Ruby - Bug #19053

Build errors in the WebAssembly case

10/14/2022 12:39 PM - jaruga (Jun Aruga)

Status:	Closed		
Priority:	Normal		
Assignee:	katei (Yuta Saito)		
Target version:			
ruby -v:		**Backport:**	2.7: UNKNOWN, 3.0: UNKNOWN, 3.1: UNKNOWN

Description

I was testing WASI feature[1][2] on Ruby 3.2.0 preview2[3] on Fedora 36 following the instruction steps[2]. Then I got the build errors below when building both ruby-3.2.0-preview2.tar.gz and the latest master branch.

Could you tell me what's wrong? Thanks.

My environment is below.

```
$ uname -m
x86_64

$ which gcc
/bin/gcc

$ gcc --version
gcc (GCC) 12.1.1 20220507 (Red Hat 12.1.1-1)
Copyright (C) 2022 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

$ which ruby
/usr/local/ruby-3.2.0-preview2/bin/ruby

$ ruby -v
ruby 3.2.0preview2 (2022-09-09 master 35cfc9a3bb) [x86_64-linux]
```

Build from the ruby-3.2.0-preview2.tar.gz

I installed the wasi-sdk to the following directory, and did set the environment variable WASI_SDK_PATH by following the instruction steps on [2].

```
$ echo $WASI_SDK_PATH
/usr/local/wasi-sdk-16.0
```

I installed the binaryen by the RPM package.

```
$ sudo dnf install binaryen

$ rpm -q binaryen
binaryen-105-2.fc36.x86_64
```

Then tried to build with the source: ruby-3.2.0-preview2.tar.gz, and I failed to build.

```
$ pwd
/home/jaruga/src/ruby-3.2.0-preview2_wasi

$ ./autogen.sh

$ ./configure LDFLAGS="-Xlinker -zstack-size=16777216" \
  --host wasm32-unknown-wasi \
  --with-destdir=./ruby-wasm32-wasi \
  --with-static-linked-ext \
```

```
  --with-ext=ripper,monitor
checking for ruby... /usr/local/ruby-3.2.0-preview2/bin/ruby
tool/config.guess already exists
tool/config.sub already exists
checking build system type... x86_64-pc-linux-gnu
checking host system type... wasm32-unknown-wasi
checking target system type... wasm32-unknown-wasi
checking for wasm32-unknown-wasi-wasm-opt... no
checking for wasm-opt... wasm-opt
checking wheather $WASI_SDK_PATH is set... yes
checking for wasm32-unknown-wasi-gcc... /usr/local/wasi-sdk-16.0/bin/clang
checking for /usr/local/wasi-sdk-16.0/bin/llvm-ar... /usr/local/wasi-sdk-16.0/bin/llvm-ar
checking for /usr/local/wasi-sdk-16.0/bin/clang++... no
checking for /usr/local/wasi-sdk-16.0/bin/llvm-nm... no
checking for /usr/local/wasi-sdk-16.0/bin/llvm-objcopy... no
checking for /usr/local/wasi-sdk-16.0/bin/llvm-objdump... no
checking for /usr/local/wasi-sdk-16.0/bin/llvm-ranlib... /usr/local/wasi-sdk-16.0/bin/llvm-ranlib
checking for /usr/local/wasi-sdk-16.0/bin/llvm-strip... no
checking for wasm32-unknown-wasi-gcc... (cached) /usr/local/wasi-sdk-16.0/bin/clang
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... yes
checking for suffix of object files... o
checking whether the compiler supports GNU C... yes
checking whether /usr/local/wasi-sdk-16.0/bin/clang accepts -g... yes
checking for /usr/local/wasi-sdk-16.0/bin/clang option to enable C11 features... none needed
checking for wasm32-unknown-wasi-g++... no
checking for wasm32-unknown-wasi-c++... no
checking for wasm32-unknown-wasi-gpp... no
checking for wasm32-unknown-wasi-aCC... no
checking for wasm32-unknown-wasi-CC... no
checking for wasm32-unknown-wasi-cxx... no
checking for wasm32-unknown-wasi-cc++... no
checking for wasm32-unknown-wasi-cl.exe... no
checking for wasm32-unknown-wasi-FCC... no
checking for wasm32-unknown-wasi-KCC... no
checking for wasm32-unknown-wasi-RCC... no
checking for wasm32-unknown-wasi-xlC_r... no
checking for wasm32-unknown-wasi-xlC... no
checking for wasm32-unknown-wasi-clang++... no
checking for g++... g++
configure: WARNING: using cross tools not prefixed with host triplet
checking whether the compiler supports GNU C++... yes
checking whether g++ accepts -g... yes
checking for g++ option to enable C++11 features... none needed
checking how to run the C preprocessor... /usr/local/wasi-sdk-16.0/bin/clang -E
checking for wasm32-unknown-wasi-ranlib... (cached) /usr/local/wasi-sdk-16.0/bin/llvm-ranlib
checking for wasm32-unknown-wasi-gar... (cached) /usr/local/wasi-sdk-16.0/bin/llvm-ar
checking for wasm32-unknown-wasi-gas... no
checking for wasm32-unknown-wasi-as... no
checking for gas... no
checking for as... as
checking for wasm32-unknown-wasi-gld... /usr/local/wasi-sdk-16.0/bin/clang
checking for wasm32-unknown-wasi-gnm... no
checking for wasm32-unknown-wasi-nm... no
checking for gnm... no
checking for nm... nm
checking for wasm32-unknown-wasi-gobjcopy... no
checking for wasm32-unknown-wasi-objcopy... no
checking for gobjcopy... no
checking for objcopy... objcopy
checking for wasm32-unknown-wasi-gobjdump... no
checking for wasm32-unknown-wasi-objdump... no
checking for gobjdump... no
checking for objdump... objdump
checking for wasm32-unknown-wasi-gstrip... no
```

```
checking for wasm32-unknown-wasi-strip... no
checking for gstrip... no
checking for strip... strip
checking for stdio.h... yes
checking for stdlib.h... yes
checking for string.h... yes
checking for inttypes.h... yes
checking for stdint.h... yes
checking for strings.h... yes
checking for sys/stat.h... yes
checking for sys/types.h... yes
checking for unistd.h... yes
checking for wchar.h... yes
checking for minix/config.h... no
checking for vfork.h... no
checking whether it is safe to define __EXTENSIONS__... yes
checking whether _XOPEN_SOURCE should be defined... no
/usr/local/wasi-sdk-16.0/bin/clang: /lib64/libtinfo.so.6: no version information available (requir
ed by /usr/local/wasi-sdk-16.0/bin/clang)
/usr/local/wasi-sdk-16.0/bin/clang: /lib64/libtinfo.so.6: no version information available (requir
ed by /usr/local/wasi-sdk-16.0/bin/clang)
/usr/local/wasi-sdk-16.0/bin/clang: /lib64/libtinfo.so.6: no version information available (requir
ed by /usr/local/wasi-sdk-16.0/bin/clang)
checking whether the linker is GNU ld... no
checking whether /usr/local/wasi-sdk-16.0/bin/clang -E accepts -o... yes
checking for /usr/local/wasi-sdk-16.0/bin/llvm-ar flags... rcD
checking whether ln -s works... yes
checking whether make sets $(MAKE)... yes
checking for a BSD-compatible install... /bin/install -c
checking for a race-free mkdir -p... /bin/mkdir -p
checking for wasm32-unknown-wasi-dtrace... no
checking for dot... dot
checking for doxygen... no
checking for wasm32-unknown-wasi-pkg-config... no
checking for pkg-config... pkg-config
checking whether it is Android... no
checking for cd using physical directory... cd -P
checking whether CFLAGS is valid... no
configure: error: something wrong with CFLAGS=" "

$ echo $?
1
```

There is the /lib64/libtinfo.so.6.

```
$ ls -l /lib64/libtinfo.so.6
lrwxrwxrwx. 1 root root 15 Jan 20  2022 /lib64/libtinfo.so.6 -> libtinfo.so.6.2*
$ rpm -qf /lib64/libtinfo.so.6
ncurses-libs-6.2-9.20210508.fc36.x86_64
```

Build on the master branch

On the master branch 5ccb625fbbd1e774636a9fdbe0bf1c3d38e085d5, I also failed to build with the error below.

```
$ pwd
/home/jaruga/git/ruby/ruby

$ ./autogen.sh

$ ./configure LDFLAGS="-Xlinker -zstack-size=16777216" \
  --host wasm32-unknown-wasi \
  --with-destdir=./ruby-wasm32-wasi \
  --with-static-linked-ext \
  --with-ext=ripper,monitor
checking for ruby... /usr/local/ruby-3.2.0-preview2/bin/ruby
tool/config.guess already exists
```

```
tool/config.sub already exists
checking build system type... x86_64-pc-linux-gnu
checking host system type... Invalid configuration `wasm32-unknown-wasi': system `wasi' not recogn
ized
configure: error: /bin/sh ./tool/config.sub wasm32-unknown-wasi failed
```

References

- [1] https://bugs.ruby-lang.org/issues/18462
- [2] https://github.com/ruby/ruby/pull/5407
- [3] https://www.ruby-lang.org/en/news/2022/09/09/ruby-3-2-0-preview2-released/

History

#1 - 10/14/2022 05:40 PM - katei (Yuta Saito)

- Assignee set to katei (Yuta Saito)

I'm not familiar with Fedora environment, but it looks like the pre-built wasi-sdk distributed in GitHub Releases is incompatible with Fedora.

The pre-built SDK is built on Ubuntu, and it expects the build machine to have libtinfo.so.6 with proper .gnu.version_d section.

As far as I've checked, libtinfo.so.6 in fedora:36 Docker image doesn't have the version definition section, so the pre-built SDK is incompatible with the environment, unfortunately.

Therefore, you need to build the SDK by yourself on Fedora, then you will be able to build for WASI target.

For Invalid configuration wasm32-unknown-wasi': system wasi' not recognized error, you need to download the latest config.guess following the README.md build steps

Thanks.

#2 - 10/20/2022 02:30 PM - jaruga (Jun Aruga)

- File Dockerfile_ubuntu added

- File Dockerfile_fedora added

> I'm not familiar with Fedora environment, but it looks like the pre-built wasi-sdk distributed in GitHub Releases is incompatible with Fedora.
>
> The pre-built SDK is built on Ubuntu, and it expects the build machine to have libtinfo.so.6 with proper .gnu.version_d section.
>
> As far as I've checked, libtinfo.so.6 in fedora:36 Docker image doesn't have the version definition section, so the pre-built SDK is incompatible with the environment, unfortunately.

Thanks for checking it! Let me check the things you mentioned step by step.

First, I checked the .gnu.version_d section for both Ubuntu 20.04 (focal = ubuntu-latest on the SDK's GitHub Actions) and Fedora 36 by using the command below with the 2 Dockerfile files. I attached the files. And I confirmed that Ubuntu does have the section, but Fedora doesn't have it.

```
$ objdump -s -j .gnu.version_d /path/to/libtinfo.so.6.2

$ docker build --rm -t test-libtifo-ubuntu -f Dockerfile_ubuntu .

$ docker run --rm -t test-libtifo-ubuntu | grep .gnu.version_d
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.
Contents of section .gnu.version_d:

$ docker build --rm -t test-libtifo-fedora -f Dockerfile_fedora .

$ docker run --rm -t test-libtifo-fedora
Emulate Docker CLI using podman. Create /etc/containers/nodocker to quiet msg.

/usr/lib64/libtinfo.so.6.2:     file format elf64-x86-64

objdump: section '.gnu.version_d' mentioned in a -j option, but not found in any input file
```

I will check other things that you mentioned, and I will share it.

#3 - 10/20/2022 04:37 PM - jaruga (Jun Aruga)

> Therefore, you need to build the SDK by yourself on Fedora, then you will be able to build for WASI target.

I built the SDK from the source and installed it on the Fedora, and I was able to build the Ruby by following the instructions - Ruby wasm README, and run the Ruby on the wasmtime command.

```
$ pwd
/home/jaruga/src/ruby-3.2.0-preview2_fedora

$ echo $WASI_SDK_PATH
/usr/local/wasi-sdk-16.0-fedora

$ ./autogen.sh
$ ./configure LDFLAGS="-Xlinker -zstack-size=16777216" \
  -prefix=/usr/local/ruby-3.2.0-preview2-wasm32-wasi \
  --host wasm32-unknown-wasi \
  --with-destdir=./ruby-wasm32-wasi \
  --with-static-linked-ext \
  --with-ext=ripper,monitor
...
Configuration summary for ruby version 3.2.0

   * Installation prefix: /usr/local/ruby-3.2.0-preview2-wasm32-wasi
   * exec prefix:         ${prefix}
   * arch:                wasm32-wasi
   * site arch:           ${arch}
   * RUBY_BASE_NAME:      ruby
   * ruby lib prefix:     ${libdir}/${RUBY_BASE_NAME}
   * site libraries path: ${rubylibprefix}/${sitearch}
   * vendor path:         ${rubylibprefix}/vendor_ruby
   * target OS:           wasi
   * compiler:            $(CC_WRAPPER) /usr/local/wasi-sdk-16.0-fedora/bin/clang
   * with thread:         none
   * with coroutine:      asyncify
   * enable shared libs:  no
   * dynamic library ext: so
   * CFLAGS:              -fdeclspec ${optflags} ${debugflags} ${warnflags}
   * LDFLAGS:             -L. -Xlinker -zstack-size=16777216
   * DLDFLAGS:            -Xlinker -zstack-size=16777216
   * optflags:            -O3 -fno-fast-math
   * debugflags:          -ggdb3
   * warnflags:           -Wall -Wextra -Wextra-tokens -Wdeprecated-declarations \
                         -Wdivision-by-zero -Wdiv-by-zero \
                         -Wimplicit-function-declaration -Wimplicit-int \
                         -Wmisleading-indentation -Wpointer-arith -Wshorten-64-to-32 \
                         -Wwrite-strings -Wold-style-definition -Wmissing-noreturn \
                         -Wno-cast-function-type -Wno-constant-logical-operand \
                         -Wno-long-long -Wno-missing-field-initializers \
                         -Wno-overlength-strings -Wno-parentheses-equality \
                         -Wno-self-assign -Wno-tautological-compare \
                         -Wno-unused-parameter -Wno-unused-value -Wunused-variable \
                         -Wundef
   * strip command:      strip
   * install doc:        rdoc
   * MJIT support:       no
   * YJIT support:       no
   * man page type:      doc
   * static-linked-ext:  yes
   * BASERUBY -v:        ruby 3.2.0preview2 (2022-09-09 master 35cfc9a3bb) \
                         [x86_64-linux]

$ make
$ sudo make install
```

I didn't need to set the sudo for the sudo make install in this case, as the --with-destdir=./ruby-wasm32-wasi was specified as configure option.

```
$ which wasmtime
~/.wasmtime/bin/wasmtime

$ wasmtime --version
wasmtime-cli 1.0.1
```

```
$ wasmtime ruby-wasm32-wasi/usr/local/ruby-3.2.0-preview2-wasm32-wasi/bin/ruby --mapdir /::./ruby-wasm32-wasi/
 -- -e 'puts RUBY_PLATFORM'
wasm32-wasi
```

#4 - 10/20/2022 04:49 PM - jaruga (Jun Aruga)

- Status changed from Open to Closed

> For Invalid configuration wasm32-unknown-wasi': system wasi' not recognized error, you need to download the latest config.guess following the README.md build steps.

I was able to run the configure after running the ruby tool/downloader.rb ... following the steps document.

```
$ ruby tool/downloader.rb -d tool -e gnu config.guess config.sub
downloading config.guess ... done
downloading config.sub ... done

$ ./autogen.sh

$ ./configure LDFLAGS="-Xlinker -zstack-size=16777216" \
  --host wasm32-unknown-wasi \
  --with-destdir=./ruby-wasm32-wasi \
  --with-static-linked-ext \
  --with-ext=ripper,monitor
```

I think I can close this ticket. Thanks for your help!

#5 - 10/21/2022 03:24 PM - jaruga (Jun Aruga)

> The pre-built SDK is built on Ubuntu, and it expects the build machine to have libtinfo.so.6 with proper .gnu.version_d section.

I was told in the Fedora project that this issue is a known issue on the Fedora ncurses RPM package including the libtinfo.so.6 . The issue ticket is below.

ncurses: Please enable symbol versioning
https://bugzilla.redhat.com/show_bug.cgi?id=1875587

The situation is that the ncurses Debian package to build the ncurses by the configure with --with-versioned-syms. But the ncurses Fedora RPM package doesn't use the the --with-versioned-syms option. Because the upstream ncurses project doesn't use the option as their build options.

Files

Dockerfile_fedora	196 Bytes	10/20/2022	jaruga (Jun Aruga)
Dockerfile_ubuntu	434 Bytes	10/20/2022	jaruga (Jun Aruga)