

Ruby - Bug #4314

spwn cause Segmantation Fault if take a long string.

01/24/2011 11:44 PM - kosaki (Motohiro KOSAKI)

Status:Closed Priority:Normal Assignee: Target version:1.9.2 ruby -v:ruby 1.9.3dev (2011-01-24 trunk 30642) [x86_64-linux]	
Backport:	
Description =begin ruby 1.9.3dev (2011-01-24 trunk 30642) [x86_64-linux] -e:1: [BUG] Segmentation fault ruby 1.9.3dev (2011-01-24 trunk 30642) [x86_64-linux] -- Control frame information ----- c:0004 p:--- s:0010 b:0010 l:000009 d:000009 CFUNC :spawn c:0003 p:0015 s:0006 b:0006 l:001f88 d:002578 EVAL -e:1 c:0002 p:--- s:0004 b:0004 l:000003 d:000003 FINISH c:0001 p:0000 s:0002 b:0002 l:001f88 d:001f88 TOP -- Ruby level backtrace information ----- -e:1:in <main>' -e:1:in spawn' -- C level backtrace information ----- ./ruby() [0x5205c5] vm_dump.c:797 ./ruby() [0x564796] error.c:249 ./ruby(rb_bug+0xb1) [0x564931] error.c:266 ./ruby() [0x4af190] signal.c:624 /lib64/libpthread.so.0() [0x382140f440] ./ruby(rb_proc_exec+0x130) [0x473920] process.c:1150 ./ruby() [0x475468] process.c:2375 ./ruby() [0x46fd25] process.c:2480 ./ruby(rb_protect+0x14d) [0x414a4d] eval.c:729 ./ruby(rb_fork_err+0x256) [0x476536] process.c:2573 ./ruby() [0x477399] process.c:2960 ./ruby() [0x50f757] vm_insnhelper.c:403 ./ruby() [0x5110d6] insns.def:1010 ./ruby() [0x515e8b] vm.c:1150 ./ruby(rb_iseq_eval_main+0x294) [0x5162c4] vm.c:1391 ./ruby() [0x414f32] eval.c:225 ./ruby(ruby_run_node+0x36) [0x416c66] eval.c:272 ./ruby() [0x414129] main.c:38 /lib64/libc.so.6(__libc_start_main+0xfd) [0x3820c1ec5d] ./ruby() [0x414019] -- Other runtime information ----- • Loaded script: -e • Loaded features: 0 enumerator.so 1 /usr/local/lib/ruby/1.9.1/x86_64-linux/enc/encdb.so 2 /usr/local/lib/ruby/1.9.1/x86_64-linux/enc/trans/transdb.so 3 /usr/local/lib/ruby/1.9.1/rubygems/defaults.rb 4 /usr/local/lib/ruby/1.9.1/thread.rb 5 /usr/local/lib/ruby/1.9.1/x86_64-linux/etc.so 6 /usr/local/lib/ruby/1.9.1/x86_64-linux/rbconfig.rb 7 /usr/local/lib/ruby/1.9.1/rubygems/exceptions.rb 8 /usr/local/lib/ruby/1.9.1/rubygems/version.rb	

9 /usr/local/lib/ruby/1.9.1/rubygems/requirement.rb
10 /usr/local/lib/ruby/1.9.1/rubygems/dependency.rb
11 /usr/local/lib/ruby/1.9.1/rubygems/gem_path_searcher.rb
12 /usr/local/lib/ruby/1.9.1/rubygems/user_interaction.rb
13 /usr/local/lib/ruby/1.9.1/rubygems/platform.rb
14 /usr/local/lib/ruby/1.9.1/rubygems/specification.rb
15 /usr/local/lib/ruby/1.9.1/rubygems/source_index.rb
16 /usr/local/lib/ruby/1.9.1/rubygems/builder.rb
17 /usr/local/lib/ruby/1.9.1/rubygems/config_file.rb
18 /usr/local/lib/ruby/1.9.1/rubygems/custom_require.rb
19 /usr/local/lib/ruby/1.9.1/rubygems.rb

• Process memory map:

00400000-00614000 r-xp 00000000 fd:03 2764210
00814000-00816000 rw-p 00214000 fd:03 2764210
00816000-0082e000 rw-p 00000000 00:00 0
00e12000-01176000 rw-p 00000000 00:00 0
3820800000-382081e000 r-xp 00000000 fd:00 4980746
3820a1e000-3820a1f000 r--p 0001e000 fd:00 4980746
3820a1f000-3820a20000 rw-p 0001f000 fd:00 4980746
3820a20000-3820a21000 rw-p 00000000 00:00 0
3820c00000-3820d75000 r-xp 00000000 fd:00 4980750
3820d75000-3820f75000 ---p 00175000 fd:00 4980750
3820f75000-3820f79000 r--p 00175000 fd:00 4980750
3820f79000-3820f7a000 rw-p 00179000 fd:00 4980750
3820f7a000-3820f7f000 rw-p 00000000 00:00 0
3821000000-3821002000 r-xp 00000000 fd:00 4981169
3821002000-3821202000 ---p 00002000 fd:00 4981169
3821202000-3821203000 r--p 00002000 fd:00 4981169
3821203000-3821204000 rw-p 00003000 fd:00 4981169
3821400000-3821417000 r-xp 00000000 fd:00 4980760
3821417000-3821616000 ---p 00017000 fd:00 4980760
3821616000-3821617000 r--p 00016000 fd:00 4980760
3821617000-3821618000 rw-p 00017000 fd:00 4980760
3821618000-382161c000 rw-p 00000000 00:00 0
3821800000-3821883000 r-xp 00000000 fd:00 4980786
3821883000-3821a82000 ---p 00083000 fd:00 4980786
3821a82000-3821a83000 r--p 00082000 fd:00 4980786
3821a83000-3821a84000 rw-p 00083000 fd:00 4980786
3821c00000-3821c07000 r-xp 00000000 fd:00 4980767
3821c07000-3821e06000 ---p 00007000 fd:00 4980767
3821e06000-3821e07000 r--p 00006000 fd:00 4980767
3821e07000-3821e08000 rw-p 00007000 fd:00 4980767
3825800000-3825816000 r-xp 00000000 fd:00 4980769
3825816000-3825a15000 ---p 00016000 fd:00 4980769
3825a15000-3825a16000 rw-p 00015000 fd:00 4980769
3831000000-3831058000 r-xp 00000000 fd:00 4981182
3831058000-3831257000 ---p 00058000 fd:00 4981182
3831257000-3831259000 rw-p 00057000 fd:00 4981182
3831259000-383125d000 rw-p 00000000 00:00 0
3831800000-3831807000 r-xp 00000000 fd:00 4981183
3831807000-3831a07000 ---p 00007000 fd:00 4981183
3831a07000-3831a08000 r--p 00007000 fd:00 4981183
3831a08000-3831a09000 rw-p 00008000 fd:00 4981183
3831a09000-3831a37000 rw-p 00000000 00:00 0
7fac4cf5e000-7fac52ebd000 rw-p 00000000 00:00 0
7fac52ebd000-7fac52ec0000 r-xp 00000000 fd:00 920209
7fac52ec0000-7fac530bf000 ---p 00003000 fd:00 920209
7fac530bf000-7fac530c0000 rw-p 00002000 fd:00 920209
7fac530c0000-7fac530c2000 r-xp 00000000 fd:00 930967
7fac530c2000-7fac532c1000 ---p 00002000 fd:00 930967
7fac532c1000-7fac532c2000 rw-p 00001000 fd:00 930967
7fac532c2000-7fac532c4000 r-xp 00000000 fd:00 930968
7fac532c4000-7fac534c3000 ---p 00002000 fd:00 930968
7fac534c3000-7fac534c4000 rw-p 00001000 fd:00 930968

/home/kosaki/linux/ruby-svn/ruby/ruby
/home/kosaki/linux/ruby-svn/ruby/ruby

[heap]

/lib64/ld-2.12.1.so
/lib64/ld-2.12.1.so
/lib64/ld-2.12.1.so

/lib64/libc-2.12.1.so
/lib64/libc-2.12.1.so
/lib64/libc-2.12.1.so
/lib64/libc-2.12.1.so

/lib64/libdl-2.12.1.so
/lib64/libdl-2.12.1.so
/lib64/libdl-2.12.1.so
/lib64/libdl-2.12.1.so
/lib64/libpthread-2.12.1.so
/lib64/libpthread-2.12.1.so
/lib64/libpthread-2.12.1.so
/lib64/libpthread-2.12.1.so
/lib64/libpthread-2.12.1.so

/lib64/libm-2.12.1.so
/lib64/libm-2.12.1.so
/lib64/libm-2.12.1.so
/lib64/libm-2.12.1.so
/lib64/librt-2.12.1.so
/lib64/librt-2.12.1.so
/lib64/librt-2.12.1.so
/lib64/librt-2.12.1.so
/lib64/libgcc_s-4.4.4-20100503.so.1
/lib64/libgcc_s-4.4.4-20100503.so.1
/lib64/libgcc_s-4.4.4-20100503.so.1
/lib64/libfreebl3.so
/lib64/libfreebl3.so
/lib64/libfreebl3.so

/lib64/libcrypt-2.12.1.so
/lib64/libcrypt-2.12.1.so
/lib64/libcrypt-2.12.1.so
/lib64/libcrypt-2.12.1.so

/usr/local/lib/ruby/1.9.1/x86_64-linux/etc.so
/usr/local/lib/ruby/1.9.1/x86_64-linux/etc.so
/usr/local/lib/ruby/1.9.1/x86_64-linux/etc.so
/usr/local/lib/ruby/1.9.1/x86_64-linux/enc/trans/transdb.so
/usr/local/lib/ruby/1.9.1/x86_64-linux/enc/trans/transdb.so
/usr/local/lib/ruby/1.9.1/x86_64-linux/enc/trans/transdb.so
/usr/local/lib/ruby/1.9.1/x86_64-linux/enc/encdb.so
/usr/local/lib/ruby/1.9.1/x86_64-linux/enc/encdb.so
/usr/local/lib/ruby/1.9.1/x86_64-linux/enc/encdb.so

```

7fac534c4000-7fac534c5000 rw-p 00000000 00:00 0
7fac534c5000-7fac534c6000 ---p 00000000 00:00 0
7fac534c6000-7fac535ca000 rw-p 00000000 00:00 0
7fac535ca000-7fac5945b000 r--p 00000000 fd:00 321758 /usr/lib/locale/locale-archive
7fac5945b000-7fac59460000 rw-p 00000000 00:00 0
7fac5947d000-7fac5947f000 rw-p 00000000 00:00 0
7ffa39e7000-7ffa39fc000 rw-p 00000000 00:00 0 [stack]
7ffa39ff000-7ffa3a00000 r-xp 00000000 00:00 0 [vdso]
ffffffff600000-ffffffff601000 r-xp 00000000 00:00 0 [vsyscall]

```

[NOTE]

You may have encountered a bug in the Ruby interpreter or extension libraries.

Bug reports are welcome.

For details: <http://www.ruby-lang.org/bugreport.html>

=end

History

#1 - 01/25/2011 07:44 AM - kosaki (Motohiro KOSAKI)

```
=begin
reproduce way.
```

```
./ruby -ve 'spawn "a"*100_000_000'
```

```
=end
```

#2 - 01/25/2011 08:44 AM - nobu (Nobuyoshi Nakada)

```
=begin
Hi,
```

At Tue, 25 Jan 2011 07:44:51 +0900,

Motohiro KOSAKI wrote in [\[ruby-core:34842\]](#):

```
reproduce way.
```

```
./ruby -ve 'spawn "a"*100_000_000'
```

I propose new APIs for exception-safe temporary buffer.

```
diff --git i/include/ruby/ruby.h w/include/ruby/ruby.h
```

```
index 4e15517..aca517c 100644
```

```
--- i/include/ruby/ruby.h
```

```
+++ w/include/ruby/ruby.h
```

```
@@ -1028,6 +1028,18 @@ NUM2CHR(VALUE x)
```

```
#define ALLOCA_N(type,n) (type*)alloca(sizeof(type)*(n))
```

```
+void *rb_alloc_tmp_buffer(volatile VALUE *store, long len);
```

```
+void rb_free_tmp_buffer(volatile VALUE store);
```

```
+/* allocates n bytes temporary buffer and stores VALUE including it
```

```
o in v. n may be evaluated twice. /
```

```
+#ifdef C_ALLOCA
```

```
+# define ALLOCV(v, n) rb_alloc_tmp_buffer(&(v), (n))
```

```
+#else
```

```
+# define ALLOCV(v, n) ((n < 1024 ? (RB_GC_GUARD(v) = 0, alloca(n)) : rb_alloc_tmp_buffer(&(v), (n)))
```

```
+#endif
```

```
+#define ALLOCA_N(type, v, n) (type)ALLOCV(&(v), sizeof(type)*(n))
```

```
+#define ALLOCV_END(v) rb_free_tmp_buffer(&(v))
```

```
#define MEMZERO(p,type,n) memset((p), 0, sizeof(type)*(n))
```

```
#define MEMCPY(p1,p2,type,n) memcpy((p1), (p2), sizeof(type)*(n))
```

```
#define MEMMOVE(p1,p2,type,n) memmove((p1), (p2), sizeof(type)*(n))
```

```
diff --git i/process.c w/process.c
```

```
index 2a6a321..2d0dc44 100644
```

```
--- i/process.c
```

```
+++ w/process.c
```

```
@@ -1027,16 +1027,23 @@ exec_with_sh(const char *prog, char argv)
```

```
--argv = (char)"sh";
```

```
execv("/bin/sh", argv);
```

```

}
#define ALLOCA_ARGV(n) ALLOCA_N(char, (n)+1)
#define ARGV_COUNT(n) ((n)+1)
#else
#define try_with_sh(prog, argv) (void)0
#define ALLOCA_ARGV(n) ALLOCA_N(char, (n))
#define ARGV_COUNT(n) (n)
#endif
#define ARGV_SIZE(n) (sizeof(char) * ARGV_COUNT(n))
#define ALLOC_ARGV(n, v) ALLOCV_N(char, (v), ARGV_COUNT(n))
#define ALLOC_ARGV_WITH_STR(n, v, s, l) \

    • (char **)((s) = ALLOCV((v), ARGV_SIZE(n) + (l)) + ARGV_SIZE(n)) - ARGV_SIZE(n))

static int
proc_exec_v(char **argv, const char *prog)
{
    char fbuf[MAXPATHLEN];
    #if defined(EMX) || defined(OS2)

        • char **new_argv = NULL;
          #endif

        if (!prog)
            prog = argv[0];
        @@ -1052,16 +1059,15 @@ proc_exec_v(char **argv, const char *prog)
            char *extension;

if ((extension = strrchr(prog, '.')) != NULL && STRCASECMP(extension, ".bat") == 0) {

    • [REDACTED]
      char *p;
      int n;

      for (n = 0; argv[n]; n++)
          /* no-op */;

    • [REDACTED]

    • [REDACTED]
      for (; n > 0; n--)
          new_argv[n + 1] = argv[n];

    • [REDACTED]

    • [REDACTED]
      for (p = new_argv[1]; *p != '\0'; p++)
          if (*p == '/')
              *p = '\\';

@@ -1078,6 +1084,12 @@ proc_exec_v(char **argv, const char *prog)
before_exec();
execv(prog, argv);
preserving_errno(try_with_sh(prog, argv); after_exec());
#if defined(EMX) || defined(OS2)

    • if (new_argv) {
    • xfree(new_argv[0]);
    • xfree(new_argv);
    • }
    #endif
    return -1;
}

@@ -1086,15 +1098,18 @@ rb_proc_exec_n(int argc, VALUE *argv, const char *prog)
{
    char **args;
    int i;

    • int ret = -1;

```

- VALUE v;
- args = ALLOCA_ARGV(argc+1);
- args = ALLOC_ARGV(argc+1, v);
 for (i=0; i<argc; i++) {
 args[i] = RSTRING_PTR(argv[i]);
 }
 args[i] = 0;
 if (args[0]) {
- return proc_exec_v(args, prog);
- ret = proc_exec_v(args, prog);
 }
- ALLOCV_END(v);
 return -1;
 }

@@ -1105,6 +1120,8 @@ rb_proc_exec(const char *str)

const char *s = str;

char *ss, *t;

char **argv, **a;

- VALUE v;
- int ret = -1;
 #endif
- while (*str && ISSPACE(*str))
 @@ -1114,6 +1131,7 @@ rb_proc_exec(const char *str)
 before_exec();
 rb_w32_spawn(P_OVERLAY, (char *)str, 0);
 after_exec();
- return -1;
 #else
 for (s=str; *s; s++) {
 if (ISSPACE(*s)) {
 @@ -1145,8 +1163,7 @@ rb_proc_exec(const char *str)
 return -1;
 }
 }
- a = argv = ALLOCA_ARGV((s-str)/2+2);
- ss = ALLOCA_N(char, s-str+1);
- a = argv = ALLOC_ARGV_WITH_STR((s-str)/2+2, v, ss, s-str+1);
 memcpy(ss, str, s-str);
 ss[s-str] = '\0';
 if ((*a++ = strtok(ss, " \t")) != 0) {
 @@ -1156,11 +1173,14 @@ rb_proc_exec(const char *str)
 *a = NULL;
 }
 if (argv[0]) {
- return proc_exec_v(argv, 0);
- ret = proc_exec_v(argv, 0);
 • }
 • else {
 • errno = ENOENT;
 • }
- errno = ENOENT;
- ALLOCV_END(v);
- return ret;
 #endif /* _WIN32 */
- return -1;
 }

#if defined(_WIN32)

```
@@ -1202,15 +1222,18 @@ proc_spawn_n(int argc, VALUE *argv, VALUE prog)
```

```
{
char **args;
int i;

• VALUE v;
• rb_pid_t pid = -1;

• args = ALLOCA_ARGV(argc + 1);

• args = ALLOC_ARGV(argc + 1, v);
  for (i = 0; i < argc; i++) {
    args[i] = RSTRING_PTR(argv[i]);
  }
  args[i] = (char*) 0;
  if (args[0])

• return proc_spawn_v(args, prog ? RSTRING_PTR(prog) : 0);
• return -1;

• pid = proc_spawn_v(args, prog ? RSTRING_PTR(prog) : 0);
• ALLOCV_END(v);
• return pid;
}
```

```
#if defined(_WIN32)
```

```
@@ -1223,6 +1246,7 @@ proc_spawn(char *str)
```

```
char *s, *t;
char **argv, **a;
rb_pid_t status;

• VALUE v;

  for (s = str; *s; s++) {
    if (*s != ' ' && !ISALPHA(s) && strchr("[]<>()~&\\$;\"'\"\\n", *s)) {
      @@ -1234,15 +1258,16 @@ proc_spawn(char *str)
      return status;
    }
  }

• a = argv = ALLOCA_ARGV((s - str) / 2 + 2);
• s = ALLOCA_N(char, s - str + 1);

• a = argv = ALLOC_ARGV_WITH_STR((s - str) / 2 + 2, v, s, s - str + 1);
  strcpy(s, str);
  if (*a++ = strtok(s, " \\t")) {
    while (t = strtok(NULL, " \\t"))
      *a++ = t;
    *a = NULL;
  }

• return argv[0] ? proc_spawn_v(argv, 0) : -1;

• status = argv[0] ? proc_spawn_v(argv, 0) : -1;
• ALLOCV_END(v);
• return status;
}
#endif
#endif
diff --git i/string.c w/string.c
index e5e24ac..c3468b9 100644
--- i/string.c
+++ w/string.c
@@ -45,6 +45,8 @@
#undef rb_str_buf_cat2
#undef rb_str_cat2
```

```
+static VALUE rb_str_clear(VALUE str);
```

```
+
VALUE rb_cString;
VALUE rb_cSymbol;
```

```
@@ -765,6 +767,22 @@ rb_str_tmp_new(long len)
```

```
return str_new(0, 0, len);
```

```

}

+void *
+rb_alloc_tmp_buffer(volatile VALUE *store, long len)
+{
    • VALUE s = rb_str_tmp_new(len);
    • *store = s;
    • return RSTRING_PTR(s);
    +}

```

```

+void
+rb_free_tmp_buffer(volatile VALUE *store)
+{
    • VALUE s = *store;
    • *store = 0;
    • if (s) rb_str_clear(s);
    +}

```

```

void
rb_str_free(VALUE str)
{

```

```

--
Nobu Nakada

```

```

=end

```

#3 - 01/27/2011 05:33 PM - kosaki (Motohiro KOSAKI)

- *Status changed from Open to Closed*

```

=begin
r30661,r30662 and r30663 fixed this issue.

```

```

=end

```