Ruby - Bug #18829

GC_COMPACTION_SUPPORTED macro should be set and detected automatically.

06/14/2022 10:58 AM - jprokop (Jarek Prokop)

Status:	Closed		
Priority:	Normal		
Assignee:			
Target version:			
ruby -v:	ruby 3.2.0dev (2022-06-14T01:21:55Z master 048f14221c) [powerpc64le-linux]	**Backport:**	2.7: DONTNEED, 3.0: REQUIRED, 3.1: REQUIRED

Description

After backporting the patch [0] for GC compaction support from https://github.com/ruby/ruby/pull/5934 (for bug https://bugs.ruby-lang.org/issues/18779), I found that there is no mechanism to automatically define the GC_COMPACTION_SUPPORTED based on the platform.

This does not seem to be a major issue with the Ruby master branch as the Ruby GC page size is bumped up to 64K so platforms like ppc64le do not struggle anymore, however, if the Ruby GC page size is set to 16K as is the case with Ruby 3.1 for example, then the compaction is enabled even in cases when it should not be.

For master I can reach a failure if I set the Ruby GC page size to 16K by editing macro in gc.c (https://github.com/ruby/ruby/blob/master/gc.c#L863) to:

```
#define HEAP_PAGE_ALIGN_LOG 14
```

As there are no compile-time checks around this, after compiling ruby and using the following script I get an exception.
script:

```
if GC.respond_to?(:compact)
  GC.verify_compaction_references(double_heap: true, toward: :empty)
end

100_000.times do |i|
  puts "i: #{i}, count: #{count}"
end
```

Running it raises an exception even though the code is in a guard clause that should prevent this:

```
~/.rubies/ruby-master/bin/ruby ./reproducer.rb
<internal:gc>:251:in `verify_compaction_references': Compaction isn't available on this platform (
NotImplementedError)
 from ./reproducer.rb:4:in `<main>'
```

Right now, our workaround is to define the macro GC_COMPACTION_SUPPORTED only on platforms that we know do support compaction. Ideally, this should not be required and the configuration script or the gc.c file should take care of defining the macro correctly.

[0]
https://src.fedoraproject.org/fork/jackorp/rpms/ruby/blob/1e34def591d4615bde83a28ff6cd336aa5879a80/f/ruby-3.2.0-define-unsupported-gc-compaction-methods-as-rb_f_notimplement.patch

Associated revisions

Revision 52d42e702375446746164a0251e1a10bce813b78 - 06/16/2022 02:18 PM - peterzhu2118 (Peter Zhu)

Rename GC_COMPACTION_SUPPORTED

Naming this macro GC_COMPACTION_SUPPORTED is misleading because it
only checks whether compaction is supported at compile time.

[Bug #18829]

Revision 79eaaf2d0b641710613f16525e4b4c439dfe854e - 06/16/2022 02:18 PM - peterzhu2118 (Peter Zhu)

Include runtime checks for compaction support

Commit 0c36ba53192c5a0d245c9b626e4346a32d7d144e changed GC compaction methods to not be implemented when not supported. However, that commit only does compile time checks (which currently only checks for WASM), but there are additional compaction support checks during run time.

This commit changes it so that GC compaction methods aren't defined during run time if the platform does not support GC compaction.

[Bug #18829]

Revision 52d42e702375446746164a0251e1a10bce813b78 - 06/16/2022 02:18 PM - peterzhu2118 (Peter Zhu)

Rename GC_COMPACTION_SUPPORTED

Naming this macro GC_COMPACTION_SUPPORTED is misleading because it only checks whether compaction is supported at compile time.

[Bug #18829]

Revision 79eaaf2d0b641710613f16525e4b4c439dfe854e - 06/16/2022 02:18 PM - peterzhu2118 (Peter Zhu)

Include runtime checks for compaction support

Commit 0c36ba53192c5a0d245c9b626e4346a32d7d144e changed GC compaction methods to not be implemented when not supported. However, that commit only does compile time checks (which currently only checks for WASM), but there are additional compaction support checks during run time.

This commit changes it so that GC compaction methods aren't defined during run time if the platform does not support GC compaction.

[Bug #18829]

Revision 52d42e70 - 06/16/2022 02:18 PM - peterzhu2118 (Peter Zhu)

Rename GC_COMPACTION_SUPPORTED

Naming this macro GC_COMPACTION_SUPPORTED is misleading because it only checks whether compaction is supported at compile time.

[Bug #18829]

Revision 79eaaf2d - 06/16/2022 02:18 PM - peterzhu2118 (Peter Zhu)

Include runtime checks for compaction support

Commit 0c36ba53192c5a0d245c9b626e4346a32d7d144e changed GC compaction methods to not be implemented when not supported. However, that commit only does compile time checks (which currently only checks for WASM), but there are additional compaction support checks during run time.

This commit changes it so that GC compaction methods aren't defined during run time if the platform does not support GC compaction.

[Bug #18829]

History

#1 - 06/14/2022 07:54 PM - jeremyevans0 (Jeremy Evans)

- Status changed from Open to Closed

- Backport changed from 2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN to 2.7: DONTNEED, 3.0: REQUIRED, 3.1: REQUIRED

Since this problem does not occur in the master branch (due to the increased page size), it seems like the bug has already been fixed/worked around. It seems unlikely we would backport the increased page size to older Ruby versions, and we don't generally apply patches to release branches that are not backported from master (it only happens in exceptional cases, and this case doesn't appear to warrant it).

I'm marking this for backport to supported versions, but the odds of a fix being backported without you providing a patch seem low. Even with a patch, it would be up to the branch maintainer whether to apply it.

Another possibility would be adding a patch to the master branch for this, and then asking for that patch to be backported, but if this is no longer an issue in the master branch due to the increased page size, I would be against applying such a patch.

#2 - 06/14/2022 08:26 PM - peterzhu2118 (Peter Zhu)

- Status changed from Closed to Open

Thank you for the ticket. I have a patch that should fix this issue, please try it out and let me know if it works.

#3 - 06/16/2022 02:19 PM - peterzhu2118 (Peter Zhu)

- Status changed from Open to Closed

Applied in changeset git|52d42e702375446746164a0251e1a10bce813b78.

Rename GC_COMPACTION_SUPPORTED

Naming this macro GC_COMPACTION_SUPPORTED is misleading because it
only checks whether compaction is supported at compile time.

[Bug #18829]

#4 - 06/22/2022 07:06 AM - jaruga (Jun Aruga)

I think the backport is to apply the 3 commits on the master branch in the chronological order below to a stable branch.
https://github.com/ruby/ruby/commit/52d42e702375446746164a0251e1a10bce813b78
https://github.com/ruby/ruby/commit/79eaaf2d0b641710613f16525e4b4c439dfe854e
https://github.com/ruby/ruby/commit/2c190863239bee3f54cfb74b16bb6ea4cae6ed20