

Ruby master - Bug #6006

Fix calculation of HEAP_OBJ_LIMIT and HEAP_BITMAP_LIMIT

02/12/2012 01:31 AM - funny_falcon (Yura Sokolov)

Status:	Closed	Backport:
Priority:	Normal	
Assignee:	authorNari (Narihiro Nakamura)	
Target version:	2.0.0	
ruby -v:	ruby 2.0.0dev (2012-02-11 trunk 34555) [i686-linux]	
Description https://github.com/ruby/ruby/pull/92		

Associated revisions

Revision 261400e7 - 02/13/2012 12:57 PM - nari

- gc.c (HEAP_OBJ_LIMIT, HEAP_BITMAP_LIMIT): HEAP_OBJ_LIMIT used sizeof(struct heaps_slot) while heap is currently allocated with struct heaps_header. HEAP_BITMAP_LIMIT were calculated from HEAP_OBJ_LIMIT/sizeof(uintptr_t) - one Byte for each object, not one Bit. [Bug #6006] patched by Sokolov Yura. <https://github.com/ruby/ruby/pull/92>

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@34581 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 34581 - 02/13/2012 12:57 PM - nari

- gc.c (HEAP_OBJ_LIMIT, HEAP_BITMAP_LIMIT): HEAP_OBJ_LIMIT used sizeof(struct heaps_slot) while heap is currently allocated with struct heaps_header. HEAP_BITMAP_LIMIT were calculated from HEAP_OBJ_LIMIT/sizeof(uintptr_t) - one Byte for each object, not one Bit. [Bug #6006] patched by Sokolov Yura. <https://github.com/ruby/ruby/pull/92>

Revision 34581 - 02/13/2012 12:57 PM - nari

- gc.c (HEAP_OBJ_LIMIT, HEAP_BITMAP_LIMIT): HEAP_OBJ_LIMIT used sizeof(struct heaps_slot) while heap is currently allocated with struct heaps_header. HEAP_BITMAP_LIMIT were calculated from HEAP_OBJ_LIMIT/sizeof(uintptr_t) - one Byte for each object, not one Bit. [Bug #6006] patched by Sokolov Yura. <https://github.com/ruby/ruby/pull/92>

Revision 34581 - 02/13/2012 12:57 PM - nari

- gc.c (HEAP_OBJ_LIMIT, HEAP_BITMAP_LIMIT): HEAP_OBJ_LIMIT used sizeof(struct heaps_slot) while heap is currently allocated with struct heaps_header. HEAP_BITMAP_LIMIT were calculated from HEAP_OBJ_LIMIT/sizeof(uintptr_t) - one Byte for each object, not one Bit. [Bug #6006] patched by Sokolov Yura. <https://github.com/ruby/ruby/pull/92>

Revision 34581 - 02/13/2012 12:57 PM - nari

- gc.c (HEAP_OBJ_LIMIT, HEAP_BITMAP_LIMIT): HEAP_OBJ_LIMIT used sizeof(struct heaps_slot) while heap is currently allocated with struct heaps_header. HEAP_BITMAP_LIMIT were calculated from HEAP_OBJ_LIMIT/sizeof(uintptr_t) - one Byte for each object, not one Bit. [Bug #6006] patched by Sokolov Yura. <https://github.com/ruby/ruby/pull/92>

Revision 34581 - 02/13/2012 12:57 PM - nari

- gc.c (HEAP_OBJ_LIMIT, HEAP_BITMAP_LIMIT): HEAP_OBJ_LIMIT used sizeof(struct heaps_slot) while heap is currently allocated with struct heaps_header. HEAP_BITMAP_LIMIT were calculated from HEAP_OBJ_LIMIT/sizeof(uintptr_t) - one Byte for each object, not one Bit. [Bug #6006] patched by Sokolov Yura. <https://github.com/ruby/ruby/pull/92>

Revision 34581 - 02/13/2012 12:57 PM - nari

- gc.c (HEAP_OBJ_LIMIT, HEAP_BITMAP_LIMIT): HEAP_OBJ_LIMIT used sizeof(struct heaps_slot) while heap is currently allocated with struct heaps_header. HEAP_BITMAP_LIMIT were calculated from HEAP_OBJ_LIMIT/sizeof(uintptr_t) - one Byte for each object, not one Bit. [Bug #6006] patched by Sokolov Yura. <https://github.com/ruby/ruby/pull/92>

Revision 8b4b4032 - 02/15/2012 02:09 AM - nari

- gc.c (HEAP_BITMAP_LIMIT): HEAP_BITMAP_LIMIT is computed on the basis of HEAP_SIZE because it must covers a whole heap block. [ruby-trunk - Bug #6006]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@34618 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

Revision 34618 - 02/15/2012 02:09 AM - nari

- gc.c (HEAP_BITMAP_LIMIT): HEAP_BITMAP_LIMIT is computed on the basis of HEAP_SIZE because it must covers a whole heap block. [ruby-trunk - Bug #6006]

Revision 34618 - 02/15/2012 02:09 AM - nari

- gc.c (HEAP_BITMAP_LIMIT): HEAP_BITMAP_LIMIT is computed on the basis of HEAP_SIZE because it must covers a whole heap block. [ruby-trunk - Bug #6006]

Revision 34618 - 02/15/2012 02:09 AM - nari

- gc.c (HEAP_BITMAP_LIMIT): HEAP_BITMAP_LIMIT is computed on the basis of HEAP_SIZE because it must covers a whole heap block. [ruby-trunk - Bug #6006]

Revision 34618 - 02/15/2012 02:09 AM - nari

- gc.c (HEAP_BITMAP_LIMIT): HEAP_BITMAP_LIMIT is computed on the basis of HEAP_SIZE because it must covers a whole heap block. [ruby-trunk - Bug #6006]

Revision 34618 - 02/15/2012 02:09 AM - nari

- gc.c (HEAP_BITMAP_LIMIT): HEAP_BITMAP_LIMIT is computed on the basis of HEAP_SIZE because it must covers a whole heap block. [ruby-trunk - Bug #6006]

Revision 34618 - 02/15/2012 02:09 AM - nari

- gc.c (HEAP_BITMAP_LIMIT): HEAP_BITMAP_LIMIT is computed on the basis of HEAP_SIZE because it must covers a whole heap block. [ruby-trunk - Bug #6006]

History

#1 - 02/12/2012 01:44 PM - funny_falcon (Yura Sokolov)

Sorry for issue without description.
Details:

HEAP_OBJ_LIMIT used sizeof(struct heaps_slot) while heap is currently allocated with struct heaps_header
HEAP_BITMAP_LIMIT were calculated from HEAP_OBJ_LIMIT/sizeof(uintptr_t) - one Byte for each object, not one Bit.

#2 - 02/13/2012 08:37 AM - authorNari (Narihiro Nakamura)

- Assignee set to authorNari (Narihiro Nakamura)

#3 - 02/13/2012 09:57 PM - authorNari (Narihiro Nakamura)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset r34581.
Yura, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

-
- gc.c (HEAP_OBJ_LIMIT, HEAP_BITMAP_LIMIT): HEAP_OBJ_LIMIT used sizeof(struct heaps_slot) while heap is currently allocated with struct heaps_header. HEAP_BITMAP_LIMIT were calculated from HEAP_OBJ_LIMIT/sizeof(uintptr_t) - one Byte for each object, not one Bit. [Bug #6006] patched by Sokolov Yura. <https://github.com/ruby/ruby/pull/92>

#4 - 02/14/2012 05:02 PM - naruse (Yui NARUSE)

- Status changed from Closed to Assigned

Narihiro Nakamura wrote:

This issue was solved with changeset r34581.

- gc.c (HEAP_OBJ_LIMIT, HEAP_BITMAP_LIMIT): HEAP_OBJ_LIMIT used sizeof(struct heaps_slot) while heap is currently allocated with struct heaps_header. HEAP_BITMAP_LIMIT were calculated from HEAP_OBJ_LIMIT/sizeof(uintptr_t) - one Byte for each object, not one Bit. [Bug #6006] patched by Sokolov Yura. <https://github.com/ruby/ruby/pull/92>

This seems to break CentOS 5.6 64bit.

<http://c5664.ruby-ci.org/~chkbuild/ruby-trunk/log/20120213T130301Z.diff.html.gz>

But it doesn't always happen...

<http://c5664.ruby-ci.org/~chkbuild/ruby-trunk/recent.html>

#5 - 02/14/2012 05:43 PM - funny_falcon (Yura Sokolov)

I think it will fix <https://github.com/funny-falcon/ruby/commit/5945932de2d682b4fe9b5de3d32b07c2d9b5179d> (updated <https://github.com/ruby/ruby/pull/92>)

It happens cause on 64-bit platform sometime there should be objs==2 instead of objs--.
But it easier to compute objs exactly.

#6 - 02/14/2012 06:52 PM - funny_falcon (Yura Sokolov)

Another one update: calculate p and objs without conditions at all.
<https://github.com/ruby/ruby/pull/92.diff>

#7 - 02/14/2012 08:42 PM - authorNari (Narihiro Nakamura)

- Status changed from Assigned to Closed

This issue seems to be solved at r34597.

<http://c5664.ruby-ci.org/~chkbuild/ruby-trunk/log/20120214T110302Z.log.html.gz>

Thanks :)

#8 - 02/15/2012 04:29 AM - naruse (Yui NARUSE)

- Status changed from Closed to Assigned

It still happens: <http://c5664.ruby-ci.org/~chkbuild/ruby-trunk/log/20120214T170302Z.diff.html.gz>

I note that I locally applied Yura's revised patch on [ruby-core:42605] to my env (FreeBSD 9.0 64bit), but it happens segv on miniruby.

```
compiling ../../ruby/dmyext.c
linking miniruby
../../ruby/tool/mkconfig.rb: [BUG] Segmentation fault
ruby 2.0.0dev (2012-02-14 trunk 34599) [x86_64-freebsd9.0]
```

```
-- Control frame information -----
c:0001 p:0000 s:0002 b:0002 l:0006c8 d:0006c8 TOP
```

```
-- C level backtrace information -----
0x44a807 at /home/naruse/obj/ruby/miniruby ../../ruby/error.c:266
0x44a922 at /home/naruse/obj/ruby/miniruby ../../ruby/error.c:285
0x5126f1 at /home/naruse/obj/ruby/miniruby ../../ruby/signal.c:603
0x800c9d723 <_pthread_sigmask+707> at /lib/libthr.so.3
0x800c9d897 <_pthread_sigmask+1079> at /lib/libthr.so.3
0x7fffffff003
0x4c1b9d at /home/naruse/obj/ruby/miniruby ../../ruby/parse.y:7236
0x4c60b9 at /home/naruse/obj/ruby/miniruby ../../ruby/parse.y:8515
0x4b0c6e at /home/naruse/obj/ruby/miniruby parse.c:5182
0x4bd926 at /home/naruse/obj/ruby/miniruby ../../ruby/parse.y:5798
0x59bd1d at /home/naruse/obj/ruby/miniruby ../../ruby/thread.c:4510
0x59bbc9 at /home/naruse/obj/ruby/miniruby ../../ruby/thread.c:4486
0x4bdab3 at /home/naruse/obj/ruby/miniruby ../../ruby/parse.y:5824
0x4be023 at /home/naruse/obj/ruby/miniruby ../../ruby/parse.y:5954
0x5110f0 at /home/naruse/obj/ruby/miniruby ../../ruby/ruby.c:1631
0x4512ba at /home/naruse/obj/ruby/miniruby ../../ruby/eval.c:742
0x511202 at /home/naruse/obj/ruby/miniruby ../../ruby/ruby.c:1668
0x510318 at /home/naruse/obj/ruby/miniruby ../../ruby/ruby.c:1404
0x5117c5 at /home/naruse/obj/ruby/miniruby ../../ruby/ruby.c:1808
0x44f75f at /home/naruse/obj/ruby/miniruby ../../ruby/eval.c:73
```

0x41682d <_start+349> at /home/naruse/obj/ruby/miniruby ../../ruby/main.c:38
0x41675e <_start+142> at /home/naruse/obj/ruby/miniruby

-- Other runtime information -----

- Loaded script: ../../ruby/tool/mkconfig.rb
- Loaded features:
0 enumerator.so

[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>

*** Signal 6

Stop in /home/naruse/obj/ruby.

#9 - 02/15/2012 11:09 AM - authorNari (Narihiro Nakamura)

- Status changed from Assigned to Closed

This issue was solved with changeset r34618.
Yura, thank you for reporting this issue.
Your contribution to Ruby is greatly appreciated.
May Ruby be with you.

-
- gc.c (HEAP_BITMAP_LIMIT): HEAP_BITMAP_LIMIT is computed on the basis of HEAP_SIZE because it must covers a whole heap block.
[ruby-trunk - Bug #6006]