

## Ruby trunk - Feature #4568

### [PATCH] file.c (rb\_group\_member): kill 256K of stack usage

04/11/2011 04:01 PM - normalperson (Eric Wong)

|  |                          |
|--|--------------------------|
| <b>Status:</b>   | Closed                   |
| <b>Priority:</b>   | Normal                   |
| <b>Assignee:</b>   | kosaki (Motohiro KOSAKI) |
| <b>Target version:</b>   | 2.0.0                    |
| <b>Description</b>   |                          |
| =begin<br>It was using 256K stack on my x86_64 machine.<br><br>Found with scripts/checkstack.pl in the Linux kernel source:<br><br>objdump -D ./ruby   ~/linux-2.6/scripts/checkstack.pl x86_64<br><br>Also pushed to my repo: git pull git://bogomips.org/ruby stack-reduce<br><br>=end |                          |

### History

#### #1 - 04/11/2011 05:23 PM - normalperson (Eric Wong)

=begin  
Lowering RUBY\_STACK\_MIN\_LIMIT to 64KB across the board in  
thread\_pthread.c seems to work fine for check, test-rubyspec,  
benchmark-each.

No real code, though, and I also don't know what outside C extensions  
do, but 64KB is the PTHREAD\_STACK\_MIN for my platform (and I've always  
felt it was too high).

```
diff --git a/thread_pthread.c b/thread_pthread.c
index ad6f716..a015873 100644
--- a/thread_pthread.c
+++ b/thread_pthread.c
@@ -630,11 +630,7 @@ use_cached_thread(rb_thread_t *th)
 }

enum {
-#ifdef SYMBIAN32
-  RUBY_STACK_MIN_LIMIT = 64 * 1024, /* 64KB: Let's be slightly more frugal on mobile platform /
-#else
-  RUBY_STACK_MIN_LIMIT = 512 * 1024, /512KB /
-#endif
+  RUBY_STACK_MIN_LIMIT = 64 * 1024, /64KB */
  RUBY_STACK_SPACE_LIMIT = 1024 * 1024
};
```

--  
Eric Wong  
=end

**#2 - 04/11/2011 09:19 PM - kosaki (Motohiro KOSAKI)**

- Status changed from Open to Closed
- Assignee set to kosaki (Motohiro KOSAKI)

```
=begin  
Committed by r31259.  
=end
```

**#3 - 04/12/2011 08:18 PM - normalperson (Eric Wong)**

```
=begin  
Lowering RUBY_STACK_MIN_LIMIT to 64KB across the board in  
thread_pthread.c seems to work fine for check, test-rubyspec,  
benchmark-each.
```

No real code, though, and I also don't know what outside C extensions do, but 64KB is the PTHREAD\_STACK\_MIN for my platform (and I've always felt it was too high).

```
diff --git a/thread_pthread.c b/thread_pthread.c  
index ad6f716..a015873 100644  
--- a/thread_pthread.c  
+++ b/thread_pthread.c  
@@ -630,11 +630,7 @@ use_cached_thread(rb_thread_t *th  
}  
  
enum {  
-#ifdef SYMBIAN32  
- RUBY_STACK_MIN_LIMIT = 64 * 1024, /* 64KB: Let's be slightly more frugal on mobile platform /  
-#else  
- RUBY_STACK_MIN_LIMIT = 512 * 1024, /512KB /  
-#endif  
+ RUBY_STACK_MIN_LIMIT = 64 * 1024, /64KB */  
RUBY_STACK_SPACE_LIMIT = 1024 * 1024  
};  
  
--  
Eric Wong  
=end
```

**#4 - 04/13/2011 08:45 AM - normalperson (Eric Wong)**

=begin

Eric Wong [normalperson@yhbt.net](mailto:normalperson@yhbt.net) wrote:

Lowering RUBY\_STACK\_MIN\_LIMIT to 64KB across the board in thread\_pthread.c seems to work fine for check, test-rubyspec, benchmark-each.

No real code, though, and I also don't know what outside C extensions do, but 64KB is the PTHREAD\_STACK\_MIN for my platform (and I've always felt it was too high).

I was wrong about 64KB on my system 16KB is the minimum with NPTL :x

The lowest successful stack size I've been able to run is 48K, I get stack corruption and GC failures with 44K and lower.

I've also run my Rainbows! web server[1] integration/torture test suite with several threaded options and everything passed with 48K and didn't notice ill effects. 44K seemed fine, too, I think, but 32K failed Rainbows! tests miserably.

To be on the safe side with existing code/extensions and maybe some overaggressive alloca() calls, I think 64KB is reasonable.

[1] - <http://rainbows.rubyforge.org/> + <http://bogomips.org/rainbows.git>

--

Eric Wong

=end

## Files

---

|  |         |            |                          |
|--|---------|------------|--------------------------|
| 0001-file.c-rb_group_member-kill-256K-of-stack-usage.patch | 1.21 KB | 04/11/2011 | normalperson (Eric Wong) |
|--|---------|------------|--------------------------|