Ruby trunk - Feature #9171

[patch] use fstrings for symbol table

11/28/2013 04:55 PM - tmm1 (Aman Gupta)

Status:	Closed
Priority:	Normal
Assignee:	nobu (Nobuyoshi Nakada)
Target version:	

Description

Here is a simple patch to use fstrings for the table backing symbols.

Unfortunately it causes a segfault in test/rdoc/test_rdoc_parser_ruby.rb. Maybe someone wants to investigate.

```
diff --git a/parse.y b/parse.y
index 8207ad7..b1f3112 100644
--- a/parse.y
+++ b/parse.y
@@ -10333,7 +10333,7 @@ register_symid(ID id, const char *name, long len, rb_encoding *enc)
 static ID
 register_symid_str(ID id, VALUE str)
 {
-    OBJ_FREEZE(str);
+    str = rb_fstring(str);

     if (RUBY_DTRACE_SYMBOL_CREATE_ENABLED()) {
    RUBY_DTRACE_SYMBOL_CREATE(RSTRING_PTR(str), rb_sourcefile(), rb_sourceline());
diff --git a/string.c b/string.c
index 231bb2f..6ae33e3 100644
--- a/string.c
+++ b/string.c
@@ -138,6 +138,9 @@ rb_fstring(VALUE str)
     st_data_t fstr;
     Check_Type(str, T_STRING);

+    if (!frozen_strings)
+   frozen_strings = st_init_table(&fstring_hash_type);
+
     if (FL_TEST(str, RSTRING_FSTR))
    return str;

@@ -8707,8 +8710,6 @@ Init_String(void)
 #undef rb_intern
 #define rb_intern(str) rb_intern_const(str)

-    frozen_strings = st_init_table(&fstring_hash_type);
-
     rb_cString  = rb_define_class("String", rb_cObject);
     rb_include_module(rb_cString, rb_mComparable);
     rb_define_alloc_func(rb_cString, empty_str_alloc);
```

Associated revisions

Revision 98a74d4d - 12/08/2013 01:39 AM - tmm1 (Aman Gupta)

parse.y: use rb_fstring() for strings stored in the symbol table

- parse.y (register_symid_str): use fstrings in symbol table [Bug #9171] [ruby-core:58656]
- parse.y (rb_id2str): ditto
- string.c (rb_fstring): create frozen_strings on first usage. this allows rb_fstring() calls from the parser (before cString is created)
- string.c (fstring_set_class_i): set klass on fstrings generated before cString was defined
- string.c (Init_String): convert frozen_strings table to String objects after boot
- ext/-test-/symbol/type.c (bug_sym_id2str): expose rb_id2str()
- test/-ext-/symbol/test_type.rb (module Test_Symbol): verify symbol table entries are fstrings

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@44057 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 44057 - 12/08/2013 01:39 AM - tmm1 (Aman Gupta)

parse.y: use rb_fstring() for strings stored in the symbol table

- parse.y (register_symid_str): use fstrings in symbol table [Bug #9171] [ruby-core:58656]
- parse.y (rb_id2str): ditto
- string.c (rb_fstring): create frozen_strings on first usage. this allows rb_fstring() calls from the parser (before cString is created)
- string.c (fstring_set_class_i): set klass on fstrings generated before cString was defined
- string.c (Init_String): convert frozen_strings table to String objects after boot
- ext/-test-/symbol/type.c (bug_sym_id2str): expose rb_id2str()
- test/-ext-/symbol/test_type.rb (module Test_Symbol): verify symbol table entries are fstrings

Revision 44057 - 12/08/2013 01:39 AM - tmm1 (Aman Gupta)

parse.y: use rb_fstring() for strings stored in the symbol table

- parse.y (register_symid_str): use fstrings in symbol table [Bug #9171] [ruby-core:58656]
- parse.y (rb_id2str): ditto
- string.c (rb_fstring): create frozen_strings on first usage. this allows rb_fstring() calls from the parser (before cString is created)
- string.c (fstring_set_class_i): set klass on fstrings generated before cString was defined
- string.c (Init_String): convert frozen_strings table to String objects after boot
- ext/-test-/symbol/type.c (bug_sym_id2str): expose rb_id2str()
- test/-ext-/symbol/test_type.rb (module Test_Symbol): verify symbol table entries are fstrings

Revision 44057 - 12/08/2013 01:39 AM - tmm1 (Aman Gupta)

parse.y: use rb_fstring() for strings stored in the symbol table

- parse.y (register_symid_str): use fstrings in symbol table [Bug #9171] [ruby-core:58656]
- parse.y (rb_id2str): ditto
- string.c (rb_fstring): create frozen_strings on first usage. this allows rb_fstring() calls from the parser (before cString is created)
- string.c (fstring_set_class_i): set klass on fstrings generated before cString was defined
- string.c (Init_String): convert frozen_strings table to String objects after boot
- ext/-test-/symbol/type.c (bug_sym_id2str): expose rb_id2str()
- test/-ext-/symbol/test_type.rb (module Test_Symbol): verify symbol table entries are fstrings

Revision 44057 - 12/08/2013 01:39 AM - tmm1 (Aman Gupta)

parse.y: use rb_fstring() for strings stored in the symbol table

- parse.y (register_symid_str): use fstrings in symbol table [Bug #9171] [ruby-core:58656]
- parse.y (rb_id2str): ditto
- string.c (rb_fstring): create frozen_strings on first usage. this allows rb_fstring() calls from the parser (before cString is created)
- string.c (fstring_set_class_i): set klass on fstrings generated before cString was defined
- string.c (Init_String): convert frozen_strings table to String objects after boot
- ext/-test-/symbol/type.c (bug_sym_id2str): expose rb_id2str()
- test/-ext-/symbol/test_type.rb (module Test_Symbol): verify symbol table entries are fstrings

Revision 44057 - 12/08/2013 01:39 AM - tmm1 (Aman Gupta)

parse.y: use rb_fstring() for strings stored in the symbol table

- parse.y (register_symid_str): use fstrings in symbol table [Bug #9171] [ruby-core:58656]
- parse.y (rb_id2str): ditto
- string.c (rb_fstring): create frozen_strings on first usage. this allows rb_fstring() calls from the parser (before cString is created)
- string.c (fstring_set_class_i): set klass on fstrings generated before cString was defined
- string.c (Init_String): convert frozen_strings table to String objects after boot
- ext/-test-/symbol/type.c (bug_sym_id2str): expose rb_id2str()
- test/-ext-/symbol/test_type.rb (module Test_Symbol): verify symbol table entries are fstrings

Revision 44057 - 12/08/2013 01:39 AM - tmm1 (Aman Gupta)

parse.y: use rb_fstring() for strings stored in the symbol table

- parse.y (register_symid_str): use fstrings in symbol table [Bug #9171] [ruby-core:58656]
- parse.y (rb_id2str): ditto
- string.c (rb_fstring): create frozen_strings on first usage. this allows rb_fstring() calls from the parser (before cString is created)
- string.c (fstring_set_class_i): set klass on fstrings generated before cString was defined
- string.c (Init_String): convert frozen_strings table to String objects after boot
- ext/-test-/symbol/type.c (bug_sym_id2str): expose rb_id2str()
- test/-ext-/symbol/test_type.rb (module Test_Symbol): verify symbol table entries are fstrings

History

#1 - 11/29/2013 06:47 PM - hsbt (Hiroshi SHIBATA)

- Assignee set to ko1 (Koichi Sasada)

ko1: Could you review this?

#2 - 12/06/2013 07:39 PM - ko1 (Koichi Sasada)

- Category set to core

- Assignee changed from ko1 (Koichi Sasada) to nobu (Nobuyoshi Nakada)

Nobu, could you check it?

#3 - 12/07/2013 06:16 AM - avit (Andrew Vit)

Would this allow the symbol table to be GC'd or is that a separate issue?

#4 - 12/07/2013 09:21 AM - nobu (Nobuyoshi Nakada)

different.

#5 - 12/07/2013 09:22 AM - nobu (Nobuyoshi Nakada)

Seems my reply hasn't caught by the ITS...

(13/11/28 16:55), tmm1 (Aman Gupta) wrote:

```
    diff --git a/parse.y b/parse.y
    index 8207ad7..b1f3112 100644
    --- a/parse.y
    +++ b/parse.y
    @@ -10333,7 +10333,7 @@ register_symid(ID id, const char *name, long len, rb_encoding *enc)
     static ID
     register_symid_str(ID id, VALUE str)
     {
    -    OBJ_FREEZE(str);
    +    str = rb_fstring(str);
```

At this point, rb_cString is not created yet.
And unfrozen string will be duplicated in rb_fstring().

```
diff --git i/parse.y w/parse.y
index 8207ad7..9f580e6 100644
--- i/parse.y
+++ w/parse.y
@@ -10334,6 +10334,7 @@ static ID
 register_symid_str(ID id, VALUE str)
 {
     OBJ_FREEZE(str);
+    str = rb_fstring(str);

     if (RUBY_DTRACE_SYMBOL_CREATE_ENABLED()) {
     RUBY_DTRACE_SYMBOL_CREATE(RSTRING_PTR(str), rb_sourcefile(), rb_sourceline());
diff --git i/string.c w/string.c
index 151170c..ec43af7 100644
--- i/string.c
+++ w/string.c
@@ -165,6 +165,9 @@ rb_fstring(VALUE str)
     VALUE fstr = Qnil;
     Check_Type(str, T_STRING);

+    if (!frozen_strings)
+   frozen_strings = st_init_table(&fstring_hash_type);
+
     if (FL_TEST(str, RSTRING_FSTR))
     return str;

@@ -173,6 +176,13 @@ rb_fstring(VALUE str)
 }

 static int
+fstring_set_class_i(st_data_t key, st_data_t val, st_data_t arg)
+{
```

```
+    RBASIC_SET_CLASS((VALUE)key, (VALUE)arg);
+    return ST_CONTINUE;
+}
+
+static int
 fstring_cmp(VALUE a, VALUE b)
 {
     int cmp = rb_str_hash_cmp(a, b);
@@ -8718,8 +8728,6 @@ Init_String(void)
 #undef rb_intern
 #define rb_intern(str) rb_intern_const(str)

-    frozen_strings = st_init_table(&fstring_hash_type);
-
     rb_cString  = rb_define_class("String", rb_cObject);
     rb_include_module(rb_cString, rb_mComparable);
     rb_define_alloc_func(rb_cString, empty_str_alloc);
@@ -8891,4 +8899,6 @@ Init_String(void)
     rb_define_method(rb_cSymbol, "swapcase", sym_swapcase, 0);

     rb_define_method(rb_cSymbol, "encoding", sym_encoding, 0);
+
+    st_foreach(frozen_strings, fstring_set_class_i, rb_cString);
 }
```

#6 - 12/07/2013 07:37 PM - tmm1 (Aman Gupta)

Ah great! Thanks for finding my bug.

With your patch, long-lived strings on our rails app heap are reduced by 6000. I think we should merge it.

#7 - 12/08/2013 10:39 AM - tmm1 (Aman Gupta)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r44057.
Aman, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

---

parse.y: use rb_fstring() for strings stored in the symbol table

- parse.y (register_symid_str): use fstrings in symbol table [Bug #9171] [ruby-core:58656]
- parse.y (rb_id2str): ditto
- string.c (rb_fstring): create frozen_strings on first usage. this allows rb_fstring() calls from the parser (before cString is created)
- string.c (fstring_set_class_i): set klass on fstrings generated before cString was defined
- string.c (Init_String): convert frozen_strings table to String objects after boot
- ext/-test-/symbol/type.c (bug_sym_id2str): expose rb_id2str()
- test/-ext-/symbol/test_type.rb (module Test_Symbol): verify symbol table entries are fstrings

#8 - 12/04/2015 06:21 AM - nobu (Nobuyoshi Nakada)

- Tracker changed from Bug to Feature

- Description updated