

Ruby master - Misc #13968

[Ruby 3.x perhaps] - A (minimal?) static variant of ruby

10/03/2017 08:42 PM - shevegen (Robert A. Heiler)

Status:	Open
Priority:	Normal
Assignee:	
Description	
<p>Hello ruby core team and everyone else,</p> <p>I think this has been brought up before, in one way or the other, but just in case it was not, or not in this context, I would like to propose that ruby can be compiled statically.</p> <p>Matz explained how this can be done for mruby:</p> <p>https://github.com/mruby/mruby/issues/3707</p> <p>This is not so difficult; allow me to copy/paste the major steps for mruby (I modified it a bit):</p> <p>Option A:</p> <pre>(1) compile your.rb files into a C file via mrbc. Example: "mrbc -Binit_yourlib -o mrblib.c *.rb" (2) your code/application has to call "init_yourlib(mrb)" after "mrb_open()" (3) compile your code/application with the generated C file (4) link via "libmruby.a" (5) static link options may have to be specified to the linker</pre> <p>Option B</p> <pre>(1) Create a binary "mrbgem". See mrbgems/mruby-bin-mirb for example.</pre> <p>And nobu added better support for statically linked extensions some years ago:</p> <p>https://bugs.ruby-lang.org/issues/9018</p> <p>Since time is a limited resource by the ruby core team, I understand that you have to prioritize on issues that are more important, or needed more; and a statically compiled ruby may not be of highest priority. Additionally, you may want to know the use case of people, as to why they may want to have a statically compiled variant of ruby.</p> <p>I have a (small) use case which I think has not been explained before.</p> <p>I often break stuff on my linux system when I compile programs. I use many ruby helper scripts to compile programs, so I need ruby. :)</p> <p>When I have a ruby available then I can often batch-compile, batch-install and so on. This works very well so far for most programs out there.</p> <p>Not always is there a host system available with ruby though; sometimes I may be in a confined environment, such as workspaces at university and a central server cluster to which I have no access (or rather, permission to change files/directory). So I tend to compile into the user directory there (home directory).</p> <p>Anyway. Since I tend to break stuff, but sometimes other stuff is also broken on such environments (looking at centos ...), I made it a habit to compile some of the core utilities in a static manner into my home directory; "make" for example or "sed". Both work very well. Others are a bit misbehaving or stubborn, such as awk - I haven't managed</p>	

a statically compiled variant of awk yet.

Of course I also use "busybox," a statically compiled busybox that is.

This allows me to recover from many problems or errors if things go awry (I can also use a live-DVD for recovery, at the least at home; or also download some binaries that I know will work on the target platform, so I have more ways to recover).

Now the thing is - most of what I do with busybox, has to do with file manipulation of some sort; setting new symlinks, removing, moving, creating files or directories.

Here I thought - I could use ruby rather than busybox. :)

But I'd want or need a statically compiled ruby, so that it does not break depending on other programs there on the system.

I hope I could explain my use case. And I agree, only very few people may ever want to have this perhaps.

I looked at my self-compiled MRI ruby, via "ldd", and these are the libraries that ruby depends on:

```
linux-vdso.so.1
libpthread.so.0 => /lib64/libpthread.so.0
libgmp.so.10 => /usr/lib/libgmp.so.10
libdl.so.2 => /lib64/libdl.so.2
libcrypt.so.1 => /lib64/libcrypt.so.1
libm.so.6 => /lib64/libm.so.6
libc.so.6 => /lib64/libc.so.6
/lib64/ld-linux-x86-64.so.2
```

I guess this is not minimal; I could probably re-compile ruby from a sandbox, to reduce the amount of .so objects that are needed, but the above is only an example anyway.

I assume that making available a ruby that can be statically compiled, is not trivial. It is probably easier for mruby, but mruby is, as far as I understand it, more aimed towards knowledgeable C/C++ hackers. I am a bit too scared to try mruby when I do not really know C.

I should also state that I do not need a "full" ruby by the way. I can live without the extension stuff, such as "readline" or "openssl" probably (though I'd love to have open-uri since that may help in recovery; ruby's open() functionality for remote data/websites is great).

I guess the core functionality I would need is, mostly, file manipulation.

So perhaps my request could be titled:

"A minimal variant of ruby - that is statically linked."

Now this may fit to mruby perhaps but I am a bit scared of mruby also because I do not know mruby very well; I mostly work with MRI ruby. (I did manage to compile mruby without problem and I could run ruby code too but I am not sure how much is available on mruby by default "out of the box". If I remember correctly, not every MRI ruby .rb file will work "out of the box")

I also thought of filing an issue request at github-mruby but then I thought that perhaps it may also fit into MRI ruby one day, which is why I prepended with the ruby 3.x label. (I suppose that it may not fit towards 2.x anymore since 2.x will probably focus on stability, bug fixing and so forth.)

I hope I could explain my use case. I am not sure if it is a good

use case; I can probably just keep on using busybox ... but I would prefer to actually work within a "ruby environment" in general, simply because ruby is a LOT nicer to work with, in particular for batch-automated tasks (I always prefer to use ruby rather than shell scripts, for example).

If possible it would be nice if this issue could remain open for a bit longer, if only to see if there is anyone else who may want to have some statically compiled variant of ruby.

I see this sometimes in other issue requests, or similar suggestions, so perhaps if there may be a sizeable amount of people who may want it, it could be put on a "long-term goal". Perhaps even past 3.x.

Last but not least I would like to repeat that I do not need a full featured ruby per se, a minimal may suffice - but that minimal ruby should ideally be easy to use+install. It's a bit like a mix between "MRI ruby" and "mruby" what I want, I think. :D

That way, I could both have a MRI ruby that uses .so, but also e. g a binary called "static-ruby" or "ruby-static" or something like that, that can be used a bit like a "recovery ruby".

I wrote a lot so now it's time to close without much further ado - thank you for reading!

History

#1 - 10/04/2017 06:21 AM - duerst (Martin Dürst)

Looking at common.mk, there are various places with the text "static". I'm not an expert in makefiles, but static compilation may already work.

#2 - 01/07/2018 01:50 AM - jwmittag (Jörg W Mittag)

Just to make sure I understand what you are talking about, when you write "Ruby", do you *really* mean "Ruby", or do you mean "YARV"?

#3 - 01/14/2018 03:24 PM - tak3 (Jerzy Grzebień)

What You think about mrubyc ?

Second idea is turn off float and stdio