**GCC-4.1.2**

**Introduction to GCC**

The GCC package contains GNU compilers. This package is useful for compiling programs written in C, C++, Fortran, Java, Objective C and Ada. Also included is GNU Treelang. Treelang is a sample language, useful only to help people understand how to implement a new language front end to GCC. It is not a useful language in itself other than as an example or basis for building a new language. Therefore only language developers are likely to have an interest in it.

[Note]

**Note**

The Fortran compiler included with the GCC-4.x package now aims to be conformant with the Fortran 95 standard, not the Fortran 77 standard as all previous versions of GCC have been. Please note the following paragraph copied directly from the GCC-4.1.2 **gfortran** man page.

“Gfortran is not yet a fully conformant Fortran 95 compiler. It can generate code for most constructs and expressions, but work remains to be done. In particular, there are known deficiencies with ENTRY, NAMELIST, and sophisticated use of MODULES, POINTERS and DERIVED TYPES. For those whose Fortran codes conform to either the Fortran 77 standard or the GNU Fortran 77 language, we recommend to use **g77** from GCC 3.4.x”

Instructions to install the 3.4.6 version of the Fortran compiler can be found in [GCC-3.3.6](https://www.linuxfromscratch.org/blfs/view/6.3/general/gcc3.html) and on the [BLFS Wiki](http://wiki.linuxfromscratch.org/blfs/wiki/gcc3).

**Package Information**

* Download (HTTP): <http://ftp.gnu.org/gnu/gcc/gcc-4.1.2/gcc-4.1.2.tar.bz2>
* Download (FTP): [ftp://ftp.gnu.org/gnu/gcc/gcc-4.1.2/gcc-4.1.2.tar.bz2](https://www.linuxfromscratch.org/blfs/view/6.3/general/ftp:/ftp.gnu.org/gnu/gcc/gcc-4.1.2/gcc-4.1.2.tar.bz2)
* Download MD5 sum: a4a3eb15c96030906d8494959eeda23c
* Download size: 39.7 MB
* Estimated disk space required: 1.9 GB
* Estimated build time: 77 SBU (build, test and install all compilers)

**GCC Dependencies**

**Recommended**

[DejaGnu-1.4.4](https://www.linuxfromscratch.org/blfs/view/6.3/general/dejagnu.html)

**Optional (Required to Build the Fortran Compiler)**

[GMP-4.2.2](https://www.linuxfromscratch.org/blfs/view/6.3/general/gmp.html) and [MPFR](http://www.mpfr.org/)

[Note]

**Note**

If you plan to compile Ada, you will need to install GNAT temporarily to satisfy the circular dependency when you recompile GCC to include Ada. The download location shown below requires that you create an account on the AdaCore site before you can download the package. Nothing is required other than a valid email address.

**Ensure you download the 2005 version of the GNAT compiler. The 2006 version will not work. The file name you need is: gnat-gpl-2005-i686-gnu-linux-gnu-libc2.3-bin.tar.gz.**

**GNAT GPL 2005 Package Information**

* Download (HTTP): <https://libre.adacore.com/dynamic/download_page>
* Download MD5 sum: 659b7f57e00b73ecaa4863dabea5e0e2
* Download size: 44.6 MB

User Notes: <http://wiki.linuxfromscratch.org/blfs/wiki/gcc>

**Installation of GNAT**

Install GNAT by running the following command as the root user:

make ins-all prefix=/opt/gnat

The GNAT compiler can be invoked by executing the **gcc** binary installed in /opt/gnat/bin.

You may now remove the GNAT source directory:

cd .. &&

rm -rf gnat-gpl-2005-i686-gnu-linux-gnu-libc2.3-bin

Prepare to compile GCC by placing the GNAT version of **gcc** at the beginning of the PATH variable by using the following commands:

PATH\_HOLD=$PATH &&

export PATH=/opt/gnat/bin:$PATH\_HOLD

**Installation of GCC**

Install GCC by running the following commands:

[Important]

**Important**

The installation process may overwrite your existing GCC **gcc** and **c++** compilers and libraries. It is highly recommended that you have the Tcl, Expect and DejaGnu packages installed before beginning the build so you can run the full suite of tests.

Do not continue with the **make install** command until you are confident the build was successful. You can compare your test results with those found at <http://gcc.gnu.org/ml/gcc-testresults/>. There's also an i686 platform test result produced by an LFS-SVN-20070304 system at <http://anduin.linuxfromscratch.org/files/BLFS/6.3/gcc412_test.txt>. You may also want to refer to the information found in the GCC section of Chapter 6 in the LFS book ([../../../../lfs/view/6.3/chapter06/gcc.html](https://www.linuxfromscratch.org/lfs/view/6.3/chapter06/gcc.html)).

The instructions below perform a “**make bootstrap**” instead of just a plain “**make**” intentionally. Even though it is assumed that the current version of GCC is installed in LFS, because this installation includes the Ada compiler as an installed language, a bootstrap is required. The GNAT compiler must be used for stage1 of the bootstrap in order to build Ada. If the process didn't use a “bootstrap” you could end up having a **gcc** installed on the system that was produced by a foriegn compiler.

sed -i 's/install\_to\_$(INSTALL\_DEST) //' libiberty/Makefile.in &&

sed -i 's@\./fixinc\.sh@-c true@' gcc/Makefile.in &&

sed -i 's/@have\_mktemp\_command@/yes/' gcc/gccbug.in &&

mkdir ../gcc-build &&

cd ../gcc-build &&

../gcc-4.1.2/configure \

--prefix=/usr \

--libexecdir=/usr/lib \

--enable-shared \

--enable-threads=posix \

--enable-\_\_cxa\_atexit \

--enable-clocale=gnu \

--enable-languages=c,c++,ada,fortran,java,objc,treelang &&

make bootstrap &&

make -k check &&

../gcc-4.1.2/contrib/test\_summary

Now, as the root user:

make install &&

ln -v -sf ../usr/bin/cpp /lib &&

ln -v -sf gcc /usr/bin/cc &&

chown -v -R root:root \

/usr/lib/gcc/i686-pc-linux-gnu/4.1.2/include &&

chown -v -R root:root \

/usr/lib/gcc/i686-pc-linux-gnu/4.1.2/ada{lib,include}

The libffi interface header is installed in a location where other packages will not be able to find it. If you included Java as one of the installed languages, create a symbolic link in /usr/include to remedy this:

ln -v -sf `find /usr/lib/gcc -name ffitarget.h` /usr/include

As the root user, remove the GNAT installation:

rm -rf /opt/gnat

Now, as the unprivileged user, restore your old PATH:

export PATH=$PATH\_HOLD &&

unset PATH\_HOLD

**Command Explanations**

The three **sed** commands are the same ones used during the build of LFS. A fourth one in LFS is ommitted intentionally, due to the bootstrap build.

**mkdir ../gcc-build; cd ../gcc-build**: The GCC documentation recommends building the package in a dedicated build directory.

*--enable-shared --enable-threads=posix --enable-\_\_cxa\_atexit*: These parameters are required to build the C++ libraries to published standards.

*--enable-clocale=gnu*: This command is a failsafe for incomplete locale data.

*--enable-languages=c,c++,ada,fortran,java,objc,treelang*: This command identifies which languages to build. You may modify this command to remove undesired languages.

**make -k check**: This command runs the test suite without stopping if any errors are encountered.

**../gcc-4.1.2/contrib/test\_summary**: This command will produce a summary of the test suite results. You can append **| grep -A7 Summ** to the command to produce an even more condensed version of the summary. You may also wish to redirect the output to a file for review and comparison later on.

**ln -v -sf ../usr/bin/cpp /lib**: This command creates a link to the C PreProcessor as some packages expect it to be installed in the /lib directory.

**ln -v -sf gcc /usr/bin/cc**: This link is created as some packages refer to the C compiler using an alternate name.

**chown -v -R root:root /usr/lib/gcc/i686-pc-linux-gnu/...**: If the package is built by a user other than root, the ownership of the installed include and adalib directories (and their contents) will be incorrect. These commands change the ownership to the root user and group . Omit the command changing the Ada directories if you did not include Ada as one of the installed languages.

**Contents**

**Installed Programs:** addr2name.awk, fastjar, gcj, gcj-dbtool,gcjh, gfortran, gij, gjnih, gnat, gnatbind, gnatbl, gnatchop, gnatclean, gnatfind, gnatkr, gnatlink, gnatls, gnatmake, gnatname, gnatprep, gnatxref, gprmake, grepjar, grmic, grmiregistry, gtreelang, jcf-dump, jv-convert, jv-scan and architecture specific names for gcj and gcjh

**Installed Libraries:** libffi.{so,a}, libgcj.{so,a}, libgfortran.{so,a}, libgfortranbegin.a, libgij.{so,a}, libobjc.{so,a} and numerous other run-time libraries and executables in /usr/lib/gcc

**Installed Directories:** /usr/include/c++/4.1.2/{gcj,gnu,java,javax,org}, /usr/lib/gcc/i686-pc-linux-gnu/4.1.2/ada{include,lib}, /usr/lib/gcj-4.1.2, /usr/lib/security and /usr/share/java

Some program and library names and descriptions are not listed here, but can be found at [../../../../lfs/view/6.3/chapter06/gcc.html#contents-gcc](https://www.linuxfromscratch.org/lfs/view/6.3/chapter06/gcc.html#contents-gcc) as they were initially installed during the building of LFS.

**Short Descriptions**

|  |  |
| --- | --- |
| **addr2name.awk** | emulates some of the functionality of addr2line. |
| **fastjar** | is an archive tool for Java archives. |
| **gcj** | is an ahead-of-time compiler for the Java language. |
| **gcj-dbtool** | is a tool for creating and manipulating class file mapping databases. |
| **gcjh** | generates header files from Java class files. |
| **gfortran** | is the Fortran compiler invoked by **gcc**. |
| **gij** | is the GNU interpreter for Java bytecode. |
| **gjnij** | is used to generate JNI header files from class files. Running it is equivalent to running **gcjh -jni**. |
| **gnat** | is the Ada compiler invoked by **gcc**. |
| **gnatbind** | is used to bind compiled objects. |
| **gnatbl** | is the Ada linker. |
| **gnatchop** | is useful for renaming files to meet the standard Ada default file naming conventions. |
| **gnatclean** | is used to remove files associated with a GNAT project. |
| **gnatfind** | is the GNAT definition/use finder. |
| **gnatkr** | is used to determine the crunched name for a given file, when crunched to a specified maximum length. |
| **gnatlink** | is used to link programs and build an executable file. |
| **gnatls** | is the compiled unit browser. |
| **gnatmake** | is an automatic **make** facility. |
| **gnatname** | will list the files associated with a GNAT project. |
| **gnatprep** | is the GNAT external preprocessor. |
| **gnatxref** | is the GNAT cross-referencer. |
| **gprmake** | is a tool used to create Makefiles that support compilation by multiple languages. |
| **grepjar** | searches jar files for a pattern. |
| **grmic** | generates stubs for Remote Method Invocation. |
| **grmiregistry** | starts a remote object registry on the current host. |
| **gtreelang** | is largely a cut down version of C, designed to showcase the features of the GCC code generation back end. Only those features that are directly supported by the GCC code generation back end are implemented. Features are implemented in a manner which is easiest and clearest to implement. Not all or even most code generation back end features are implemented. The intention is to add features incrementally until most features of the GCC back end are implemented in Treelang. |
| **jcf-dump** | prints information about Java class files. |
| **jv-convert** | converts files from one encoding to another. |
| **jv-scan** | prints information about Java source files. |

Show more

Last updated on 2008-03-12 11:05:41 -0500