Ruby master - Feature #14462
MJIT enabled should be displayed in the version string
02/10/2018 09:23 PM - chrisseaton (Chris Seaton)

Status: Closed
Priority: Normal
Assignee: k0kubun (Takashi Kokubun)
Target version:

Description
Whether or not MJIT is enabled will probably be an important part of diagnosing bugs. It's also useful for when benchmarking - we often run benchmarks with -v, and other implementations that have JITs give information about whether their JIT is enabled and with what basic options in this string.

For example TruffleRuby (Graal being the JIT)

```
truffleruby 0.31, like ruby 2.3.5 <GraalVM 0.31 1.8.0_151-b12 with Graal> [darwin-x86_64]
```

And JRuby

```
jruby 9.1.13.0 (2.3.3) 2017-09-06 8e1c115 Java HotSpot(TM) 64-Bit Server VM 25.144-b01 on 1.8.0_14 4-b01 +jit [darwin-x86_64]
```

CRuby could display the version string something like this

```
ruby 2.6.0dev (2018-02-07 jit-version 62270) [x86_64-darwin17] +JIT
```

I've attached a patch, but I wasn't sure how to get the information into the version module, and I wasn't sure if rb_version was part of the API so I didn't want to modify that, so it may be a bit of a messy solution I'm afraid.

Associated revisions
Revision 171c496e - 02/22/2018 02:53 PM - k0kubun (Takashi Kokubun)
version.c: show +JIT when --jit is passed
in version output.
version.h: ditto
ruby.c: propagate option for it
common.mk: updated dependency for version.c

mjit.c: overwrites the RUBY_DESCRIPTION to have +JIT when --jit is passed

test/ruby/test_rubyoptions.rb: add test for them

Only ruby --jit -v will have "+JIT", but this is intentional.
This may not be convenient for debugging by ticket with ruby -v,
but it's convenient for benchmark tools that pass options (--jit)
when showing it. At least such behavior is planned for benchmark_driver.gem
and this behavior is designed for it. Other benchmark tools are
recommended to follow the behavior too if they show version.
RUBY_DESCRIPTION might be useful for it too.

The position of "+JIT" is changed from original proposal because other
platforms like JRuby and TruffleRuby end it with architecture.
It's made similar to JRuby, but it's upper-cased because Matz made approval
for "+JIT" in the ticket.

Example:
```
$ ruby -v
ruby 2.6.0dev (2018-02-22 trunk 62529) [x86_64-linux]
$ ruby --jit -v
ruby 2.6.0dev (2018-02-22 trunk 62529) +JIT [x86_64-linux]
```

After --jit is made default in the future, this output may be removed.
So do not rely on this output if possible.

[Feature #14462]
Revision 62530 - 02/22/2018 02:53 PM - k0kubun (Takashi Kokubun)

version.c: show +JIT when --jit is passed

in version output.

version.h: ditto

ruby.c: propagate option for it

common.mk: updated dependency for version.c

mjit.c: overwrites the RUBY_DESCRIPTION to have +JIT when --jit is passed

test/ruby/test_rubyoptions.rb: add test for them

Only ruby --jit -v will have "+JIT", but this is intentional.
This may not be convenient for debugging by ticket with ruby -v,
but it's convenient for benchmark tools that pass options (--jit)
when showing it. At least such behavior is planned for benchmark_driver.gem
and this behavior is designed for it. Other benchmark tools are
recommended to follow the behavior too if they show version.
RUBY_DESCRIPTION might be useful for it too.

The position of "+JIT" is changed from original proposal because other
platforms like JRuby and TruffleRuby end it with architecture.
It's made similar to JRuby, but it's upper-cased because Matz made approval
for "+JIT" in the ticket.

Example:

$ ruby -v
ruby 2.6.0dev (2018-02-22 trunk 62529) [x86_64-linux]
$ ruby --jit -v
ruby 2.6.0dev (2018-02-22 trunk 62529) +JIT [x86_64-linux]

After --jit is made default in the future, this output may be removed.
So do not rely on this output if possible.

[Feature #14462]

Revision 62530 - 02/22/2018 02:53 PM - k0kubun (Takashi Kokubun)

version.c: show +JIT when --jit is passed

in version output.

version.h: ditto

ruby.c: propagate option for it

common.mk: updated dependency for version.c

mjit.c: overwrites the RUBY_DESCRIPTION to have +JIT when --jit is passed

test/ruby/test_rubyoptions.rb: add test for them

Only ruby --jit -v will have "+JIT", but this is intentional.
This may not be convenient for debugging by ticket with ruby -v,
but it's convenient for benchmark tools that pass options (--jit)
when showing it. At least such behavior is planned for benchmark_driver.gem
and this behavior is designed for it. Other benchmark tools are
recommended to follow the behavior too if they show version.
RUBY_DESCRIPTION might be useful for it too.

The position of "+JIT" is changed from original proposal because other
platforms like JRuby and TruffleRuby end it with architecture.
It's made similar to JRuby, but it's upper-cased because Matz made approval
for "+JIT" in the ticket.

Example:

$ ruby -v
ruby 2.6.0dev (2018-02-22 trunk 62529) [x86_64-linux]
$ ruby --jit -v
ruby 2.6.0dev (2018-02-22 trunk 62529) +JIT [x86_64-linux]

After --jit is made default in the future, this output may be removed.
So do not rely on this output if possible.

[Feature #14462]
I somewhat agree with the general notion of the suggestion. I am not entirely sure whether it should also be added to the ruby version string, but I also have no problem with it either, so I am fine either way.

For benchmarks, it may be useful to automatically add the information whether mjit is enabled or not, but to also provide means to disable that extra information when a ruby user wants to do so. Your suggestion to do so not by default but via -v is also fine (in this case it would mean that, if we omit "-v" there, we'd not get the information that mjit is used, in benchmarks; correct me if I am wrong please, I just want to define some of the behaviour).

Considering that mjit will be an important addition in 2018, the more information and documentation that exists with it, the better, in my opinion - perhaps even have a subsection on the official docs that refer specifically to mjit and what it is doing (does not have to be long, just a little page for people to get a headstart about mjit, when they never heard about it before; not everyone follows all changes or watches every video on ruby conferences :)).

I assume that it helps to distinguish causes of bugs too, so I'm in favor of it.

I think this makes sense and makes it easier to distinguish for benchmarks/tests/bug reports/etc.

I think the +JIT should be part of RUBY_DESCRIPTION for consistency.

But that seems a bit tricky, could you take a look at it, k0kubun (Takashi Kokubun)?

I agree with adding +JIT.

Matz.

version.c: show +JIT when --jit is passed

in version output.

version.h: ditto

ruby.c: propagate option for it

common.mk: updated dependency for version.c

mjit.c: overwrites the RUBY_DESCRIPTION to have +JIT when --jit is passed

Only ruby --jit -v will have "+JIT", but this is intentional.

This may not be convenient for debugging by ticket with ruby -v, but it's convenient for benchmark tools that pass options (--jit) when showing it. At least such behavior is planned for benchmark_driver.gem and this behavior is designed for it. Other benchmark tools are recommended to follow the behavior too if they show version. RUBY_DESCRIPTION might be useful for it too.

The position of "+JIT" is changed from original proposal because other platforms like JRuby and TruffleRuby end it with architecture. It's made similar to JRuby, but it's upper-cased because Matz made approval for "$JIT" in the ticket.
Example:

```bash
$ ruby -v
ruby 2.6.0dev (2018-02-22 trunk 62529) [x86_64-linux]
$ ruby --jit -v
ruby 2.6.0dev (2018-02-22 trunk 62529) +JIT [x86_64-linux]
```

After --jit is made default in the future, this output may be removed.
So do not rely on this output if possible.

[Feature #14462]

<table>
<thead>
<tr>
<th>Files</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>jit-version.diff</td>
<td>1.43 KB</td>
<td>02/10/2018</td>
<td>chrisseaton (Chris Seaton)</td>
</tr>
</tbody>
</table>