Ruby master - Bug #17021
"arm64" and "arm" are mixed in RbConfig on Apple silicon

07/09/2020 11:57 AM - watson1978 (Shizuo Fujita)

Status: Closed
Priority: Normal
Assignee: 
Target version: 
ruby -v: ruby 2.8.0dev (2020-07-09T09:14:53Z master addbb7d16e) [arm64-darwin20]
Backport: 2.5: UNKNOWN, 2.6: UNKNOWN, 2.7: UNKNOWN

Description
When I checked the value of RbConfig::CONFIG on the Apple silicon macOS, I found "arm64" and "arm" are mixed as following.

$ ruby -v
ruby 2.8.0dev (2020-07-09T09:14:53Z master addbb7d16e) [arm64-darwin20]

$ ruby -e 'require "rbconfig"; pp RbConfig::CONFIG' | grep "arm"
"/Users/watson/.rbenv/versions/2.8.0/include/ruby-2.8.0/vendor_ruby/arm64-darwin20",
"/Users/watson/.rbenv/versions/2.8.0/include/ruby-2.8.0/site_ruby/arm64-darwin20",
"/Users/watson/.rbenv/versions/2.8.0/include/ruby-2.8.0/arm64-darwin20",
"/Users/watson/.rbenv/versions/2.8.0/lib/ruby/vendor_ruby/2.8.0/arm64-darwin20",
"/Users/watson/.rbenv/versions/2.8.0/lib/ruby/site_ruby/2.8.0/arm64-darwin20",
"/Users/watson/.rbenv/versions/2.8.0/lib/ruby/2.8.0/arm64-darwin20",
"sitearch"=>"arm64-darwin20",
"arch"=>"arm64-darwin20",
"/Users/watson/.rbenv/versions/2.8.0/include/arm64-darwin20",
"/Users/watson/.rbenv/versions/2.8.0/include/arm64-darwin20",
"/Users/watson/.rbenv/versions/2.8.0/lib/arm64-darwin20",
"sitearchlibdir"=>"/Users/watson/.rbenv/versions/2.8.0/lib/arm64-darwin20",
"archlibdir"=>"/Users/watson/.rbenv/versions/2.8.0/lib/arm64-darwin20",
"/Users/watson/.rbenv/versions/2.8.0/lib/arm64-darwin20",
"/Users/watson/.rbenv/versions/2.8.0/lib/arm64-darwin20",
"Target: arm64-apple-darwin20.0.0
"target_cpu"=>"arm64",
"target"=>"arm64-apple-darwin20",
"host_cpu"=>"arm",
"host"=>"arm-apple-darwin20.0.0",
"build_cpu"=>"arm",
"build"=>"arm-apple-darwin20.0.0",
"/Users/watson/.rbenv/versions/2.8.0/lib/ruby/2.8.0/arm64-darwin20",
"topdir"=>"/Users/watson/.rbenv/versions/2.8.0/lib/ruby/2.8.0/arm64-darwin20")

"target_cpu" indicates "arm64", however, "host_cpu" has "arm"...

I'd like to know whether this behavior is the right thing.

Related issues:
Related to Ruby master - Bug #3565: make test-all fails to run Closed 07/13/2010
Related to Ruby master - Bug #17605: Please backport Apple Silicon arm64 vs a... Closed

Associated revisions
Revision 43a9a974 - 11/21/2020 02:21 PM - nobu (Nobuyoshi Nakada)
[Bug #17021] Make host_* values consistent with target_*

Revision f03ba00f - 02/11/2021 04:52 AM - nagachika (Tomoyuki Chikanaga)
merge revision(s) 43a9a974e276dc49b03ec81ccace0adb534a3d20: [Backport #17021] [Backport #17605]

[Bug #17021] Make host_* values consistent with target_*

---
configure.ac | 7 +++++++-
1 file changed, 6 insertions(+), 1 deletion(-)
IMHO, "host_cpu" should have "arm64". "arm" looks like a bug to me. Would be nice to get this resolved, since it blocks https://github.com/ffi/ffi/pull/801.

This seems a clear bug, at least host_cpu and build_cpu. arm means 32-bit and is already used e.g. on Linux to mean that.

So it should be arm64 or aarch64.

target_cpu is already arm64 interestingly.

As I have no access to Apple silicon machines, can anyone share config.log file on that platform?

Maybe, uname -m (and tool/config.guess) returns "arm", then RUBY_UNIVERSAL_ARCH replaces target_cpu only?

I heard that config.guess claims it "arm", while uname -m reports "arm64".
First of all, $host_cpu and others are used only for cross-compilers and nonsense in the other cases. Why do they matter?

- Related to Bug #3565: make test-all fails to run added

- Related to Bug #17021: Make host_* values consistent with target_*

- Related to Bug #17605: Please backport Apple Silicon arm64 vs arm bugfix to 2.6 and 2.7 added