INSTALL GRASS from source code

Please read *all* text below.

Table of contents

PREREQUISITES

The install order matters. GRASS needs at least two libraries which have to be installed before installing/compiling GRASS: For links to the software, see ./REQUIREMENTS.html in this directory:

Installation order:

1. PROJ4
2. GDAL-OGR (compiled without GRASS support)
3. optionally: databases such as PostgreSQL, MySQL, sqlite
4. GRASS GIS
5. optionally: GDAL-OGR-GRASS plugin

(A) SOURCE CODE DISTRIBUTION

GRASS source code is currently distributed in 2 forms:

1) Officially released source code (e.g. grass-7.8.0.tar.gz or later)

  The Full source code version contains all the GRASS source code required for compilation. It is distributed as one file (*.tar.gz package) and the version is composed of 3 numbers, e.g. 7.8.0, 7.8.1 etc.

2) Snapshots of source code (generated from GitHub)

```
  This version of the source code can be acquired either from the
GitHub
  repository (https://github.com/OSGeo/grass/) or as a auto-generated
snapshot
  (*.tar.gz package) of the GitHub repository. The snapshot name
  contains the date when the snapshot was created (checked out from
  the GitHub repository), e.g.
grass-7.8.git_src_snapshot_2019_07_11.tar.gz
  from https://grass.osgeo.org/grass78/source/snapshot/
  Further instructions at https://trac.osgeo.org/grass/wiki/
DownloadSource


(B) COMPILATION

IMPORTANT: All Unix based distributions are different.
           For Solaris, see hints below.

The command,

   ./configure --help

explains the options used to disable the compilation of non-mandatory
GRASS modules. See REQUIREMENTS.html for details.

First step of the compilation (-g for debugging, or -O2 for
optimization):

    CFLAGS="-g -Wall" ./configure


Explanation of make targets:

    make install       - installs the binary

    make bindist       - make a binary package with install script

    make srcdist       - make a source package for distribution
    make srclibsdist   - make a source package for library
distribution

    make libs          - make libraries only

    make clean         - delete all files created by 'make'
    make distclean     - 'make clean' + delete all files created by
'./configure'
    make libsclean     - clean libraries compiled by 'make libs'

    make htmldocs      - generate programmer's documentation as HTML
```

files
    make packagehtmldocs - package programmer's documentation in HTML
    make pdfdocs   - generate programmer's documentation as PDF
files

Next step is the compilation itself:

    make

Detailed Wiki notes for various operating systems (MS-Windows, GNU/
Linux distros,
FreeBSD, AIX, etc) are available at:
https://grasswiki.osgeo.org/wiki/Compile_and_Install

Note for Solaris users (see also Wiki page above):

To configure GRASS correctly on a system which doesn't have a suitable
install program (AC_PROG_INSTALL ignores versions which are known to
have problems), you need to ensure that $srcdir is an absolute path,
by using e.g.:

        `pwd`/configure ...
or:
        ./configure --srcdir=`pwd` ...

Then proceed as described above.

Note when using a compiler different from "gcc":

By setting environment variables, the compiler
names can be defined (C and C++):

        CC=cc CPP=cpp ./configure ...

(C) COMPILATION NOTES for 64bit platforms

To successfully compile GRASS on 64bit platforms, the required
FFTW2 library has to be compiled with -fPIC flag:

    #this applies to FFTW3, not to GRASS GIS:
    cd fftw-3.3.4/
    CFLAGS="-fPIC" ./configure
    make
    make install

To fully enable 64bit library usage for GRASS on 64bit platforms,
the following additional parameters are recommended/required:

```
    ./configure \
       --enable-64bit \
       --with-libs=/usr/lib64 \
       ...
```

See also CODE OPTIMIZATION below.

(D) INSTALLATION (first time)

After compilation, the resulting code is stored in the directory
    ./dist.$ARCH
and the scripts (grass78, ...) in
    ./bin.$ARCH

To run GRASS, simply start
    ./bin.$ARCH/grass78

or run

```
    make install
    grass78
```

(E) INSTALLATION ON MACOSX

See the ReadMe.rtf in the ./macosx/ folder and the Wiki page above.

(F) RUNNING GRASS GIS

Download a sample data package from the GRASS web site, see
https://grass.osgeo.org/download/sample-data/

Extract the data set and point the "Database" field in the
GRASS GIS startup menu to the extracted directory.

Enjoy.

(G) UPDATE OF SOURCE CODE

Assuming that you want to update your current installation from
GitHub, you have to perform a few steps. In general:

- update from GitHub
- configure, compile

In detail:

```
    cd /where/your/grass7sourcecode/lives/
    git fetch --all
    git merge upstream/master
    ./configure ...
    make
    make install
```

For details, see https://trac.osgeo.org/grass/wiki/HowToGit

(H) COMPILING INDIVIDUAL MODULES - OWN MODULES

To compile (self-made) GRASS modules or to compile modified modules at least the GRASS libraries have to be compiled locally. This is done by launching:

```
    make libs
```

Then change into the module's directory and launch the "make" command. The installation can be either done with "make install" from the main source code directory or locally with
    "INST_NOW=y make"

You may want to define an alias for this:
```
    alias gmake7='INST_NOW=y make'
```
Then simply compile/install the current module with
```
    gmake7
```

Note: If you keep your module source code outside the standard GRASS source code directory structure, you will have to change the relative path(s) in the Makefile to absolute path(s).

(I) CODE OPTIMIZATION

If you would like to set compiler optimisations, for a possibly faster binary, type (don't enter a ";" anywhere):

```
        CFLAGS=-O ./configure
```
or,
```
        setenv CFLAGS -O
        ./configure
```

whichever works on your shell. Use -O2 instead of -O if your compiler supports this (note: O is the letter, not zero). Using the "gcc" compiler,
you can also specify processor specific flags (examples, please suggest
better settings to us):

```
  CFLAGS="-mcpu=athlon -O2" # AMD Athlon processor with code
optimisations
  CFLAGS="-mcpu=pentium"    # Intel Pentium processor
  CFLAGS="-mcpu=pentium4"   # Intel Pentium4 processor
  CFLAGS="-O2 -msse -msse2 -mfpmath=sse -minline-all-stringops" #
Intel XEON 64bit processor
  CFLAGS="-mtune=nocona -m64 -minline-all-stringops"            #
Intel Pentium 64bit processor

Note: As of version 4.3.0, GCC offers the -march=native switch that
enables CPU auto-detection and automatically selects optimizations
supported
by the local machine at GCC runtime including -mtune.

To find out optional CFLAGS for your platform, enter:
  gcc -dumpspecs

See also: http://gcc.gnu.org/

A real fast GRASS version (and small binaries) will be created with
LDFLAGS set to "stripping" (but this disables debugging):

  CFLAGS="-O2 -mcpu=<cpu_see_above> -Wall" LDFLAGS="-s" ./configure


(J) DEBUGGING OPTIONS

The LDFLAGS="" part must be undefined as "-s" will strip the debugging
information.

Don't use -O for CFLAGS if you want to be able to step through
function
bodies. When optimisation is enabled, the compiler will re-order
statements
and re-arrange expressions, resulting in object code which barely
resembles
the source code.

The -g and -Wall compiler flags are often useful for assisting
debugging:

  CFLAGS="-g -Wall" ./configure

See also the file ./doc/debugging.txt and the Wiki page
https://grasswiki.osgeo.org/wiki/GRASS_Debugging


(K) LARGE FILE SUPPORT (for raster maps)

GRASS >= 7.0.0 includes improved support for reading and writing large
```

files
(> 2GB) if it is possible in your operating system. If you compile with
  configure [...] --enable-largefile
you should be able to have raster and vector maps which are larger than 2GB.

While most code has been updated, individual programs may not yet work with
large files - please report.

See also
https://grasswiki.osgeo.org/wiki/GRASS_GIS_Performance
https://grasswiki.osgeo.org/wiki/Software_requirements_specification

(L) SUPPORT

Note that this code is still actively being developed and errors inevitably
turn up. If you find a bug, please report it to the GRASS bug tracking system
so we can fix it. See https://grass.osgeo.org/contribute/

If you are interested in helping to develop GRASS, please join the GRASS
developers mailing list. See https://grass.osgeo.org/development/

(M) GRASS PROGRAMMER'S MANUAL

The Programmer's manual is generated with doxygen from the source code.
Please see the README file and the files at:
https://grass.osgeo.org/programming7/

(N) CONTRIBUTING CODE AND PATCHES

Please see ./SUBMITTING in this directory, or better,
https://trac.osgeo.org/grass/wiki/Submitting

(O) DRAFT TUTORIAL

https://grass.osgeo.org/documentation/first-time-users/

Last changed: $Date$