configure prints a warning when cross-compiling

01/21/2022 08:12 AM - mame (Yusuke Endoh)

$ sudo apt-get install gcc-aarch64-linux-gnu
$ ./configure --host=aarch64-linux-gnu
downloading config.guess ... done
downloading config.sub ... done
checking build system type... x86_64-pc-linux-gnu
checking host system type... aarch64-unknown-linux-gnu
checking target system type... aarch64-unknown-linux-gnu
checking for aarch64-linux-gnu-gcc... aarch64-linux-gnu-gcc
checking for aarch64-linux-gnu-aarch64-linux-gnu-ld... no
checking for aarch64-linux-gnu-ld... aarch64-linux-gnu-ld
configure: WARNING: using cross tools not prefixed with host triplet

It attempts to find aarch64-linux-gnu-aarch64-linux-gnu-ld, which fails. Then it finds aarch64-linux-gnu-ld, but also it prints "WARNING: using cross tools not prefixed with host triplet".

I guess this was triggered by 2c96e04868477eaa1420945d57bf5b3adb521e84. shyouhei (Shyouhei Urabe) Could you take a look?

Associated revisions
Revision 4113862c - 02/17/2022 01:47 PM - nobu (Nobuyoshi Nakada)
Do not search for commands with double tool prefixes [Bug #18504]

The CC found by AC_CHECK_TOOL is prefixed by the host triplet when cross compiling. To search for commands with AC_CHECK_TOOL based on that CC means to search also doubly prefixed names.

History
#1 - 01/21/2022 08:50 AM - shyouhei (Shyouhei Urabe)
I know what is going on. About the only thing we can do is to settle for that warning. It works nonetheless.

The problem is our --with-gcc configure option. That way you can instruct which compiler to use. Then we have to find the right linker that understands proper link-time optimisation scheme that the user-input compiler employs.

In case of cross compilation autoconf also sets $CC. But from our point of view there is no way to tell if that aarch64-linux-gnu-gcc value was user-input or auto-generated. We have to do the same thing (find the linker that comes with the compiler), which eventually succeeds, with a warning.

In order to suppress the warning we have to detect where the $CC value came from; which is impractical. Please just live with the annoyance, or give up --with-gcc.

#2 - 02/15/2022 11:22 AM - nobu (Nobuyoshi Nakada)
What about this?

diff --git a/configure.ac b/configure.ac
index 4feefb57915..01e87c0a9b4 100644
--- a/configure.ac
+++ b/configure.ac
@@ -129,9 +129,9 @@ AC_ARG_WITH(gcc,
             AS_ADMINSTRING([--without-gcc], [never use gcc]),
             [AS_ADMINSTRING([--without-gcc], [never use gcc]),
               [AS_ADMINSTRING([--without-gcc], [never use gcc]),

03/16/2022 1/2
dnl If the user switches compilers, we can't believe the cache
AS_IF([test ! -z "$ac_cv_prog_CC" -a ! -z "$CC" -a "$CC" != "$ac_cv_prog_CC"], [AC_MSG_ERROR(cached CC is different -- throw away $cache_file
@@ -143,16 +143,16 @@ RUBY_WASMTOOLS
AS_CASE(["${build_os}"],
    [linux*|cygwin*|msys*], [    
        # Naruse prefers GCC on Linux
-       AC_CHECK_TOOLS([CC], [gcc clang cc])
+       AC_CHECK_TOOLS([CC], [${with_gcc} gcc clang cc])
    ], [solaris*], [
        # Clang on Solaris is largely untested.
        # https://bugs.ruby-lang.org/issues/17949
-       AC_CHECK_TOOLS([CC], [cc gcc])
+       AC_CHECK_TOOLS([CC], [${with_gcc} cc gcc])
    ], [
        # OpenBSD wants to prefer cc over gcc.
        # See https://github.com/ruby/ruby/pull/2443
-       AC_CHECK_TOOLS([CC], [cl.exe clang cc gcc c99 /usr/ucb/cc])
+       AC_CHECK_TOOLS([CC], [${with_gcc} cl.exe clang cc gcc c99 /usr/ucb/cc])
    ]
    AC_ARG_VAR([AR], [Archiver command])

#3 - 02/17/2022 01:47 PM - nobu (Nobuyoshi Nakada)
- Status changed from Assigned to Closed

Applied in changeset git|4113862c0068a8a95d752f5fd1f4980f92cd41d7.

Do not search for commands with double tool prefixes [Bug #18504]

The CC found by AC_CHECK_TOOL is prefixed by the host triplet when cross compiling. To search for commands with AC_CHECK_TOOL based on that CC means to search also doubly prefixed names.