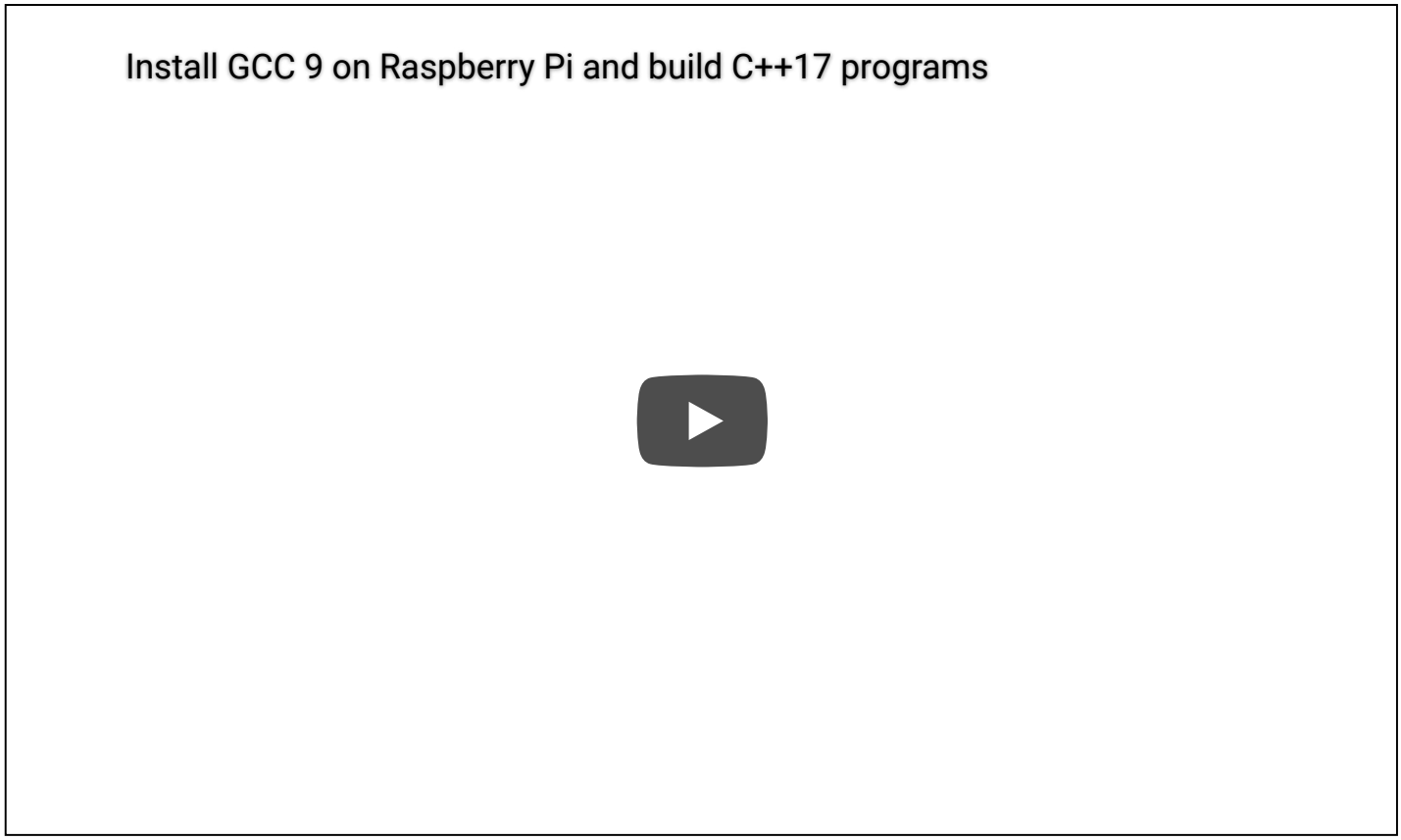
Raspberry Pi - Install GCC 9 and compile C++17 programs

Posted on December 8, 2017 by Paul

Updated 28 June 2019

In this article I will show you how to install GCC 9 on your Raspberry Pi system and how to compile C++17 programs. At the time of this writing Raspbian is based on Debian Buster, which comes with the stable but slightly outdated *GCC 8.3* as the default C and C++ compiler.

There is also a video version of this tutorial:



If you want to compile *GCC 9* from sources check my article.

If also you want to install *Clang 8* on your Raspberry Pi, check my article.

First, make sure that your Raspbian is updated:

```
sudo apt update && sudo apt upgrade -y
```

If you don't have *git* on your Raspbian, you can install it with:

```
sudo apt install git
```

Let's start the *GCC* installation process. Open a Terminal and download a binary of *GCC 9*:

```
git clone https://bitbucket.org/sol_prog/raspberry-pi-gcc-binary.git
```

Next, extract the archive, move the extracted compilers to */opt* and remove the repository:

```
cd raspberry-pi-gcc-binary
tar -xjvf gcc-9.1.0-armhf-raspbian.tar.bz2
sudo mv gcc-9.1.0 /opt
cd ..
rm -rf raspberry-pi-gcc-binary
```

Next, we are going to add the new compilers to the path and create a few symbolic links:

```
echo 'export PATH=/opt/gcc-9.1.0/bin:$PATH' >> ~/.bashrc
echo 'export LD_LIBRARY_PATH=/opt/gcc-9.1.0/lib:$LD_LIBRARY_PATH' >> ~/.bashrc
. ~/.bashrc
sudo ln -s /usr/include/arm-linux-gnueabihf/sys /usr/include/sys
sudo ln -s /usr/include/arm-linux-gnueabihf/bits /usr/include/bits
sudo ln -s /usr/include/arm-linux-gnueabihf/gnu /usr/include/gnu
sudo ln -s /usr/include/arm-linux-gnueabihf/asm /usr/include/asm
sudo ln -s /usr/lib/arm-linux-gnueabihf/crti.o /usr/lib/crti.o
sudo ln -s /usr/lib/arm-linux-gnueabihf/crt1.o /usr/lib/crt1.o
sudo ln -s /usr/lib/arm-linux-gnueabihf/crtn.o /usr/lib/crtn.o
```

At this point, you should be able to invoke the compilers with *gcc-9.1*, *g++-9.1* or *gfortran-9.1*.

You can check if everything is properly setup by printing the version of the installed compiler:

```
gcc-9.1 --version
```

This is what I see on my Pi:

```
pi@raspberrypi:~ $ gcc-9.1 --version
gcc-9.1 (GCC) 9.1.0
Copyright (C) 2019 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.

pi@raspberrypi:~ $
```

If, at some point in the future, you'll want to get rid of *GCC 9* from your system, all you have to do is to remove the *gcc-9.1.0* folder from */opt*, example:

```
sudo rm -rf /opt/gcc-9.1.0
```

The above procedure will keep *GCC 8.3* as the default C and C++ compiler for any package that depends on it. If you want to compile C programs you could use *gcc-9.1* and for C++ *g++-9.1*.

Let's try to compile and run a C++17 code that uses an if block with init-statement (the example is a bit silly, but it will show you how to compile C++17 programs):

```
#include <iostream>

int main() {
    // if block with init-statement:
    if(int a = 5; a < 8) {
        std::cout << "Local variable a is < 8\n";
    } else {
        std::cout << "Local variable a is >= 8\n";
    }
    return 0;
}
```

Save the above code in a file named *if_test.cpp* and compile it with:

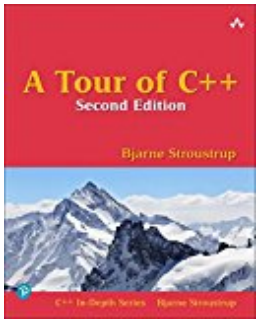
```
g++-9.1 -std=c++17 -Wall -pedantic if_test.cpp -o if_test
```

This is what I see on my Pi:

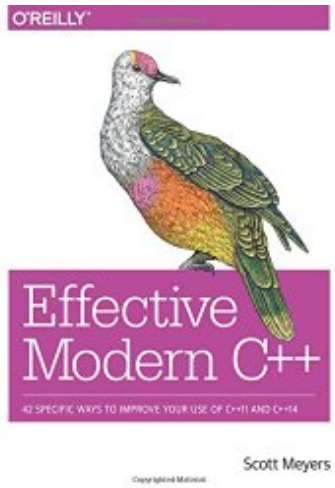
```
pi@raspberrypi:~ $ g++-9.1 -std=c++17 -Wall -pedantic if_test.cpp -o if_test
pi@raspberrypi:~ $ ./if_test
Local variable a is < 8
pi@raspberrypi:~ $
```

For an overview of C++17 support in GCC see https://gcc.gnu.org/projects/cxx-status.html.

If you are interested to learn more about modern C++ I would recommend reading A Tour of C++ by Bjarne Stroustrup.



or Effective Modern C++ by Scott Meyers.



Show Comments

Categories

C# Charts C++11 Regex Scheme Multithreading Posix Books C++ C++14 C++17 OSX Python Objective-C Windows Clang Fortran CUDA Roguelike Perlin Cling C++20 Linux WSL Fractals OpenGL JavaScript OpenCV BeagleBone Productivity Raspberry Pi OpenMP iOS Node.js macOS NumPy SciPy Matplotlib GCC Swift C C99 C11 Arduino Videos Armadillo Chromebook ChromeOS Docker x86-64 F# dotnet C17 Lisp

Recent posts

- Install GCC 9 on Windows - Build C, C++ and Fortran programs
- C++20 span tutorial
- A Happy Halloween from C++ and an IIFE lambda
- C++17 - find the greatest common divisor, gcd, of two or more integers
- Install OpenCV 4 with Python 3 on macOS Catalina
- Compiling GCC 9 on macOS Catalina
- Install OpenCV 4 on Raspberry Pi for C++ and Python development
- Building SBCL - Steel Bank Common Lisp on Windows
- Cross compiling OpenCV 4 for Raspberry Pi Zero
- Python - using C and C++ libraries with ctypes

solarianprogrammer.com does not collect any personal information about its visitors except that which they provide voluntarily when leaving comments. This information will never be disclosed to any third party for any purpose. Some of the links contained within this site have my referral id, which provides me with a small commission for each sale. Thank you for understanding.