

## Ruby master - Feature #8015

### [patch] tuneable HEAP\_GROWTH\_FACTOR

03/05/2013 10:17 AM - tmm1 (Aman Gupta)

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	authorNari (Narihiro Nakamura)
<b>Target version:</b>	2.1.0
<b>Description</b>	
<pre>diff --git a/gc.c b/gc.c index 925e496..71f509f 100644 --- a/gc.c +++ b/gc.c @@ -71,11 +71,13 @@ #endif #define HEAP_MIN_SLOTS 10000 #define FREE_MIN 4096 +#define HEAP_GROWTH_FACTOR 1.8  typedef struct { unsigned int initial_malloc_limit; unsigned int initial_heap_min_slots; unsigned int initial_free_min;  <ul style="list-style-type: none"> <li>• double initial_growth_factor; #if defined(ENABLE_VM_OBJSPACE) &amp;&amp; ENABLE_VM_OBJSPACE int gc_stress; #endif @@ -85,6 +87,7 @@ static ruby_gc_params_t initial_params = { GC_MALLOC_LIMIT, HEAP_MIN_SLOTS, FREE_MIN,</li> <li>• HEAP_GROWTH_FACTOR, #if defined(ENABLE_VM_OBJSPACE) &amp;&amp; ENABLE_VM_OBJSPACE FALSE, #endif @@ -287,6 +290,7 @@ int *ruby_initial_gc_stress_ptr = &amp;rb_objspace.gc_stress; #define initial_malloc_limit initial_params.initial_malloc_limit #define initial_heap_min_slots initial_params.initial_heap_min_slots #define initial_free_min initial_params.initial_free_min +#define initial_growth_factor initial_params.initial_growth_factor</li> </ul> <pre>#define is_lazy_sweeping(objspace) ((objspace)-&gt;heap.sweep_slots != 0)  @@ -605,7 +609,7 @@ initial_expand_heap(rb_objspace_t *objspace) static void set_heaps_increment(rb_objspace_t *objspace) { <ul style="list-style-type: none"> <li>• size_t next_heaps_length = (size_t)(heaps_used * 1.8);</li> <li>• size_t next_heaps_length = (size_t)(heaps_used * initial_growth_factor);</li> </ul> <pre>if (next_heaps_length == heaps_used) { next_heaps_length++; @@ -3333,7 +3337,7 @@ rb_gc_disable(void) void rb_gc_set_params(void) { <ul style="list-style-type: none"> <li>• char *malloc_limit_ptr, *heap_min_slots_ptr, *free_min_ptr;</li> <li>• char *malloc_limit_ptr, *heap_min_slots_ptr, *free_min_ptr, *growth_factor_ptr;</li> </ul> <pre>if (rb_safe_level() &gt; 0) return;  @@ -3360,6 +3364,16 @@ rb_gc_set_params(void) } }  <ul style="list-style-type: none"> <li>• growth_factor_ptr = getenv("RUBY_HEAP_SLOTS_GROWTH_FACTOR");</li> <li>• if (growth_factor_ptr != NULL) {</li> </ul> </pre></pre></pre></pre>	

- double growth\_factor\_f = atof(growth\_factor\_ptr);
- if (RTEST(ruby\_verbose))
- fprintf(stderr, "growth\_factor=%f (%f)\n", growth\_factor\_f, initial\_growth\_factor);
- if (growth\_factor\_f > 0) {
- initial\_growth\_factor = growth\_factor\_f;
- }
- } + free\_min\_ptr = getenv("RUBY\_FREE\_MIN"); if (free\_min\_ptr != NULL) { int free\_min\_i = atoi(free\_min\_ptr);

#### Related issues:

Related to Ruby master - Feature #2837: Compile-time constant for HEAP\_GROWTH...

Closed

03/05/2010

#### Associated revisions

##### Revision 2f7ca893 - 03/13/2013 02:52 PM - nari

- gc.c: allow to tune growth of heap by environment variable RUBY\_HEAP\_SLOTS\_GROWTH\_FACTOR. patched by tmm1(Aman Gupta). [Feature #8015] [ruby-core:53131]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@39746 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 39746 - 03/13/2013 02:52 PM - nari

- gc.c: allow to tune growth of heap by environment variable RUBY\_HEAP\_SLOTS\_GROWTH\_FACTOR. patched by tmm1(Aman Gupta). [Feature #8015] [ruby-core:53131]

##### Revision 39746 - 03/13/2013 02:52 PM - nari

- gc.c: allow to tune growth of heap by environment variable RUBY\_HEAP\_SLOTS\_GROWTH\_FACTOR. patched by tmm1(Aman Gupta). [Feature #8015] [ruby-core:53131]

##### Revision 39746 - 03/13/2013 02:52 PM - nari

- gc.c: allow to tune growth of heap by environment variable RUBY\_HEAP\_SLOTS\_GROWTH\_FACTOR. patched by tmm1(Aman Gupta). [Feature #8015] [ruby-core:53131]

##### Revision 39746 - 03/13/2013 02:52 PM - nari

- gc.c: allow to tune growth of heap by environment variable RUBY\_HEAP\_SLOTS\_GROWTH\_FACTOR. patched by tmm1(Aman Gupta). [Feature #8015] [ruby-core:53131]

##### Revision 39746 - 03/13/2013 02:52 PM - nari

- gc.c: allow to tune growth of heap by environment variable RUBY\_HEAP\_SLOTS\_GROWTH\_FACTOR. patched by tmm1(Aman Gupta). [Feature #8015] [ruby-core:53131]

##### Revision 39746 - 03/13/2013 02:52 PM - nari

- gc.c: allow to tune growth of heap by environment variable RUBY\_HEAP\_SLOTS\_GROWTH\_FACTOR. patched by tmm1(Aman Gupta). [Feature #8015] [ruby-core:53131]

#### History

##### #1 - 03/05/2013 10:53 AM - kosaki (Motohiro KOSAKI)

The idea seems good.

- growth\_factor\_ptr = getenv("RUBY\_HEAP\_SLOTS\_GROWTH\_FACTOR");
- if (growth\_factor\_ptr != NULL) {
- double growth\_factor\_f = atof(growth\_factor\_ptr);

atof() don't have proper error check. it should be strtod().

- if (RTEST(ruby\_verbose))
- fprintf(stderr, "growth\_factor=%f (%f)\n", growth\_factor\_f, initial\_growth\_factor);
- if (growth\_factor\_f > 0) {
- initial\_growth\_factor = growth\_factor\_f;
- }

This seems don't work when `growth_factor_f` is less than 1.

## #2 - 03/05/2013 12:38 PM - tmm1 (Aman Gupta)

Thanks for the review. How about this?

@@ -3366,10 +3535,11 @@ rb\_gc\_set\_params(void)

```
growth_factor_ptr = getenv("RUBY_HEAP_SLOTS_GROWTH_FACTOR");
if (growth_factor_ptr != NULL) {
```

- `double growth_factor_f = atof(growth_factor_ptr);`
- `double growth_factor_f = strtod(growth_factor_ptr, NULL);` if (RTEST(ruby\_verbose))
- `fprintf(stderr, "growth_factor=%f (%f)\n", growth_factor_f, initial_growth_factor);`
- `if (growth_factor_f > 0) {`
- `fprintf(stderr, "heap_slots_growth_factor=%f (%f)\n",`
- `growth_factor_f, initial_growth_factor);`
- `if (growth_factor_f > 1) { initial_growth_factor = growth_factor_f; } }`

## #3 - 03/08/2013 04:01 PM - tmm1 (Aman Gupta)

Are there any objections to this patch?

## #4 - 03/08/2013 09:25 PM - authorNari (Narihiro Nakamura)

- Assignee set to *authorNari (Narihiro Nakamura)*

I agree this patch. I'll merge it few days later, if there are no objections.

Thank you!

## #5 - 03/11/2013 12:34 PM - drbrain (Eric Hodel)

- Category set to *core*

- Status changed from *Open* to *Assigned*

- Target version set to *2.1.0*

## #6 - 03/13/2013 11:52 PM - authorNari (Narihiro Nakamura)

- Status changed from *Assigned* to *Closed*

- % Done changed from *0* to *100*

This issue was solved with changeset r39746.

Aman, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

- 
- gc.c: allow to tune growth of heap by environment variable `RUBY_HEAP_SLOTS_GROWTH_FACTOR`. patched by tmm1(Aman Gupta). [Feature [#8015](#)] [ruby-core:53131]