

# Ruby trunk - Feature #7095

## Non-recursive marking

10/02/2012 12:21 AM - authorNari (Narihiro Nakamura)

<b>Status:</b>	Closed
<b>Priority:</b>	Normal
<b>Assignee:</b>	authorNari (Narihiro Nakamura)
<b>Target version:</b>	2.0.0

### Description

nari

GC

[https://github.com/authorNari/ruby/compare/non\\_recursion\\_marking](https://github.com/authorNari/ruby/compare/non_recursion_marking)  
[https://github.com/authorNari/ruby/compare/non\\_recursion\\_marking.patch](https://github.com/authorNari/ruby/compare/non_recursion_marking.patch)

=  
gc\_mark()  
GC

2

- 1.
- 2.

1. GC

2. <https://bugs.ruby-lang.org/issues/3781>  
FIBER\_USE\_NATIVE  
SEGV

SEGV

=

= mark benchmark OPTS="-r 5"  
"average total difference is -7.793935319666676"

<https://gist.github.com/3806667>  
(target 0, target 1  
"average total difference is -7.793935319666676"

<https://gist.github.com/3812118>

depth 240  
origin GC(sec): 1.96  
non-recursive GC(sec): 1.87

depth 500  
origin GC(sec): 14.73  
non-recursive GC(sec): 5.49

#### Related issues:

Related to Ruby trunk - Bug #7141: ALT\_STACK\_SIZE is not enough

Closed

10/11/2012

#### Associated revisions

##### Revision 8d53a08e - 10/03/2012 12:30 PM - nari

- gc.c: Use the non-recursive marking instead of recursion. The recursion marking of CRuby needs checking stack overflow and the fail-safe system, but these systems not good at partial points, for example, marking deep tree structures. [ruby-dev:46184] [Feature #7095]
- configure.in (GC\_MARK\_STACKFRAME\_WORD): removed. It's used by checking stack overflow of marking.
- win32/Makefile.sub (GC\_MARK\_STACKFRAME\_WORD): ditto.

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37075 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

##### Revision 37075 - 10/03/2012 12:30 PM - nari

- gc.c: Use the non-recursive marking instead of recursion. The recursion marking of CRuby needs checking stack overflow and the fail-safe system, but these systems not good at partial points, for example, marking deep tree structures. [ruby-dev:46184] [Feature #7095]
- configure.in (GC\_MARK\_STACKFRAME\_WORD): removed. It's used by checking stack overflow of marking.
- win32/Makefile.sub (GC\_MARK\_STACKFRAME\_WORD): ditto.

##### Revision 37075 - 10/03/2012 12:30 PM - nari

- gc.c: Use the non-recursive marking instead of recursion. The recursion marking of CRuby needs checking stack overflow and the fail-safe system, but these systems not good at partial points, for example, marking deep tree structures. [ruby-dev:46184] [Feature #7095]
- configure.in (GC\_MARK\_STACKFRAME\_WORD): removed. It's used by checking stack overflow of marking.
- win32/Makefile.sub (GC\_MARK\_STACKFRAME\_WORD): ditto.

##### Revision 37075 - 10/03/2012 12:30 PM - nari

- gc.c: Use the non-recursive marking instead of recursion. The recursion marking of CRuby needs checking stack overflow and the

fail-safe system, but these systems not good at partial points, for example, marking deep tree structures. [ruby-dev:46184] [Feature #7095]

- `configure.in` (GC\_MARK\_STACKFRAME\_WORD): removed. It's used by checking stack overflow of marking.
- `win32/Makefile.sub` (GC\_MARK\_STACKFRAME\_WORD): ditto.

#### Revision 37075 - 10/03/2012 12:30 PM - nari

- `gc.c`: Use the non-recursive marking instead of recursion. The recursion marking of CRuby needs checking stack overflow and the fail-safe system, but these systems not good at partial points, for example, marking deep tree structures. [ruby-dev:46184] [Feature #7095]
- `configure.in` (GC\_MARK\_STACKFRAME\_WORD): removed. It's used by checking stack overflow of marking.
- `win32/Makefile.sub` (GC\_MARK\_STACKFRAME\_WORD): ditto.

#### Revision 37075 - 10/03/2012 12:30 PM - nari

- `gc.c`: Use the non-recursive marking instead of recursion. The recursion marking of CRuby needs checking stack overflow and the fail-safe system, but these systems not good at partial points, for example, marking deep tree structures. [ruby-dev:46184] [Feature #7095]
- `configure.in` (GC\_MARK\_STACKFRAME\_WORD): removed. It's used by checking stack overflow of marking.
- `win32/Makefile.sub` (GC\_MARK\_STACKFRAME\_WORD): ditto.

#### Revision 37075 - 10/03/2012 12:30 PM - nari

- `gc.c`: Use the non-recursive marking instead of recursion. The recursion marking of CRuby needs checking stack overflow and the fail-safe system, but these systems not good at partial points, for example, marking deep tree structures. [ruby-dev:46184] [Feature #7095]
- `configure.in` (GC\_MARK\_STACKFRAME\_WORD): removed. It's used by checking stack overflow of marking.
- `win32/Makefile.sub` (GC\_MARK\_STACKFRAME\_WORD): ditto.

#### Revision ec62e301 - 10/04/2012 02:48 PM - nari

- `gc.c` (`init_heap`): call `init_mark_stack` before to allocate altstack. This change avoid the stack overflow at the signal handler on 32bit, but I don't understand reason... [Feature #7095]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/trunk@37088 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

#### Revision 37088 - 10/04/2012 02:48 PM - nari

- gc.c (init\_heap): call init\_mark\_stack before to allocate altstack. This change avoid the stack overflow at the signal handler on 32bit, but I don't understand reason... [Feature #7095]

**Revision 37088 - 10/04/2012 02:48 PM - nari**

- gc.c (init\_heap): call init\_mark\_stack before to allocate altstack. This change avoid the stack overflow at the signal handler on 32bit, but I don't understand reason... [Feature #7095]

**Revision 37088 - 10/04/2012 02:48 PM - nari**

- gc.c (init\_heap): call init\_mark\_stack before to allocate altstack. This change avoid the stack overflow at the signal handler on 32bit, but I don't understand reason... [Feature #7095]

**Revision 37088 - 10/04/2012 02:48 PM - nari**

- gc.c (init\_heap): call init\_mark\_stack before to allocate altstack. This change avoid the stack overflow at the signal handler on 32bit, but I don't understand reason... [Feature #7095]

**Revision 37088 - 10/04/2012 02:48 PM - nari**

- gc.c (init\_heap): call init\_mark\_stack before to allocate altstack. This change avoid the stack overflow at the signal handler on 32bit, but I don't understand reason... [Feature #7095]

**Revision 37088 - 10/04/2012 02:48 PM - nari**

- gc.c (init\_heap): call init\_mark\_stack before to allocate altstack. This change avoid the stack overflow at the signal handler on 32bit, but I don't understand reason... [Feature #7095]

**Revision ae2df330 - 11/13/2012 10:02 AM - usa (Usaku NAKAMURA)**

merged revision(s) 37075,37076,37082,37083,37088: [Backport #7123]

- gc.c: Use the non-recursive marking instead of recursion. The recursion marking of CRuby needs checking stack overflow and the fail-safe system, but these systems not good at partial points, for example, marking deep tree structures. [ruby-dev:46184] [Feature #7095]
- configure.in (GC\_MARK\_STACKFRAME\_WORD): removed. It's used by checking stack overflow of marking.
- win32/Makefile.sub (GC\_MARK\_STACKFRAME\_WORD): ditto.
- gc.c (free\_stack\_chunks): it is used only when per-VM object space is enabled.
- gc.c (rb\_objspace\_call\_finalizer): mark self-referencing finalizers before run finalizers, to fix SEGV from btest on 32bit.
- gc.c (gc\_mark\_stacked\_objects): extract from gc\_marks().
- gc.c (rb\_objspace\_call\_finalizer): call gc\_mark\_stacked\_objects at suitable point.
- gc.c (init\_heap): call init\_mark\_stack before to allocate altstack. This change avoid the stack overflow at the signal handler on 32bit, but I don't understand reason... [Feature #7095]

git-svn-id: svn+ssh://ci.ruby-lang.org/ruby/branches/ruby\_1\_9\_3@37648 b2dd03c8-39d4-4d8f-98ff-823fe69b080e

## History

---

**#1 - 10/02/2012 01:23 AM - ko1 (Koichi Sasada)**

(2012/10/02 0:21), authorNari (Narihiro Nakamura) wrote:

Issue [#7095](#) has been reported by authorNari (Narihiro Nakamura).

---

Bug #7095: Non-recursive marking  
<https://bugs.ruby-lang.org/issues/7095>

Author: authorNari (Narihiro Nakamura)  
Status: Open  
Priority: Normal  
Assignee: authorNari (Narihiro Nakamura)  
Category: core  
Target version: 2.0.0  
ruby -v: ruby 2.0.0dev (2012-09-25 trunk 37032) [x86\_64-linux]

nari

GC

: [https://github.com/authorNari/ruby/compare/non\\_recursion\\_marking](https://github.com/authorNari/ruby/compare/non_recursion_marking)  
: [https://github.com/authorNari/ruby/compare/non\\_recursion\\_marking.patch](https://github.com/authorNari/ruby/compare/non_recursion_marking.patch)

```
=
gc_mark()
GC
```

+1

- page
- GC malloc + alloc
- +#ifndef LP64 -2

--  
// SASADA Koichi at atdot dot net

**#2 - 10/02/2012 03:23 AM - kosaki (Motohiro KOSAKI)**

2012/10/1 SASADA Koichi [ko1@atdot.net](mailto:ko1@atdot.net):

(2012/10/02 0:21), authorNari (Narihiro Nakamura) wrote:

Issue #7095 has been reported by authorNari (Narihiro Nakamura).

Bug #7095: Non-recursive marking  
<https://bugs.ruby-lang.org/issues/7095>

Author: authorNari (Narihiro Nakamura)  
Status: Open  
Priority: Normal  
Assignee: authorNari (Narihiro Nakamura)  
Category: core  
Target version: 2.0.0  
ruby -v: ruby 2.0.0dev (2012-09-25 trunk 37032) [x86\_64-linux]

nari

GC

: [https://github.com/authorNari/ruby/compare/non\\_recursion\\_marking](https://github.com/authorNari/ruby/compare/non_recursion_marking)  
: [https://github.com/authorNari/ruby/compare/non\\_recursion\\_marking.patch](https://github.com/authorNari/ruby/compare/non_recursion_marking.patch)

```
=
gc_mark()
GC
```

+1

+100

malloc

page

- page

page size 4k malloc

- GC malloc alloc

free mark

- +#ifndef LP64

Linux + glibc malloc header size\_t\*2

#3 - 10/02/2012 02:38 PM - authorNari (Narihiro Nakamura)

kosaki (Motohiro KOSAKI) wrote:

2012/10/1 SASADA Koichi ko1@atdot.net:

(2012/10/02 0:21), authorNari (Narihiro Nakamura) wrote:

(snip)

malloc

page

- page

page size 4k malloc

- +#ifndef LP64

Linux + glibc malloc header size\_t\*2

page chunk segment

page chunk segment

- GC malloc alloc

free mark

4 free stack

#4 - 10/03/2012 05:02 PM - authorNari (Narihiro Nakamura)

- Tracker changed from Bug to Feature

#5 - 10/03/2012 09:30 PM - authorNari (Narihiro Nakamura)

- Status changed from Open to Closed

- % Done changed from 0 to 100

This issue was solved with changeset [r37075](#).

Narihiro, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

---

- gc.c: Use the non-recursive marking instead of recursion. The recursion marking of CRuby needs checking stack overflow and the fail-safe system, but these systems not good at partial points, for example, marking deep tree structures. [ruby-dev:46184] [Feature [#7095](#)]
- configure.in (GC\_MARK\_STACKFRAME\_WORD): removed. It's used by checking stack overflow of marking.
- win32/Makefile.sub (GC\_MARK\_STACKFRAME\_WORD): ditto.

#### #6 - 10/04/2012 11:00 AM - naruse (Yui NARUSE)

- Status changed from Closed to Assigned

authorNari (Narihiro Nakamura) wrote:

This issue was solved with changeset [r37075](#).

Narihiro, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

### May Ruby be with you.

- gc.c: Use the non-recursive marking instead of recursion. The recursion marking of CRuby needs checking stack overflow and the fail-safe system, but these systems not good at partial points, for example, marking deep tree structures. [ruby-dev:46184] [Feature [#7095](#)]
- configure.in (GC\_MARK\_STACKFRAME\_WORD): removed. It's used by checking stack overflow of marking.
- win32/Makefile.sub (GC\_MARK\_STACKFRAME\_WORD): ditto.

```
Linux 32bit SEGV backtrace
```

```
%. /ruby -e'Process.kill :SEGV, $$'
```

```
-e:1: [BUG] Segmentation fault
```

```
ruby 2.0.0dev (2012-10-03 trunk 37076) [i686-linux]
```

```
zsh: segmentation fault ./ruby -e'Process.kill :SEGV, $$'
```

<http://u32.ruby-ci.org/~chkbuild/ruby-trunk/log/20121003T130302Z.diff.html.gz>

#### #7 - 10/04/2012 11:48 PM - authorNari (Narihiro Nakamura)

- Status changed from Assigned to Closed

This issue was solved with changeset [r37088](#).

Narihiro, thank you for reporting this issue.

Your contribution to Ruby is greatly appreciated.

May Ruby be with you.

---

- gc.c (init\_heap): call init\_mark\_stack before to allocate altstack. This change avoid the stack overflow at the signal handler on 32bit, but I don't understand reason... [Feature [#7095](#)]