Ubuntu to Ubuntu Cross C++ Development

Pieter P

Work in progress

In the meantime, you might find some answers in the guide about Ubuntu to Raspberry Pi OS Cross C++ Development.

In short

Brief instructions for Ubuntu 21.10 (Impish Indri), 64-bit:

```
# Install the necessary tools:
sudo apt install ubuntu-dev-tools
# Then create a schroot environment for the arm64 environment, using Ubuntu
# Impish (remember the name you used, you'll need it later):
mk-sbuild --arch=arm64 --skip-proposed --skip-updates --skip-security --name=rpi3-impish impish
# If this is the first time you run mk-sbuild, you'll have to follow some
# instructions. Since we won't be publishing any software, you can simply accept
# the default configuration. Afterwards, reboot or log out and back in again, as
# instructed. Alternatively, use
su - $USER # to flush group membership.
# Then run the command again, this time it will actually create the schroot:
mk-sbuild --arch=arm64 --skip-proposed --skip-updates --skip-security --name=rpi3-impish impish
# Install some dependencies in the schroot. Use the sbuild-apt wrapper around
# the apt-get tool, and give it the name of the schroot you created earlier,
# with the architecture as suffix:
sudo sbuild-apt rpi3-impish-arm64 apt-get install libboost-all-dev
# Download and extract the cross-compilation toolchain:
wget -qO- https://github.com/tttapa/docker-arm-cross-toolchain/releases/latest/download/x-tools-aarch64-rpi3-linux-gnu.tar.bz2
| tar xJ -C ~/opt
# Add the toolchain to your path:
export PATH="$HOME/opt/x-tools/aarch64-rpi3-linux-gnu/bin:$PATH"
# Download the repository with the example CMake C++ project:
git clone https://github.com/tttapa/RPi-Cross-Cpp-Development.git
# Enter it:
cd RPi-Cross-Cpp-Development
# Replace the schroot name in the toolchain file by the name used earlier:
sed -i 's/schroot-name-arm64/rpi3-impish-arm64/' cmake/aarch64-rpi3-linux-gnu.cmake
# Configure the project for cross-compilation:
cmake -S. -Bbuild -DCMAKE_TOOLCHAIN_FILE="$PWD/cmake/aarch64-rpi3-linux-gnu.cmake"
# Build the project:
cmake --build build -j
# Install the project into the staging area:
cmake --install build
# Install the dependencies on the Raspberry Pi:
ssh RPi3 sudo apt-get install -y libboost-program-options1.74.0
# Copy the “Hello World” example program to the Raspberry Pi:
ssh RPi3 mkdir -p '~/.local/bin'
scp ~/RPi-dev/staging-aarch64-rpi3/bin/hello RPi3:~/.local/bin
# Run the example program:
ssh RPi3 bash --login -c hello
# Or simply log in using
ssh RPi3
# and then run the program there:
hello
```