

Linux/macOS 下编译 GMT 源码

贡献者： [田冬冬](#)

这一节介绍如何在 Linux 或 macOS 下编译 GMT 源代码。

安装依赖软件

GMT 的编译及运行依赖于其他软件。

必须的依赖软件包括：

- [CMake](#) ($\geq 2.8.12$)
- [netCDF](#) (≥ 4.0 且支持 netCDF-4/HDF5)
- [curl](#)

可选的依赖软件包括：

- [Ghostscript](#)：生成 PDF、JPG 等格式的图片
- [GDAL](#)：读写多种格式的地理空间数据（未安装则无法使用高精度地形数据）
- [GEOS](#)：地理信息系统的几何算法库
- [PCRE](#)：正则表达式支持
- [FFTW](#)：快速傅里叶变换库（ ≥ 3.3 ，macOS 下不需要）
- [GLib](#)：GTHREAD 多线程支持（ ≥ 2.32 ）
- LAPACK：快速矩阵反演库（macOS 下不需要）
- BLAS：快速矩阵运算库（macOS 下不需要）
- [GraphicsMagick](#)：生成 GIF 格式的动画
- [FFmpeg](#)：生成 MP4 格式的动画

Fedora:

```
# 安装必须软件包
$ sudo dnf install gcc cmake make glibc netcdf-devel libcurl-devel
$ sudo dnf install ghostscript gdal gdal-devel geos-devel lapack-devel openblas-devel glib2-devel pcre-devel fftw-devel
# 安装可选软件包
$ sudo dnf install https://download1.rpmfusion.org/free/fedora/rpmfusion-free-release-`rpm -E %fedora`.noarch.rpm
$ sudo dnf install GraphicsMagick ffmpeg
```

CentOS:

```
# 安装并启用 EPEL 源
$ sudo yum install epel-release
# 安装必须软件包
$ sudo yum install gcc cmake make glibc netcdf-devel libcurl-devel
$ sudo yum install ghostscript gdal gdal-devel geos-devel lapack-devel openblas-devel glib2-devel pcre-devel fftw-devel
# 安装可选软件包
$ sudo yum localinstall --nogpgcheck https://download1.rpmfusion.org/free/el/rpmfusion-free-release-`rpm -E %rhel`.noarch.rpm
$ sudo yum install GraphicsMagick ffmpeg
```

Ubuntu/Debian:

```
# 更新软件包列表
$ sudo apt update
# 安装必须软件包
$ sudo apt install build-essential cmake libcurl4-gnutls-dev libnetcdf-dev
$ sudo apt install ghostscript gdal-bin libgdal-dev libgeos-dev libglib2.0-dev libpcre3-dev libfftw3-dev liblapack-dev
# 安装可选软件包
$ sudo apt install graphicsmagick ffmpeg
```

macOS 用户可以使用 [Homebrew](#) 安装依赖（未安装 Homebrew 的用户，可以参考《[macOS 配置指南](#)》了解如何安装与使用）：

```
# 安装必须软件包
$ brew install cmake curl netcdf
$ brew install ghostscript gdal geos pcre2 glib fftw
# 安装可选软件包
$ brew install graphicsmagick ffmpeg
```

下载源码及数据

编译 GMT 需要下载如下三个文件：

1. GMT 6.4.0 源码：[gmt-6.4.0-src.tar.xz](#)
2. 全球海岸线数据 GSHHG：[gshhg-gmt-2.3.7.tar.gz](#)
3. 全球数字图表 DCW：[dcw-gmt-2.1.1.tar.gz](#)

❶ 备注

如果想编译 GMT 开发版，可以使用如下命令获取 GMT 最新开发版源码：

```
$ git clone --depth 50 https://github.com/GenericMappingTools/gmt
```

其余操作与编译 GMT 正式版没有区别。

安装 GMT

将下载的三个压缩文件放在同一个目录里，按照如下步骤进行安装：

```
# 解压三个压缩文件
$ tar -xvf gmt-6.4.0-src.tar.xz
$ tar -xvf gshhg-gmt-2.3.7.tar.gz
$ tar -xvf dcw-gmt-2.1.1.tar.gz

# 将 gshhg 和 dcw 数据复制到 gmt 的 share 目录下
$ mv gshhg-gmt-2.3.7 gmt-6.4.0/share/gshhg-gmt
$ mv dcw-gmt-2.1.1 gmt-6.4.0/share/dcw-gmt

# 切换到 gmt 源码目录下
$ cd gmt-6.4.0

# 用文本编辑器新建并打开 CMake 用户配置文件
# Linux 用户
$ gedit cmake/ConfigUser.cmake
# macOS 用户
$ touch cmake/ConfigUser.cmake
$ open -a TextEdit cmake/ConfigUser.cmake
```

向 `cmake/ConfigUser.cmake` 文件中加入如下语句：

```
set (CMAKE_INSTALL_PREFIX "/opt/GMT-6.4.0")
set (GMT_USE_THREADS TRUE)
```

- **CMAKE_INSTALL_PREFIX** 用于设置 GMT 的安装路径，上面的语句会将 GMT 安装在 `/opt/GMT-6.4.0` 目录下，用户可以自行修改为其他路径。没有 root 权限的一般用户，可以将安装路径设置为 `/home/xxx/opt/GMT-6.4.0` 等有可读写权限的路径
- **GMT_USE_THREADS** 设置为 **TRUE** 会为 GMT 的某些模块增加多线程并行功能以加速计算，也可以不设置

❶ 小技巧

此处为了便于一般用户理解，只向 `cmake/ConfigUser.cmake` 中写入了必要的语句。用户可以将 GMT 提供的配置模板 `cmake/ConfigUserTemplate.cmake` 复制为 `cmake/ConfigUser.cmake` 并根据配置文件中的大量注释说明信息自行修改配置文件。也可以进一步将高级配置模板 `cmake/ConfigUserAdvancedTemplate.cmake` 复制为 `cmake/ConfigUserAdvanced.cmake` 并根据注释说明信息修改高级配置。

继续执行如下命令以检查 GMT 的依赖是否满足：

❗ 备注

以下的 `mkdir build` 命令新建的 `build` 文件夹位于 GMT 源码压缩包解压出来的 `gmt-6.4.0` 目录下。不是 `gmt-6.4.0/cmake` 目录下，更不是 `/opt/GMT-6.4.0`。

```
$ mkdir build
$ cd build/
$ cmake ..
```

`cmake ..` 会检查系统软件是否满足 GMT 的依赖关系，过程中会输出大量信息，并在最后汇总输出检查结果。我们只需要关注检查结果是否正确即可。正常情况下结果结果如下，若存在一些差异也没有问题。只要过程中不出现报错，即可。如果出现报错，则需要检查之前的步骤是否有误，检查完成后删除原 build 目录再新建 build，继续执行 `cmake ..`，直到出现类似的检查结果：

```

*
* GMT Version:                : 6.4.0
*
* Options:
* Found GSHHG database        : /home/user/GMT/gmt-6.4.0/share/gshhg (2.3.7)
* Found DCW-GMT database      : /home/user/GMT/gmt-6.4.0/share/dcw-gmt (2.1.1)
* Found GMT data server       : oceania
* NetCDF library              : /usr/lib/x86_64-linux-gnu/libnetcdf.so
* NetCDF include dir          : /usr/include
* Curl library                : /usr/lib/x86_64-linux-gnu/libcurl.so
* Curl include dir            : /usr/include/x86_64-linux-gnu
* GDAL library                 : /usr/lib/libgdal.so
* GDAL include dir            : /usr/include/gdal
* GEOS library                 : /usr/lib/x86_64-linux-gnu/libgeos_c.so
* GEOS include dir            : /usr/include
* FFTW library                 : /usr/lib/x86_64-linux-gnu/libfftw3f.so
* FFTW threads library        : /usr/lib/x86_64-linux-gnu/libfftw3f_threads.so
* FFTW include dir            : /usr/include
* Accelerate Framework        :
* Regex support               : PCRE (/usr/lib/x86_64-linux-gnu/libpcre.so)
* ZLIB library                 : /usr/lib/x86_64-linux-gnu/libz.so
* ZLIB include dir            : /usr/include
* LAPACK library               : yes
* BLAS library                 : yes
* License restriction          : no
* Triangulation method        : Shewchuk
* OpenMP support               : disabled (GMT_ENABLE_OPENMP not set)
* GLIB GTHREAD support        : enabled (2.64.6)
* Build generator              : Unix Makefiles
* Build GMT core               : always [libgmt.so]
* Build PSL library           : always [libpostscriptlight.so]
* Build GMT supplements        : yes [supplements.so]
* Build GMT for developers    : yes
* Build proto supplements      : none
* Build module links           : no
* Found Ghostscript (gs)      : yes (9.50)
* Found GraphicsMagick (gm)    : yes (1.3.35)
* Found ffmpeg                 : yes (4.2.4)
* Found open                   : no
* Found ogr2ogr                : yes (3.0.4)
* Found gdal_translate         : yes (3.0.4)
*
* Locations:
* Installing GMT in           : /opt/GMT-6.4.0
* GMT_DATADIR                  : /opt/GMT-6.4.0/share
* GMT_DOCDIR                    : /opt/GMT-6.4.0/share/doc
* GMT_MANDIR                    : /opt/GMT-6.4.0/share/man
-- Configuring done
-- Generating done

```

⚠ 警告

Anaconda 用户请注意！由于 Anaconda 中也安装了 FFTW、GDAL、netCDF 等库文件，GMT 在配置过程中可能会找到 Anaconda 提供的库文件，进而导致配置、编译或执行过程中出错。

解决办法是，在 Shell 配置文件 (`~/.bashrc` 或 `~/.zshrc`) 中将 Anaconda 相关的环境变量注释掉，以保证 GMT 在配置和编译过程中找到的不是 Anaconda 提供的库文件。待 GMT 安装完成后，再将 Anaconda 相关环境变量改回即可。

检查完毕后，开始编译和安装:

```
$ make -j
$ sudo make -j install
```

❗ 备注

`-j` 选项可以实现并行编译以减少编译时间。但据用户报告，某些 Ubuntu 发行版下使用 `-j` 选项会导致编译过程卡死。Ubuntu 用户建议上面的两条命令中去掉 `-j` 选项。

修改环境变量

打开终端，使用如下命令用文件编辑器打开 Shell 配置文件:

```
# Linux 用户
$ gedit ~/.bashrc

# macOS 用户
$ open ~/.zshrc
```

然后向文件末尾加入如下语句以修改环境变量。修改完成后保存文件并退出，然后重启终端使其生效:

```
export GMT6HOME=/opt/GMT-6.4.0
export PATH=${GMT6HOME}/bin:$PATH
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:${GMT6HOME}/lib64
```

说明：

- 第一个命令添加了环境变量 **GMT6HOME**
- 第二个命令修改 GMT6 的 `bin` 目录加入到 **PATH** 中，使得在终端或脚本中可以找到 GMT 命令
- 第三个命令将 GMT6 的 `lib` 目录加入到动态链接库路径中。通常，32 位系统的路径为 `lib`，64 位系统的路径为 `lib64`

测试是否安装成功

重新打开一个终端，键入如下命令，若正确显示 GMT 版本号，则表示安装成功:

```
$ gmt --version  
6.4.0
```

升级/卸载 GMT

按照上面的配置，GMT 会被安装到 `/opt/GMT-6.4.0` 目录下。若想要卸载 GMT，可以直接删除整个 `/opt/GMT-6.4.0` 即可。

GMT 不支持自动更新，因而若想要升级 GMT，通常建议先卸载 GMT，然后再下载新版源码并按照上面的步骤重新编译安装。

当然，高级用户也可以同时安装多个版本的 GMT，但需要注意环境变量 **PATH** 的设置。