There are three distinct steps in this process:

+

**1. Configure the software**

+

The configure script is responsible for getting ready to build the software on your specific system. It makes sure all of the dependencies for the rest of the build and install process are available, and finds out whatever it needs to know to use those dependencies.

+

Unix programs are often written in C, so we’ll usually need a C compiler to build them. In these cases the configure script will establish that your system does indeed have a C compiler, and find out what it’s called and where to find it.

+

**2. Build the software**

+

Once configure has done its job, we can invoke make to build the software. This runs a series of tasks defined in a Makefile to build the finished program from its source code.

+

The tarball you download usually doesn’t include a finished Makefile. Instead it comes with a template called Makefile.in and the configure script produces a customisedMakefile specific to your system.

+

**3. Install the software**

+

Now that the software is built and ready to run, the files can be copied to their final destinations. The make install command will copy the built program, and its libraries and documentation, to the correct locations.

+

This usually means that the program’s binary will be copied to a directory on your PATH, the program’s manual page will be copied to a directory on your MANPATH, and any other files it depends on will be safely stored in the appropriate place.

+

Since the install step is also defined in the Makefile, where the software is installed can change based on options passed to the configure script, or things the configure script discovered about your system.

+

Depending on where the software is being installed, you might need escalated permissions for this step so you can copy files to system directories. Using sudo will often do the trick.