Instantly share code, notes, and snippets.

nchaigne / **build-gcc-9.2.0-on-centos7.md**

Last active 2 months ago

Embed ▾ `<script src="https://gi` Download ZIP

Building GCC 9.2.0 on CentOS 7

`build-gcc-9.2.0-on-centos7.md`

Building GCC 9.2.0 on CentOS 7

Introduction

CentOS 7 distribution (as well as RHEL 7) ships with a somewhat outdated version of the GCC compiler (4.8.5 on CentOS 7.5), which may not be suitable to your compilation requirements. For example, C11 - which supersedes C99 - is fully supported only starting from GCC 4.9).

Additionally, recent versions of GCC (GCC6, GCC7, GCC8, GCC9) come with improvements which help detect issues at build time and offer suggestions on how to fix them. Sometimes, these are even actually helpful!

This note describes how to build the latest GCC (9.2.0 as of October 2019) from sources on CentOS 7. This should be applicable as is on RHEL 7. For other Linux distributions, adapt as needed.

While this is not overly complicated, building GCC takes quite some time. So you might want to plan to do something else while it builds... a coffee break just won't make it.

Prerequisites

Prerequisites are described here: https://gcc.gnu.org/install/prerequisites.html

- C++ compiler

> **`yum install gcc gcc-c++`**

Required support libraries, listed hereafter, can be downloaded automatically using script `download_prerequisites` included in the GCC archive. It's convenient, so we'll do that.

- GNU Multiple Precision Library (GMP) version 4.3.2 (or later)
- MPFR Library version 2.4.2 (or later)
- MPC Library version 0.8.1 (or later)
- ISL library version 0.15 (or later)

Build and install gcc

```
cd /home/build
GCC_VERSION=9.2.0
wget https://ftp.gnu.org/gnu/gcc/gcc-${GCC_VERSION}/gcc-${GCC_VERSION}.tar.gz
tar xzvf gcc-${GCC_VERSION}.tar.gz
mkdir obj.gcc-${GCC_VERSION}
cd gcc-${GCC_VERSION}
./contrib/download_prerequisites
cd ../obj.gcc-${GCC_VERSION}
../gcc-${GCC_VERSION}/configure --disable-multilib --enable-languages=c,c++
make -j $(nproc)
make install
```

Notes:

- If you have several processors available, you can benefit from a parallel build. For example, `make -j 6` will use 6 CPUs. (You might want to save a few for yourself, so you can do things on your server while gcc builds.)
- Make sure you have enough space in `/home/build` (or whatever location you choose). You will need ~1 GB for gcc sources, ~6 GB for the build). Be prepared.
- This will install gcc in `/usr/local/bin/gcc` (default prefix is `/usr/local`). Your distro gcc (`/usr/bin/gcc`) will not be overwritten, but if later on you need to invoke it, you will have to do so explicitly. Configure with option `--prefix` if you want to change this.
- Option `--disable-multilib` prevents building multiple target libraries (I don't need them, and it is simpler).
- Option `--enable-langagues` allows to have a leaner and faster build if you only need (for example) C and C++.
- See GCC documentation for the full list of configure options.

Libruh commented on Feb 3

gzip: stdin: unexpected end of file

cyrusbehr commented on Feb 25

Great guide.
To enable gcc 9 as default, may have to run the following commands:

```
sudo yum install devtoolset-9-toolchain
scl enable devtoolset-9 bash
```

kalyanmysore commented on Mar 2

This is a very helpful guide. Helped me install gcc 9.2 on an Amazon Linux AMI based EC2 instance.
Would recommend using this at the start:
screen -S gcc
This way, if you get ssh disconnected in the middle, you can ssh back in and then resume using
screen -r gcc