

Ruby master - Bug #9934

High memory usage from file_expand_path_*

06/12/2014 01:57 AM - tmm1 (Aman Gupta)

Status:	Closed	
Priority:	Normal	
Assignee:		
Target version:	2.2.0	
ruby -v:	current	Backport: 2.0.0: DONE, 2.1: DONE
Description		
<p>All the file expansion routines use EXPAND_PATH_BUFFER() which allocates PATH_MAX bytes on the heap per invocation. The strings returned by File.expand_path are never realloc'd after they are populated, so they continue using 4kb (on linux) per string.</p> <p>In our rails app, 22MB of heap usage is due to expanded path name strings.</p>		
<pre>\$ ruby -robjectspace -e ' puts ObjectSpace.dump(File.expand_path("/foo")) ' {"address":"0x007fa2b44dd6c8", "type":"STRING", "class":"0x007fa2b3f99608", "bytesize":4, "capacity":4098, "value":"/foo", "encoding":"US-ASCII", "memsize":4099, "flags":{"wb_protected":true}}</pre>		
<p>The following failing patch demonstrates the issue as well:</p>		
<pre>diff --git a/test/ruby/test_file_exhaustive.rb b/test/ruby/test_file_exhaustive.rb index 2c945ea..49be9de 100644 --- a/test/ruby/test_file_exhaustive.rb +++ b/test/ruby/test_file_exhaustive.rb @@ -458,6 +458,12 @@ class TestFileExhaustive < Test::Unit::TestCase end end + def test_expand_path_memsize + require "objspace" + path = File.expand_path("/foo") + assert_equal 5, ObjectSpace.memsize_of(path) + end + def test_expand_path_encoding drive = (DRIVE ? 'C:' : '') if Encoding.find("filesystem") == Encoding::CP1251</pre>		

Associated revisions

Revision e2890123 - 06/12/2014 04:09 AM - nobu (Nobuyoshi Nakada)

file.c: shrink expanded path

- file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46410 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46410 - 06/12/2014 04:09 AM - nobu (Nobuyoshi Nakada)

file.c: shrink expanded path

- file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

Revision 46410 - 06/12/2014 04:09 AM - nobu (Nobuyoshi Nakada)

file.c: shrink expanded path

- file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

Revision 46410 - 06/12/2014 04:09 AM - nobu (Nobuyoshi Nakada)

file.c: shrink expanded path

- file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

Revision 46410 - 06/12/2014 04:09 AM - nobu (Nobuyoshi Nakada)

file.c: shrink expanded path

- file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

Revision 46410 - 06/12/2014 04:09 AM - nobu (Nobuyoshi Nakada)

file.c: shrink expanded path

- file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

Revision 46410 - 06/12/2014 04:09 AM - nobu (Nobuyoshi Nakada)

file.c: shrink expanded path

- file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

Revision 559a8275 - 06/13/2014 12:00 AM - tmm1 (Aman Gupta)

add more test coverage for [ruby-core:63136] [Bug #9934]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46414 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46414 - 06/13/2014 12:00 AM - tmm1 (Aman Gupta)

add more test coverage for [ruby-core:63136] [Bug #9934]

Revision 46414 - 06/13/2014 12:00 AM - tmm1 (Aman Gupta)

add more test coverage for [ruby-core:63136] [Bug #9934]

Revision 46414 - 06/13/2014 12:00 AM - tmm1 (Aman Gupta)

add more test coverage for [ruby-core:63136] [Bug #9934]

Revision 46414 - 06/13/2014 12:00 AM - tmm1 (Aman Gupta)

add more test coverage for [ruby-core:63136] [Bug #9934]

Revision 46414 - 06/13/2014 12:00 AM - tmm1 (Aman Gupta)

add more test coverage for [ruby-core:63136] [Bug #9934]

Revision 46414 - 06/13/2014 12:00 AM - tmm1 (Aman Gupta)

add more test coverage for [ruby-core:63136] [Bug #9934]

Revision 106f3e9d - 06/15/2014 01:09 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix assertion

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): wrong expected value, considering a prefix (drive letter or UNC) on DOSISH platforms. [Bug #9934]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46436 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46436 - 06/15/2014 01:09 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix assertion

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): wrong expected value, considering a prefix (drive letter or UNC) on DOSISH platforms. [Bug #9934]

Revision 46436 - 06/15/2014 01:09 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix assertion

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): wrong expected value, considering a prefix (drive letter or UNC) on DOSISH platforms. [Bug #9934]

Revision 46436 - 06/15/2014 01:09 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix assertion

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): wrong expected value, considering a prefix (drive letter or UNC) on DOSISH platforms. [Bug #9934]

Revision 46436 - 06/15/2014 01:09 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix assertion

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): wrong expected value, considering a prefix (drive letter or UNC) on DOSISH platforms. [Bug #9934]

Revision 46436 - 06/15/2014 01:09 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix assertion

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): wrong expected value, considering a prefix (drive letter or UNC) on DOSISH platforms. [Bug #9934]

Revision 46436 - 06/15/2014 01:09 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix assertion

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): wrong expected value, considering a prefix (drive letter or UNC) on DOSISH platforms. [Bug #9934]

Revision f3db73f1 - 06/15/2014 11:49 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix expected value

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): correct expected value, count terminator byte. [Bug #9934]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@46437 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 46437 - 06/15/2014 11:49 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix expected value

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): correct expected value, count terminator byte. [Bug #9934]

Revision 46437 - 06/15/2014 11:49 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix expected value

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): correct expected value, count terminator byte. [Bug #9934]

Revision 46437 - 06/15/2014 11:49 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix expected value

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): correct expected value, count terminator byte. [Bug #9934]

Revision 46437 - 06/15/2014 11:49 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix expected value

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): correct expected value, count terminator byte. [Bug #9934]

Revision 46437 - 06/15/2014 11:49 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix expected value

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): correct expected value, count terminator byte. [Bug #9934]

Revision 46437 - 06/15/2014 11:49 AM - nobu (Nobuyoshi Nakada)

test_file_exhaustive.rb: fix expected value

- test/ruby/test_file_exhaustive.rb (test_expand_path_memsizes): correct expected value, count terminator byte. [Bug #9934]

Revision 9355bcad - 08/18/2014 02:34 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) r46408,r46410,r46413,r46414,r46424,r46436,r46437: [Backport #9934]

```
string.c: shrink too big buffer
```

```
* string.c (rb_str_resize): shrink the buffer even if new length
```

is same but it is enough smaller than the capacity.

* file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

```
* string.c (rb_str_resize): should consider the capacity instead of the old length, as pointed out by nagachika.
```

```
* string.c (rb_str_resize): update capa only when buffer get reallocated.
```

```
http://d.hatena.ne.jp/nagachika/20140613/ruby_trunk_changes_46413_46420#r46413
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_1@47215 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 47215 - 08/18/2014 02:34 PM - nagachika (Tomoyuki Chikanaga)

merge revision(s) r46408,r46410,r46413,r46414,r46424,r46436,r46437: [Backport #9934]

string.c: shrink too big buffer

```
* string.c (rb_str_resize): shrink the buffer even if new length
```

is same but it is enough smaller than the capacity.

* file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

```
* string.c (rb_str_resize): should consider the capacity instead of the old length, as pointed out by nagachika.
```

```
* string.c (rb_str_resize): update capa only when buffer get reallocated.
```

```
http://d.hatena.ne.jp/nagachika/20140613/ruby_trunk_changes_46413_46420#r46413
```

Revision b9e63060 - 09/05/2014 04:23 AM - usa (Usaku NAKAMURA)

merge revision(s) 46408,46410,46413,46414,46424,46436,46437: [Backport #9934]

string.c: shrink too big buffer

```
* string.c (rb_str_resize): shrink the buffer even if new length
```

is same but it is enough smaller than the capacity.

* file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

```
* string.c (rb_str_resize): should consider the capacity instead of the old length, as pointed out by nagachika.
```

```
* string.c (rb_str_resize): update capa only when buffer get reallocated.
```

```
http://d.hatena.ne.jp/nagachika/20140613/ruby_trunk_changes_46413_46420#r46413
```

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby_2_0_0@47399 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 47399 - 09/05/2014 04:23 AM - usa (Usaku NAKAMURA)

merge revision(s) 46408,46410,46413,46414,46424,46436,46437: [Backport #9934]

string.c: shrink too big buffer

```
* string.c (rb_str_resize): shrink the buffer even if new length
```

is same but it is enough smaller than the capacity.

* file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

```
* string.c (rb_str_resize): should consider the capacity instead of the old length, as pointed out by nagachika.
```

```
* string.c (rb_str_resize): update capa only when buffer get reallocated.
```

```
http://d.hatena.ne.jp/nagachika/20140613/ruby_trunk_changes_46413_46420#r46413
```

History

#1 - 06/12/2014 02:34 AM - tmm1 (Aman Gupta)

This is the best I could come up with. Definitely not ideal since it allocates another ruby object.

```
diff --git a/file.c b/file.c
index 77facac..71e2d93 100644
--- a/file.c
+++ b/file.c
@@ -3391,6 +3391,9 @@ rb_file_expand_path_internal(VALUE fname, VALUE dname, int abs_mode, int long_na
     rb_str_set_len(result, p - buf);
     rb_enc_check(fname, result);
     ENC_CODERANGE_CLEAR(result);
+   if (OBJ_TAINTED(result)) tainted = 1;
+   result = rb_enc_str_new(RSTRING_PTR(result), p - buf, rb_enc_get(result));
+   if (tainted) OBJ_TAINT(result);
     return result;
 }
#endif /* _WIN32 */
diff --git a/test/ruby/test_file_exhaustive.rb b/test/ruby/test_file_exhaustive.rb
index 2c945ea..ed43ec0 100644
--- a/test/ruby/test_file_exhaustive.rb
+++ b/test/ruby/test_file_exhaustive.rb
@@ -458,6 +458,12 @@ class TestFileExhaustive < Test::Unit::TestCase
   end
 end

+ def test_expand_path_memsizes
+   require "objspace"
+   path = File.expand_path("/foo")
+   assert_operator ObjectSpace.memsizes_of(path), :<=, 5
+ end
+
+ def test_expand_path_encoding
+   drive = (DRIVE ? 'C:' : '')
+   if Encoding.find("filesystem") == Encoding::CP1251
```

#2 - 06/12/2014 03:40 AM - normalperson (Eric Wong)

ruby@tmm1.net wrote:

This is the best I could come up with. Definitely not ideal since it allocates another ruby object.

What about using `rb_str_resize`?

`rb_str_resize` may work even better if we remove the 1024 byte threshold and trust the `realloc` implementation to do the right thing.

Additionally, `rb_str_freeze` may also be smarter and force a resize unconditionally, as keeping the buffer is useless for frozen strings. Not sure if changing `rb_str_freeze` this way breaks compatibility (but we should be able to safely resize for `rb_fstring`, at least).

#3 - 06/12/2014 04:09 AM - nobu (Nobuyoshi Nakada)

- Status changed from Open to Closed

- % Done changed from 0 to 100

Applied in changeset r46410.

file.c: shrink expanded path

- file.c (expand_path): shrink expanded path which no longer needs rooms to append. [ruby-core:63114] [Bug #9934]

#4 - 06/12/2014 11:40 AM - dbussink (Dirkjan Bussink)

This does actually not fix the bug. This only works if a path expanded by `File.expand_path` is smaller than the string embedded length. If a string is longer than that length, it isn't made smaller and it still uses up too much memory.

#5 - 06/12/2014 02:51 PM - nobu (Nobuyoshi Nakada)

I think r46413 has fixed it.

#6 - 06/13/2014 11:31 AM - dbussink (Dirkjan Bussink)

Thanks for the fix!

#7 - 06/13/2014 06:45 PM - tmm1 (Aman Gupta)

- *Tracker changed from Bug to Backport*
- *Project changed from Ruby master to Backport21*
- *Status changed from Closed to Open*

#8 - 06/14/2014 03:04 AM - nobu (Nobuyoshi Nakada)

- *Tracker changed from Backport to Bug*
- *Project changed from Backport21 to Ruby master*
- *Description updated*
- *Category set to core*
- *Status changed from Open to Closed*
- *Target version set to 2.2.0*
- *ruby -v set to current*
- *Backport set to 2.0.0: REQUIRED, 2.1: REQUIRED*

Don't move projects, update Backport instead.

#9 - 08/18/2014 02:36 PM - nagachika (Tomoyuki Chikanaga)

- *Backport changed from 2.0.0: REQUIRED, 2.1: REQUIRED to 2.0.0: REQUIRED, 2.1: DONE*

r46408, r46410, r46413, r46414, r46424, r46436, r46437 and partially r44827 (to resolve conflict of r46408) were backported into ruby_2_1 branch at r47215.

#10 - 09/05/2014 04:25 AM - usa (Usaku NAKAMURA)

- *Backport changed from 2.0.0: REQUIRED, 2.1: DONE to 2.0.0: DONE, 2.1: DONE*

backported into ruby_2_0_0 at r47399.

note: ruby 2.0.0 doesn't recognize the length of the terminator of the string's encoding.