_cfp=0x7fc7bdc4ec58, ec=0x55946da519c8) at vm_inshelper.c:1928
#205 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ec58, calling=, ci=0x55946e24bb20, cc=) at vm_inshelper.c:1944
#206 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ec58, calling=, ci=, cc=) at vm_inshelper.c:2417
#207 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#208 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#209 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e3a7390) at vm.c:2046
#210 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993616120, wrap=) at load.c:611
#211 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095993616120) at load.c:642
#212 rb_f_load (argc=, argv=) at load.c:710
#213 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e6a00d0, cc=, calling=, reg_cfp=0x7fc7bdc4ecc8, ec=0x55946da519c8) at vm_inshelper.c:1928
#214 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ecc8, calling=, ci=0x55946e6a00d0, cc=) at vm_inshelper.c:1944
#215 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ecc8, calling=, ci=, cc=) at vm_inshelper.c:2417
#216 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#217 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804

#218 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e470100) at vm.c:2046
#219 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993672200, wrap=) at load.c:611
#220 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095993672200) at load.c:642
#221 rb_f_load (argc=, argv=) at load.c:710
#222 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e3306a0, cc=, calling=, reg_cfp=0x7fc7bdc4efa0, ec=0x55946da519c8) at vm_inshelper.c:1928
#223 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4efa0, calling=, ci=0x55946e3306a0, cc=) at vm_inshelper.c:1944
#224 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4efa0, calling=, ci=, cc=) at vm_inshelper.c:2417
#225 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#226 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#227 0x000055946cf119d5 in rb_iseq_eval_main (iseq=iseq@entry=0x55946e4bbba0) at vm.c:2057
#228 0x000055946cd83d54 in ruby_exec_internal (n=0x55946e4bbba0) at eval.c:247
#229 0x000055946cd87fdf in ruby_exec_node (n=0x55946e4bbba0) at eval.c:311
#230 ruby_run_node (n=) at eval.c:303
#231 0x000055946cd831bf in main (argc=22, argv=0x7ffd6f6d0148) at ./main.c:42

Revision 62433 - 02/16/2018 02:45 PM - k0kubun (Takashi Kokubun)

mjit.c: fix deadlock on class serial increment

This is reported by @hasimo. Fixing a case like this:

#0 ill_lock_wait () at ../sysdeps/unix/sysv/linux/x86_64/lowlevellock.S:135
#1 0x00007fc7bd824dbd in __GI_pthread_mutex_lock (mutex=mutex@entry=0x55946d294440) at ../nptl/pthread_mutex_lock.c:80 4
#2 0x000055946cec54d9 in rb_native_mutex_lock (lock=lock@entry=0x55946d294440) at thread_pthread.c:211
#3 0x000055946cde10ca in CRITICAL_SECTION_START (msg=0x55946cfb5423 "mjit_gc_start_hook", level=4) at mjit.c:392
#4 mjit_gc_start_hook () at mjit.c:412
#5 0x000055946cda0dfe in gc_enter (event=0x55946cfaf91e "gc_rest", objspace=0x55946da51760) at gc.c:6623
#6 gc_rest (objspace=objspace@entry=0x55946da51760) at gc.c:6515
#7 0x000055946cd9f1cf in gc_rest (objspace=0x55946da51760) at gc.c:7841
#8 objspace_malloc_increase (objspace=objspace@entry=0x55946da51760, new_size=, old_size=old_size@entry=0, type=type@entry=MEMOP_TYPE_MALLOC, mem=0x7fc7a4439010) at gc.c:7842
#9 0x000055946cda1706 in objspace_malloc_fixup (size=, mem=0x7fc7a4439010, objspace=0x55946da51760) at gc.c:7910
#10 objspace_xmalloc0 (objspace=0x55946da51760, size=, size@entry=3145728) at gc.c:7939
#11 0x000055946cda3620 in ruby_xmalloc0 (size=3145728) at gc.c:8006
#12 ruby_xmalloc (size=size@entry=3145728) at gc.c:8015
#13 0x000055946ce93f4c in st_init_table_with_size (type=0x55946d28da30, size=) at st.c:602
#14 0x000055946ce94287 in rebuild_table (tab=tab@entry=0x55946db669f0) at st.c:777
#15 0x000055946ce963f7 in rebuild_table_if_necessary (tab=0x55946db669f0) at st.c:1139
#16 st_add_direct_with_hash (hash=8577035585096733536, value=20, key=808451, tab=0x55946db669f0) at st.c:1207
#17 st_update (tab=0x55946db669f0, key=key@entry=808451, func=, arg=140726472841392) at st.c:1512
#18 0x000055946cda9e27 in tbl_update (optional_arg=, func=, key=, hash=) at hash.c:561
#19 rb_hash_aset (hash=94095983218480, key=key@entry=808451, val=val@entry=20) at hash.c:1654
#20 0x000055946cde243a in mjit_add_class_serial (class_serial=class_serial@entry=404225) at mjit.c:1414 3
#21 0x000055946cefcfab in rb_next_class_serial () at vm.c:321
#22 0x000055946cf48324 in class_alloc (klass=, flags=28) at class.c:178
#23 rb_include_class_new (module=module@entry=94096115733840, super=0) at class.c:820
#24 0x000055946cf487ac in include_modules_at (klass=klass@entry=94096135960920, c=, module=, module@entry=94096115734160, search_super=search_super@entry=1) at class.c:913
#25 0x000055946cf48ac8 in rb_include_module (klass=94096135960920, module=module@entry=94096115734160) at class.c:870
#26 0x000055946cd84993 in rb_mod_append_features (module=94096115734160, include=) at eval.c:1178
#27 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6c9a20, cc=0x7ffd6f6c9ba0, argv=0x7ffd6f6c9ba0, ec=0x55946da519c8) at vm_eval.c:87
#28 vm_call0_cfunc (argv=0x7ffd6f6c9ba0, cc=0x7ffd6f6c9ba0, ci=0x7ffd6f6c9a20, calling=0x7ffd6f6c9a30, ec=0x55946da519c8) at vm_eval.c:102
#29 vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6c9ae0, ci=ci@entry=0x7ffd6f6c9ad0, cc=cc@entry=0x7ffd6f6c9b00, argv=argv@entry=0x7ffd6f6c9ba0) at vm_eval.c:133
#30 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6c9ba0, argc=1, id=4849, recv=94096115734160, ec=0x55946da519c8) at vm_eval.c:60
#31 rb_call0 (ec=0x55946da519c8, recv=94096115734160, mid=4849, mid@entry=94096135960920, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6c9ba0, scope=scope@entry=CALL_FCALL, self=94096135960920) at vm_eval.c:302
#32 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x7ffd6f6c9ba0, argc=1, mid=94096135960920, recv=) at vm_eval.c:595
#33 rb_funcallv (recv=, mid=mid@entry=4849, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6c9ba0) at vm_eval.c:825
#34 0x000055946cd848a7 in rb_mod_include (argc=0, argv=0x7fc7bdb4fce8, module=94096135960920) at eval.c:1203
#35 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594710c06b0, cc=, calling=, reg_cfp=0x7fc7bdc4df00, ec=0x55946da519c8) at vm_inshelper.c:1928
#36 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4df00, calling=, ci=0x5594710c06b0, cc=) at vm_inshelper.c:1944
#37 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#38 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#39 0x000055946cf035fc in invoke_block (captured=0x7ffd6f6ca0a0, opt_pc=, type=, cref=0x559476c23930, self=94096135960920, iseq=0x559471335ee0, ec=0x55946da519c8) at vm.c:1005
#40 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7ffd6f6ca0a0, self=94096135960920, argc=, argv=, passed_block_handler=0, cref=0x559476c23930, is_lambda=0) at vm.c:1057
#41 0x000055946cf04520 in invoke_block_from_c_bh (ec=ec@entry=0x55946da519c8, block_handler=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108, cref=, is_lambda=, is_lambda@entry=0, force_blockarg=0, passed_block_handler=0) at vm.c:1075

#42 0x000055946cf04958 in vm_yield_with_cref (is_lambda=0, cref=, argv=0x7ffd6f6ca108, argc=1, ec=0x55946da519c8) at vm.c:1112
#43 yield_under (under=94096135960920, self=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108) at vm_eval.c:1572
#44 0x000055946cf04b12 in rb_mod_module_exec (argc=argc@entry=1, argv=argv@entry=0x7ffd6f6ca108, mod=) at vm_eval.c:1770
#45 0x000055946ce00fc6 in rb_mod_initialize (module=94096135960920) at object.c:1978
#46 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7ffd6f6ca130, cc=0x7fc7bdb4fc98, argv=0x7fc7bdb4fc98, calling=0x7ffd6f6ca140, ec=0x55946da519c8) at vm_eval.c:87
#47 vm_call0_cfunc (argv=0x7fc7bdb4fc98, cc=0x7fc7bdb4fc98, ci=0x7ffd6f6ca130, calling=0x7ffd6f6ca140, ec=0x55946da519c8) at vm_eval.c:102
#48 vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6ca1f0, ci=ci@entry=0x7ffd6f6ca1e0, cc=cc@entry=0x7ffd6f6ca210, argv=argv@entry=0x7fc7bdb4fc98) at vm_eval.c:133
#49 0x000055946cf074b2 in vm_call0 (me=, argv=0x7fc7bdb4fc98, argc=0, id=3057, recv=94096135960920, ec=0x55946da519c8) at vm_eval.c:60
#50 rb_call0 (ec=0x55946da519c8, recv=recv@entry=94096135960920, mid=mid@entry=3057, argc=argc@entry=3057, argv=argv@entry=0x0, scope=scope@entry=CALL_FCALL, self=94095983466120) at vm_eval.c:302
#51 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x0, argc=3057, mid=3057, recv=94096135960920) at vm_eval.c:595
#52 rb_funcallv (recv=recv@entry=94096135960920, mid=mid@entry=3057, argc=argc@entry=0, argv=argv@entry=0x7fc7bdb4fc98) at vm_eval.c:825
#53 0x000055946cd89673 in rb_obj_call_init (obj=obj@entry=94096135960920, argc=argc@entry=0, argv=argv@entry=0x7fc7bdb4fc98) at eval.c:1590
#54 0x000055946ce048a1 in rb_class_s_new (argc=0, argv=0x7fc7bdb4fc98, klass=) at object.c:2153
#55 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x559471339c80, cc=, calling=, reg_cfp=0x7fc7bdc4dfa8, ec=0x55946da519c8) at vm_inshelper.c:1928
#56 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4dfa8, calling=, ci=0x559471339c80, cc=) at vm_inshelper.c:1944
#57 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#58 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#59 0x000055946cf03274 in invoke_bmethod (ec=ec@entry=0x55946da519c8, iseq=iseq@entry=0x55946f15cbc0, self=self@entry=94096115734640, me=me@entry=0x559475664e28, type=type@entry=572653825, opt_pc=0, captured=0x55947158e8a0) at vm.c:1026
#60 0x000055946cf03534 in invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x55947158e8a0, self=94096115734640, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=1) at vm.c:1060
#61 0x000055946cf036ce in invoke_block_from_c_proc (ec=, proc=, self=, argc=1, argv=, passed_block_handler=, is_lambda=) at vm.c:1150
#62 0x000055946cf03811 in vm_invoke_bmethod (block_handler=, argv=, argc=1, self=, proc=, ec=0x55946da519c8) at vm.c:1175
#63 vm_call_bmethod_body (ci=, cc=0x55946f625960, argv=, calling=0x7ffd6f6ca9b0, ec=0x55946da519c8) at vm_inshelper.c:1971
#64 vm_call_bmethod (ec=0x55946da519c8, cfp=0x7fc7bdc4e0c0, calling=0x7ffd6f6ca9b0, ci=, cc=0x55946f625960) at vm_inshelper.c:1988
#65 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e0c0, calling=, ci=, cc=) at vm_inshelper.c:2417
#66 0x000055946cf0e63e in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:797
#67 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#68 0x000055946cf06767 in vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7ffd6f6cad00, ci=ci@entry=0x7ffd6f6cacf0, cc=cc@entry=0x7ffd6f6cad20, argv=argv@entry=0x7ffd6f6cadb0) at vm_eval.c:129
#69 0x000055946cf074b2 in vm_call0 (me=, argv=0x7ffd6f6cadb0, argc=1, id=3681, recv=94096115734640, ec=0x55946da519c8) at vm_eval.c:60
#70 rb_call0 (ec=0x55946da519c8, recv=recv@entry=94096115734640, mid=3681, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cad90, scope=scope@entry=CALL_FCALL, self=94095983452880) at vm_eval.c:302
#71 0x000055946cf07b9b in rb_call (scope=CALL_FCALL, argv=0x7ffd6f6cad90, argc=1, mid=, recv=94096115734640) at vm_eval.c:595
#72 rb_funcallv (recv=recv@entry=94096115734640, mid=, argc=argc@entry=1, argv=argv@entry=0x7ffd6f6cadb0) at vm_eval.c:825
#73 0x000055946cf477d2 in rb_class_inherited (super=super@entry=94096115734640, klass=klass@entry=94096135961440) at class.c:625
#74 0x000055946cf0f796 in vm_declare_class (super=, cbase=94096094868200, flags=, id=847387) at vm_inshelper.c:3134
#75 vm_define_class (super=, cbase=, flags=, id=847387) at vm_inshelper.c:3167
#76 vm_find_or_create_class_by_id (super=, cbase=, flags=, id=847387) at vm_inshelper.c:3196
#77 vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:685
#78 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#79 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x559471f1b548) at vm.c:2046
#80 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94096135872160, wrap=wrap@entry=0) at load.c:611
#81 0x000055946cdd41f1 in rb_require_internal (fname=94096135872400, fname@entry=94096135872440, safe=0) at load.c:992
#82 0x000055946cdd4493 in rb_require_safe (safe=, fname=94096135872440) at load.c:1038
#83 rb_f_require (obj=, fname=94096135872440) at load.c:820
#84 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x5594708700a0, cc=, calling=, reg_cfp=0x7fc7bdc4e168, ec=0x55946da519c8) at vm_inshelper.c:1928
#85 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e168, calling=, ci=0x5594708700a0, cc=) at vm_inshelper.c:1944
#86 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e168, calling=, ci=, cc=) at vm_inshelper.c:2417
#87 0x000055946cf0e63e in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:797
#88 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#89 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4e490, opt_pc=, type=, cref=0x0, self=94096096426480, iseq=0x55946e49f0b8, ec=0x55946da519c8) at vm.c:1005
#90 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4e490, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#91 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#92 vm_yield (argc=1, argv=0x7ffd6f6cb938, ec=) at vm.c:1120
#93 rb_yield_0 (argv=0x7ffd6f6cb938, argc=1) at vm_eval.c:980
#94 rb_yield_1 (val=94096130169040) at vm_eval.c:986
#95 rb_yield (val=) at vm_eval.c:996
#96 0x000055946cf2113d in rb_ary_each (ary=94096076222560) at array.c:1820
#97 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946ed0d090, cc=, calling=, reg_cfp=0x7fc7bdc4e478, ec=0x55946da519c8) at

vm_inshelper.c:1928
#98 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e478, calling=, ci=0x55946ed0d090, cc=) at vm_inshelper.c:1944
#99 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#100 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#101 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4e500, opt_pc=, type=, cref=0x0, self=94096096426480, iseq=0x55946e49f298, ec=0x55946da519c8) at vm.c:1005
#102 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4e500, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#103 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#104 vm_yield (argc=1, argv=0x7fd6f6cbdf8, ec=) at vm.c:1120
#105 rb_yield_0 (argv=0x7fd6f6cbdf8, argc=1) at vm_eval.c:980
#106 rb_yield_1 (val=94096095502480) at vm_eval.c:986
#107 rb_yield (val=) at vm_eval.c:996
#108 0x000055946cf2113d in rb_ary_each (ary=94096095328480) at array.c:1820
#109 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946e8552a0, cc=, calling=, reg_cfp=0x7fc7bdc4e4e8, ec=0x55946da519c8) at vm_inshelper.c:1928
#110 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e4e8, calling=, ci=0x55946e8552a0, cc=) at vm_inshelper.c:1944
#111 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#112 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#113 0x000055946cf06767 in vm_call0_body (ec=ec@entry=0x55946da519c8, calling=calling@entry=0x7fd6f6cc2d0, ci=ci@entry=0x7fd6f6cc2c0, cc=cc@entry=0x7fd6f6cc2f0, argv=argv@entry=0x7fd6f6cc390) at vm_eval.c:129
#114 0x000055946cf074b2 in vm_call0 (me=, argv=0x7fd6f6cc390, argc=0, id=135807, recv=94096096186440, ec=0x55946da519c8) at vm_eval.c:60
#115 rb_call0 (ec=0x55946da519c8, recv=94096096186440, mid=135807, argc=, argv=argv@entry=0x8, scope=scope@entry=CALL_PUBLIC, self=94095993048320) at vm_eval.c:302
#116 0x000055946cf0a31a in rb_call (scope=CALL_PUBLIC, argv=0x8, argc=, mid=, recv=) at vm_eval.c:595
#117 rb_funcall_with_block (recv=, mid=, argc=argc@entry=0, argv=argv@entry=0x7fd6f6cc390, passed_procval=passed_procval@entry=8) at vm_eval.c:857
#118 0x000055946ceb319c in rb_sym_proc_call (mid=, argc=argc@entry=1, argv=argv@entry=0x7fd6f6cc388, passed_proc=passed_proc@entry=8) at string.c:10480
#119 0x000055946cf0477c in vm_yield_with_symbol (block_handler=0, argv=0x7fd6f6cc388, argc=1, symbol=, ec=) at vm_inshelper.c:2573
#120 invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1084
#121 vm_yield (argc=1, argv=0x7fd6f6cc388, ec=) at vm.c:1120
#122 rb_yield_0 (argv=0x7fd6f6cc388, argc=1) at vm_eval.c:980
#123 rb_yield_1 (val=94096096186440) at vm_eval.c:986
#124 rb_yield (val=) at vm_eval.c:996
#125 0x000055946cf2113d in rb_ary_each (ary=94095993048320) at array.c:1820
#126 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x5594744d8280, cc=, calling=, reg_cfp=0x7fc7bdc4e590, ec=0x55946da519c8) at vm_inshelper.c:1928
#127 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e590, calling=, ci=0x5594744d8280, cc=) at vm_inshelper.c:1944
#128 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e590, calling=, ci=, cc=) at vm_inshelper.c:2417
#129 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#130 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#131 0x000055946cf035fc in invoke_block (captured=0x7fd6f6cc8e0, opt_pc=, type=, cref=0x559474604128, self=94096096426480, iseq=0x5594745027e8, ec=0x55946da519c8) at vm.c:1005
#132 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fd6f6cc8e0, self=94096096426480, argc=, argv=, passed_block_handler=0, cref=0x559474604128, is_lambda=0) at vm.c:1057
#133 0x000055946cf04520 in invoke_block_from_c_bh (ec=ec@entry=0x55946da519c8, block_handler=, argc=argc@entry=1, argv=argv@entry=0x7fc7bdb4f7f8, cref=, is_lambda=, is_lambda@entry=0, force_blockarg=0, passed_block_handler=0) at vm.c:1075
#134 0x000055946cf04958 in vm_yield_with_cref (is_lambda=0, cref=, argv=0x7fc7bdb4f7f8, argc=1, ec=0x55946da519c8) at vm.c:1112
#135 yield_under (under=94096093646040, self=, argc=1, argv=0x7fc7bdb4f7f8) at vm_eval.c:1572
#136 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946e5cd230, cc=, calling=, reg_cfp=0x7fc7bdc4e600, ec=0x55946da519c8) at vm_inshelper.c:1928
#137 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e600, calling=, ci=0x55946e5cd230, cc=) at vm_inshelper.c:1944
#138 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#139 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#140 0x000055946cf035fc in invoke_block (captured=0x5594744c4fb0, opt_pc=, type=, cref=0x0, self=94095990659320, iseq=0x55946e192aa0, ec=0x55946da519c8) at vm.c:1005
#141 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x5594744c4fb0, self=94095990659320, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#142 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#143 vm_yield (argc=1, argv=0x7fd6f6ccdc8, ec=) at vm.c:1120
#144 rb_yield_0 (argv=0x7fd6f6ccdc8, argc=1) at vm_eval.c:980
#145 rb_yield_1 (val=94096094867920) at vm_eval.c:986
#146 rb_yield (val=) at vm_eval.c:996
#147 0x000055946cf2113d in rb_ary_each (ary=94096094867160) at array.c:1820
#148 0x000055946cf06829 in vm_call0_cfunc_with_frame (ci=0x7fd6f6cce00, cc=0x7fd6f6cce70, argv=0x7fc7bdb4f6b8, calling=0x7fd6f6cce50, ec=0x55946da519c8) at vm_eval.c:87
#149 vm_call0_cfunc (argv=0x7fc7bdb4f6b8, cc=0x7fd6f6cce70, ci=0x7fd6f6cce00, calling=0x7fd6f6cce50, ec=0x55946da519c8) at vm_eval.c:102

#150 vm_call0_body (ec=0x55946da519c8, calling=calling@entry=0x7ffd6f6ccec0, ci=ci@entry=0x7ffd6f6cceb0, cc=cc@entry=0x7ffd6f6ccee0, argv=0x7fc7bdc4f6b8) at vm_eval.c:133
#151 0x000055946cf06c50 in vm_call0 (me=, argv=, argc=, id=, recv=, ec=) at vm_eval.c:60
#152 rb_vm_call (ec=, recv=, id=, argc=, argv=, me=) at vm_eval.c:209
#153 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946dba2780, cc=, calling=, reg_cfp=0x7fc7bdc4e7c0, ec=0x55946da519c8) at vm_inshelper.c:1928
#154 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e7c0, calling=, ci=0x55946dba2780, cc=) at vm_inshelper.c:1944
#155 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e7c0, calling=, ci=, cc=) at vm_inshelper.c:2417
#156 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#157 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#158 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946dfed3a8) at vm.c:2046
#159 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095988939080, wrap=wrap@entry=0) at load.c:611
#160 0x000055946cdd41f1 in rb_require_internal (fname=94095988939160, fname@entry=94095988939200, safe=0) at load.c:992
#161 0x000055946cdd4493 in rb_require_safe (safe=, fname=94095988939200) at load.c:1038
#162 rb_f_require (obj=, fname=94095988939200) at load.c:820
#163 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946dffa920, cc=, calling=, reg_cfp=0x7fc7bdc4e948, ec=0x55946da519c8) at vm_inshelper.c:1928
#164 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e948, calling=, ci=0x55946dffa920, cc=) at vm_inshelper.c:1944
#165 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e948, calling=, ci=, cc=) at vm_inshelper.c:2417
#166 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#167 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#168 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946dfee438) at vm.c:2046
#169 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993039280, wrap=wrap@entry=0) at load.c:611
#170 0x000055946cdd41f1 in rb_require_internal (fname=fname@entry=94095993043840, safe=0) at load.c:992
#171 0x000055946cdd4493 in rb_require_safe (safe=, fname=94095993043840) at load.c:1038
#172 rb_f_require (obj=, fname=94095993043840) at load.c:820
#173 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946ea12160, cc=, calling=, reg_cfp=0x7fc7bdc4e9b8, ec=0x55946da519c8) at vm_inshelper.c:1928
#174 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4e9b8, calling=, ci=0x55946ea12160, cc=) at vm_inshelper.c:1944
#175 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4e9b8, calling=, ci=, cc=) at vm_inshelper.c:2417
#176 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#177 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#178 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e3d7c48) at vm.c:2046
#179 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095988920840, wrap=) at load.c:611
#180 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095988920840) at load.c:642
#181 rb_f_load (argc=, argv=) at load.c:710
#182 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946e7c96e0, cc=, calling=, reg_cfp=0x7fc7bdc4ea28, ec=0x55946da519c8) at vm_inshelper.c:1928
#183 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ea28, calling=, ci=0x55946e7c96e0, cc=) at vm_inshelper.c:1944
#184 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ea28, calling=, ci=, cc=) at vm_inshelper.c:2417
#185 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#186 0x000055946cf02f4c in vm_exec (ec=ec@entry=0x55946da519c8) at vm.c:1804
#187 0x000055946cf035fc in invoke_block (captured=0x7fc7bdc4eae8, opt_pc=, type=, cref=0x0, self=9409598891400, iseq=0x55946e4dae10, ec=0x55946da519c8) at vm.c:1005
#188 invoke_iseq_block_from_c (ec=0x55946da519c8, captured=0x7fc7bdc4eae8, self=9409598891400, argc=, argv=, passed_block_handler=0, cref=0x0, is_lambda=0) at vm.c:1057
#189 0x000055946cf04699 in invoke_block_from_c_bh (argc=, passed_block_handler=, cref=, is_lambda=, force_blockarg=, argv=, block_handler=, ec=) at vm.c:1075
#190 vm_yield (argc=1, argv=0x7ffd6f6ce8e8, ec=) at vm.c:1120
#191 rb_yield_0 (argv=0x7ffd6f6ce8e8, argc=1) at vm_eval.c:980
#192 rb_yield_1 (val=94095988924840) at vm_eval.c:986
#193 rb_yield (val=) at vm_eval.c:996
#194 0x000055946cf2113d in rb_ary_each (ary=94095988920960) at array.c:1820
#195 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946e421000, cc=, calling=, reg_cfp=0x7fc7bdc4ead0, ec=0x55946da519c8) at vm_inshelper.c:1928
#196 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ead0, calling=, ci=0x55946e421000, cc=) at vm_inshelper.c:1944
#197 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ead0, calling=, ci=, cc=) at vm_inshelper.c:2417
#198 0x000055946cf0cb05 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at /tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:716
#199 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#200 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e35f2c0) at vm.c:2046
#201 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095992643280, wrap=) at load.c:611
#202 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095992643280) at load.c:642
#203 rb_f_load (argc=, argv=) at load.c:710
#204 0x000055946cfb61f in vm_call_cfunc_with_frame (ci=0x55946e24bb20, cc=, calling=, reg_cfp=0x7fc7bdc4ec58, ec=0x55946da519c8) at vm_inshelper.c:1928
#205 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ec58, calling=, ci=0x55946e24bb20, cc=) at vm_inshelper.c:1944
#206 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ec58, calling=, ci=, cc=) at vm_inshelper.c:2417
#207 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at

```
/tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#208 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#209 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e3a7390) at vm.c:2046
#210 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993616120, wrap=) at load.c:611
#211 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095993616120) at load.c:642
#212 rb_f_load (argc=, argv=) at load.c:710
#213 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e6a00d0, cc=, calling=, reg_cfp=0x7fc7bdc4ecc8, ec=0x55946da519c8) at
vm_inshelper.c:1928
#214 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4ecc8, calling=, ci=0x55946e6a00d0, cc=) at vm_inshelper.c:1944
#215 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4ecc8, calling=, ci=, cc=) at vm_inshelper.c:2417
#216 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at
/tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#217 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#218 0x000055946cf118d1 in rb_iseq_eval (iseq=iseq@entry=0x55946e470100) at vm.c:2046
#219 0x000055946cdd2164 in rb_load_internal0 (ec=ec@entry=0x55946da519c8, fname=fname@entry=94095993672200, wrap=) at load.c:611
#220 0x000055946cdd2850 in rb_load_internal (wrap=0, fname=94095993672200) at load.c:642
#221 rb_f_load (argc=, argv=) at load.c:710
#222 0x000055946cefb61f in vm_call_cfunc_with_frame (ci=0x55946e3306a0, cc=, calling=, reg_cfp=0x7fc7bdc4efa0, ec=0x55946da519c8) at
vm_inshelper.c:1928
#223 vm_call_cfunc (ec=0x55946da519c8, reg_cfp=0x7fc7bdc4efa0, calling=, ci=0x55946e3306a0, cc=) at vm_inshelper.c:1944
#224 0x000055946cf03ea3 in vm_call_method (ec=0x55946da519c8, cfp=0x7fc7bdc4efa0, calling=, ci=, cc=) at vm_inshelper.c:2417
#225 0x000055946cf0b5c2 in vm_exec_core (ec=ec@entry=0x55946da519c8, initial=initial@entry=0) at
/tmp/ruby-build.20180216151216.13740/ruby-trunk/insns.def:779
#226 0x000055946cf02f4c in vm_exec (ec=0x55946da519c8) at vm.c:1804
#227 0x000055946cf119d5 in rb_iseq_eval_main (iseq=iseq@entry=0x55946e4bbba0) at vm.c:2057
#228 0x000055946cd83d54 in ruby_exec_internal (n=0x55946e4bbba0) at eval.c:247
#229 0x000055946cd87fdf in ruby_exec_node (n=0x55946e4bbba0) at eval.c:311
#230 ruby_run_node (n=) at eval.c:303
#231 0x000055946cd831bf in main (argc=22, argv=0x7ffd6f6d0148) at ./main.c:42
```

History

#1 - 06/09/2008 06:10 PM - knu (Akinori MUSHYA)

```
=begin
SortedSet cannot be put in a SortedSet because it is not comparable (unsortable).
```

```
How do you define comparison between two (sorted) sets?
=end
```

#2 - 06/09/2008 06:11 PM - Anonymous

```
=begin
Issue #118 has been updated by Akinori MUSHYA.
```

```
SortedSet cannot be put in a SortedSet because it is not comparable (unsortable).
```

```
How do you define comparison between two (sorted) sets?
```

Bug [#118](#): SortedSet#inspect does not work with nested SortedSets
<http://redmine.ruby-lang.org/issues/show/118>

Author: Arthur Schreiber
Status: Open
Priority: Normal
Assigned to: Akinori MUSHYA
Category:
Target version:

```
SortedSet["a", "b", SortedSet["c"]].inspect # => raises an ArgumentError: comparison of String with SortedSet failed
```

You have received this notification because you have either subscribed to it, or are involved in it.
To change your notification preferences, please click here: <http://redmine.ruby-lang.org/my/account>

```
=end
```

#3 - 06/10/2008 05:43 PM - NoKarma (Arthur Schreiber)

```
- File set.patch added
```

```
=begin
Here is a patch that makes SortedSet#add check that any object added to a SortedSet includes the Comparable module.
It also extends the SortedSet documentation to make it more clear how SortedSet works.
```

=end

#4 - 06/10/2008 05:58 PM - Anonymous

=begin

Issue [#118](#) has been updated by Arthur Schreiber.

File set.patch added

Here is a patch that makes SortedSet#add check that any object added to a SortedSet includes the Comparable module. It also extends the SortedSet documentation to make it more clear how SortedSet works.

Bug [#118](#): SortedSet#inspect does not work with nested SortedSets
<http://redmine.ruby-lang.org/issues/show/118>

Author: Arthur Schreiber
Status: Open
Priority: Normal
Assigned to: Akinori MUSHA
Category:
Target version:

SortedSet["a", "b", SortedSet["c"]].inspect # => raises an ArgumentError: comparison of String with SortedSet failed

You have received this notification because you have either subscribed to it, or are involved in it.
To change your notification preferences, please click here: <http://redmine.ruby-lang.org/my/account>

=end

#5 - 06/10/2008 06:17 PM - knu (Akinori MUSHA)

=begin

add() is not the only method that adds new elements to a set, but merge() also is.

For merge(), you can omit the check when the given enumerable object is a SortedSet.

=end

#6 - 06/10/2008 06:24 PM - Anonymous

=begin

Issue [#118](#) has been updated by Akinori MUSHA.

add() is not the only method that adds new elements to a set, but merge() also is.

For merge(), you can omit the check when the given enumerable object is a SortedSet.

Bug [#118](#): SortedSet#inspect does not work with nested SortedSets
<http://redmine.ruby-lang.org/issues/show/118>

Author: Arthur Schreiber
Status: Open
Priority: Normal
Assigned to: Akinori MUSHA
Category:
Target version:

SortedSet["a", "b", SortedSet["c"]].inspect # => raises an ArgumentError: comparison of String with SortedSet failed

You have received this notification because you have either subscribed to it, or are involved in it.
To change your notification preferences, please click here: <http://redmine.ruby-lang.org/my/account>

=end

#7 - 06/10/2008 06:50 PM - NoKarma (Arthur Schreiber)

=begin

SortedSet#merge calls super (Set#merge), which in turn calls SortedSet#add for non-Set objects.
So the only check needed is on SortedSet#add.

=end

#8 - 06/10/2008 06:58 PM - Anonymous

=begin

Issue [#118](#) has been updated by Arthur Schreiber.

SortedSet#merge calls super (Set#merge), which in turn calls SortedSet#add for non-Set objects. So the only check needed is on SortedSet#add.

Bug [#118](#): SortedSet#inspect does not work with nested SortedSets

<http://redmine.ruby-lang.org/issues/show/118>

Author: Arthur Schreiber
Status: Open
Priority: Normal
Assigned to: Akinori MUSHA
Category:
Target version:

SortedSet["a", "b", SortedSet["c"]].inspect # => raises an ArgumentError: comparison of String with SortedSet failed

You have received this notification because you have either subscribed to it, or are involved in it.

To change your notification preferences, please click here: <http://redmine.ruby-lang.org/my/account>

=end

#9 - 06/10/2008 09:56 PM - NoKarma (Arthur Schreiber)

- File set.patch added

=begin

I attached a new patch for the set library.

I changed Set#merge to only directly use the passed objects @hash instance variable when self and the passed object are instances of the same class.

This prevents the addition of non-Comparable Objects when calling SortedSet#merge with a Set.

I hope the patch is ok now.

=end

#10 - 06/10/2008 10:04 PM - Anonymous

=begin

Issue [#118](#) has been updated by Arthur Schreiber.

File set.patch added

I attached a new patch for the set library.

I changed Set#merge to only directly use the passed objects @hash instance variable, when self and the passed object are instances of the same class.

This prevents bugs when using a SortedSet in conjunction with the rbtree library. Additionally, It prevents the merging of non-Comparable Objects when calling SortedSet#merge with a Set.

I hope the patch is ok now.

Bug [#118](#): SortedSet#inspect does not work with nested SortedSets

<http://redmine.ruby-lang.org/issues/show/118>

Author: Arthur Schreiber
Status: Open
Priority: Normal
Assigned to: Akinori MUSHA
Category:
Target version:

SortedSet["a", "b", SortedSet["c"]].inspect # => raises an ArgumentError: comparison of String with SortedSet failed

You have received this notification because you have either subscribed to it, or are involved in it.

To change your notification preferences, please click here: <http://redmine.ruby-lang.org/my/account>

=end

#11 - 05/01/2009 04:57 PM - knu (Akinori MUSHHA)

- Status changed from Open to Closed

- % Done changed from 0 to 100

=begin

Applied in changeset [r23322](#).

=end

Files

set.patch	2.17 KB	06/10/2008	NoKarma (Arthur Schreiber)
set.patch	2.58 KB	06/10/2008	NoKarma (Arthur Schreiber)