

Ruby - Bug #7141

ALT_STACK_SIZE is not enough

10/11/2012 09:56 PM - authorNari (Narihiro Nakamura)

Status:	Closed	Backport:
Priority:	Normal	
Assignee:	kosaki (Motohiro KOSAKI)	
Target version:	2.0.0	
ruby -v:	ruby 2.0.0dev (2012-10-05 trunk 36982) [x86_64-linux]	

Description

nari

<http://bugs.ruby-lang.org/issues/7095#note-6>

r37088

altstackmprotectLinux64bit

SEGV

--- ---

diff --git a/gc.c b/gc.c

index f1f7aaa..dbe3c3d 100644

--- a/gc.c

+++ b/gc.c

@@ -588,6 +588,8 @@ add_heap_slots(rb_objspace_t *objspace, size_t add)

heaps_inc = 0;

}

+#include <sys/mman.h>

+

static void

init_heap(rb_objspace_t *objspace)

{

@@ -599,7 +601,17 @@ init_heap(rb_objspace_t objspace)

/altstack of another threads are allocated in another place */

rb_thread_t *th = GET_THREAD();

void *tmp = th->altstack;

-
-
-
-
-
-
-
-

- [REDACTED]
- [REDACTED]
- [REDACTED]
- [REDACTED]

```
free(tmp); /* free previously allocated area */
```

```
}
#endif
diff --git a/vm.c b/vm.c
index ae201dc..95d8202 100644
--- a/vm.c
+++ b/vm.c
@@ -1736,7 +1736,7 @@ thread_free(void *ptr)
else {
#ifdef USE_SIGALTSTACK
if (th->altstack) {
```

- [REDACTED]

- [REDACTED]

```
#endif
ruby_xfree(ptr);
--- [REDACTED] ---
```

```
[REDACTED] ./miniruby [REDACTED]
```

```
% ./miniruby -e 'Process.kill :SEGV, $$'
altstack: 0x7fd3f23a7000-0x7fd3f23a7190...0x7fd3f23a8190-0x7fd3f23a8320
-e:1: [BUG] Segmentation fault
ruby 2.0.0dev (2012-10-05 trunk 36982) [x86_64-linux]
```

```
-- Control frame information -----
zsh: segmentation fault (core dumped) ./miniruby -e 'Process.kill :SEGV, $$'
```

```
% ./miniruby -v
altstack: 0x7f7ee72fb000-0x7f7ee72fb190...0x7f7ee72fc190-0x7f7ee72fc320
ruby 2.0.0dev (2012-10-05 trunk 36982) [x86_64-linux]
```

```
[REDACTED] gdb [REDACTED]
```

```
% gdb ./miniruby
GNU gdb (Ubuntu/Linaro 7.4-2012.04-0ubuntu2) 7.4-2012.04
```

```
...
(gdb) r -e 'Process.kill :SEGV, $$'
```

```
...
Continuing.
-e:1: [BUG] Segmentation fault
ruby 2.0.0dev (2012-10-05 trunk 36982) [x86_64-linux]
```

```
-- Control frame information -----
```

```
Program received signal SIGSEGV, Segmentation fault.
buffered_vfprintf (s=0x7ffff747c180, format=0x5555557358af "c:%04td ", args=0x7ffff7ff76a8) at vfprintf.c:2313
2313 vfprintf.c: No such file or directory.
(gdb) bt
#0 buffered_vfprintf (s=0x7ffff747c180, format=0x5555557358af "c:%04td ", args=0x7ffff7ff76a8) at vfprintf.c:2313
#1 0x00007ffff710bbfe in _IO_vfprintf_internal (s=0x7ffff747c180, format=0x5555557358af "c:%04td ", ap=0x7ffff7ff76a8) at vfprintf.c:1316
#2 0x00007ffff7116857 in __fprintf (stream=, format=) at fprintf.c:33
#3 0x000055555556fad8a in control_frame_dump (th=0x5555559de530, cfp=0x7ffff7fcff08) at vm_dump.c:111
```

```
#4 0x00005555556fafbf in rb_vmdebug_stack_dump_raw (th=0x5555559de530, cfp=0x7fff7fcff08) at vm_dump.c:163
#5 0x00005555556fb626 in rb_vm_bugreport () at vm_dump.c:610
#6 0x00005555555b1599 in report_bug (file=0x555555a4dad8 "-e", line=1, fmt=0x55555572ebaf "Segmentation fault",
args=0x7fff7ff7b58) at error.c:306
#7 0x00005555555b16b6 in rb_bug (fmt=0x55555572ebaf "Segmentation fault") at error.c:325
#8 0x0000555555568085e in sigsegv (sig=11, info=0x7fff7ff7db0, ctx=0x7fff7ff7c80) at signal.c:607
#9
....
(gdb) i f 0
Stack frame at 0x7fff7ff7030:
...
Locals at 0x7fff7ff4ed8, Previous frame's sp is 0x7fff7ff7030
...
(gdb) i f 8
Stack frame at 0x7fff7ff7c80:
...
(gdb) p 0x7fff7ff4ed8 - 0x7fff7ff7c80
$1 = -11688

XXXXXXXXXXLocalsXXXXXXXXXX0x7fff7ff7c80XXXX
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX...X

XXXXXXXXALT_STACK_SIZEX5XXXXXXXXXXXXXXXXXXXXXXXXXXXXX

diff --git a/vm_core.h b/vm_core.h
index 8d51407..13f12d4 100644
--- a/vm_core.h
+++ b/vm_core.h
@@ -416,9 +416,9 @@ struct rb_unblock_callback {
struct rb_mutex_struct;

#ifdef MINSIGSTKSZ
-#define ALT_STACK_SIZE (MINSIGSTKSZ2)
+#define ALT_STACK_SIZE (MINSIGSTKSZ10)
#else
-#define ALT_STACK_SIZE (41024)
+#define ALT_STACK_SIZE (201024)
#endif

typedef struct rb_thread_struct {

XXXXXXXXXXXXXXXXXXXX
http://bugs.ruby-lang.org/issues/5139
```

Associated revisions

- `signal.c (rb_sigaltstack_size)`: new. calculate stack size for `sigsegv` handler. enlarge value when `x86` or `x86_64` on Linux. Linux has very small `MINSIGSTKSZ` size (2048 bytes) and our `sigsegv` routine need 5KiB at least. [Bug #7141]
- `internal.h`: add declaration of `rb_sigaltstack_size()`.
- `vm_core.h`: remove `ALT_STACK_SIZE` definition.
- `signal.c (rb_register_sigaltstack)`: replace `ALT_STACK_SIZE` with `rb_sigaltstack_size()`;

- git-svn-id: [svn+ssh://ci.ruby-lang.org/ruby/trunk@38409](https://ci.ruby-lang.org/ruby/trunk@38409) b2dd03c8-39d4-4d8f-98ff-823fe69b080e

- `signal.c (rb_sigaltstack_size)`: new. calculate stack size for `sigsegv` handler. enlarge value when `x86` or `x86_64` on Linux. Linux has very small `MINSIGSTKSZ` size (2048 bytes) and our `sigsegv` routine need 5KiB at least. [Bug #7141]
- `internal.h`: add declaration of `rb_sigaltstack_size()`.
- `vm_core.h`: remove `ALT_STACK_SIZE` definition.
- `signal.c (rb_register_sigaltstack)`: replace `ALT_STACK_SIZE` with `rb_sigaltstack_size()`;
- `gc.c (Init_heap)`: ditto.
- `vm.c (th_init)`: ditto.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@38409 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#1 - 10/12/2012 07:28 AM - kosaki (Motohiro KOSAKI)

[illegible]

□□□□□□□□□□□□□□□□

```
setarch x86_64 -R ./miniruby -e 'Process.kill :SEGV, $$'
```

altstack: 0x7ffff7d83000-0x7ffff7d84000...0x7ffff7d85000-0x7ffff7d86000

\$RSP	stack_usage	stack_usage_total
-------	-------------	-------------------

```
#0 BSD_vfprintf 0x7ffff7d83e60 1520
#2 control_frame_dump 0x7ffff7d84450 320 1840
#3 rb_vmdebug_stack_dump_raw 0x7ffff7d84590 288 2128
#4 rb_vm_bugreport 0x7ffff7d846b0 328 2456
#5 report_bug 0x7ffff7d84820 336 2792
#6 rb_bug 0x7ffff7d84970 272 3064
#7 sigseqv 0x7ffff7d84a80
```

```

0000 BSD_vfprintf char buff[1024] 0000000000000000float00000000
68 OK00000000float00000000alloca000000000000
2.000000000000 ALT_STACK_SIZE 0000000000000000409600081920
0000000000

```

```

#####fprintf#####
snprintf + write#####
#####ruby#####snprintf#####thread safe#####async signal safe#####

```

- Status changed from Assigned to Closed
- % Done changed from 0 to 100

This issue was solved with changeset r38409.

Narihiro, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

- `signal.c (rb_sigaltstack_size)`: new. calculate stack size for `sigsegv` handler. enlarge value when `x86` or `x86_64` on Linux. Linux has very small `MINSIGSTKSZ` size (2048 bytes) and our `sigsegv` routine need 5KiB at least. [Bug [#7141](#)]
- `internal.h`: add declaration of `rb_sigaltstack_size()`.
- `vm_core.h`: remove `ALT_STACK_SIZE` definition.
- `signal.c (rb_register_sigaltstack)`: replace `ALT_STACK_SIZE` with `rb_sigaltstack_size()`;
- `gc.c (Init_heap)`: ditto.
- `vm.c (th_init)`: ditto.

#4 - 12/15/2012 11:38 PM - kosaki (Motohiro KOSAKI)

[illegible]

altstack: 0x7fff7d92000-0x7fff7d93000...0x7fff7d94000-0x7fff7d95000

```
$rsp                                stack_usage
```

```
#0 _IO_vfprintf_internal 0x7ffffd92f80 1728
#1 __fprintf_chk 0x7ffffd93640 240
#2 fprintf 0x7ffffd93730 0
#3 rb_vmdebug_stack_dump_raw 0x7ffffd93730
#4 rb_vm_bugreport
#5 report_bug
#6 rb_bug
#7 sigsegv
#8
```