PanpanGIS 文档

PostgreSQL, PostGIS, SQLCarto, Apache, PHP, QT, QML

作者：麓山老将，PcWang，隔壁老王

组织：CSU.GIS.250 + 1

时间：2024 年 10 月 15 日

版本：1.0

邮箱：sqlcarto@163.com

支离东北风尘际，漂泊西南天地间。三峡楼台淹日月，五溪衣服共云山。
羯胡事主终无赖，词客哀时且未还。庾信平生最萧瑟，暮年诗赋动江关。
—— 杜甫（唐）

目录

第一章 缘起 **1**

1.1 还是干点啥吧 1

第二章 地图符号 **2**

2.1 类结构 2

第三章 Apache 安装 **4**

3.1 Apache 跨域设置 4

第四章 Windows WSL(Ubuntu) 子系统 **5**

4.1 WSL 安装 5

4.2 将 WSL 迁移到非系统盘 5

4.2.1 查看所有分发版本 5

4.2.2 导出分发版为 tar 文件 5

4.2.3 注销当前分发版 5

4.2.4 重新导入并安装分发版 5

4.2.5 设置默认登陆用户为安装时用户名 5

4.2.6 cmake 权限 5

第五章 Windows msys2 之 PostGIS/PostgreSQL 编译安装 **7**

5.1 安装 msys2 7

5.2 启动 msys2 ucrt 7

5.3 安装 ucrt 编译工具链 8

5.3.1 查询工具链 8

5.3.2 安装工具链 8

5.4 编译 postgresql 8

5.4.1 设置环境变量 8

5.4.2 安装支持库 9

5.4.3 安装 uuid 9

5.4.4 安装 postgresql 10

5.4.5 安装 geos 12

5.4.6 安装 gdal 12

5.4.7 安装 postgis 20

5.5 打包 postgresql 22

第六章 Ubuntu 之 PostGIS/PostgreSQL 编译安装 **24**

6.1 系统版本 24

6.2 安装编译工具链 24

6.3 安装支持库 24

6.4 设置环境变量 25

6.5 安装 postgresql 25

6.6 安装 geos 26

6.7 安装 gdal 26

6.8 安装 postgis 27

第七章 PostGIS/PostgreSQL 扩展方法 **28**

第一章 缘起

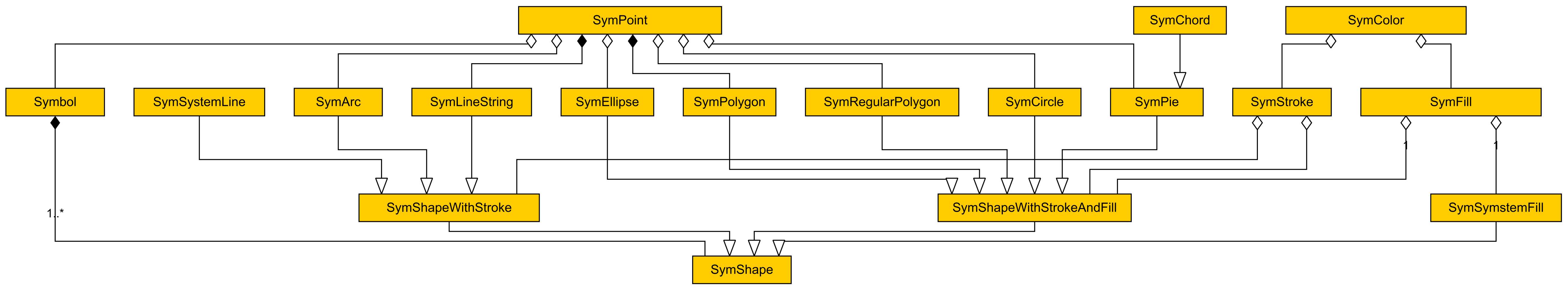
1.1 还是干点啥吧

去年冬日，麓山的树被冰雪压弯，折断不少枝叶，掉落在地。初春上午，孤独行走山间，绕过断枝残叶，漫无目的爬向山顶。路边的小野花开得正旺，串串的淡黄，间或有几只蜜蜂起落其间。这种灌木开出的小花被父亲叫做青叶子花，是初春重要的蜜源。脚下的残枝败叶与枝梢间的嫩芽正昭示着世间真理：旧的已去，新的正来。当一个人在某个行当久了，成为资深，是否也正如这掉落的枝叶，在某个时间节点，无可奈地要随风去!?

外观落幕，内自落寞。离开吧！过去五年，这个声音一直耳边响着。从五楼搬到二楼，换到规律而 boring 的工作。无人打扰，时间充足。Tong 对我说，一个人一天接不到几个电话，接到的电话还是亲朋老友打过来的，说明没有希望了。

第二章 地图符号

2.1 类结构


SymPoint
SymChord
SymColor
Symbol
SymSystemLine
SymArc
SymLineString
SymEllipse
SymPolygon
SymRegularPolygon
SymCircle
SymPie
SymStroke
SymFill
1
1
1..*
SymShapeWithStroke
SymShapeWithStrokeAndFill
SymSymstemFill
SymShape

第三章 Apache 安装

3.1 Apache 跨域设置

如果 web 服务器的跨域参数设置不正确，在客户端为 webassembly、服务端为 https 时，会出现“a breakdown in protocol was detected (parsing error, invalid or unexpected responses, etc.)”的错误。

修改 apache.conf, 添加如下设置：

```
<Directory />
  AllowOverride none
  Require all denied
  Header set Access-Control-Allow-Origin *
  Header set Cross-Origin-Embedder-Policy 'require-corp'
  Header set Cross-Origin-Opener-Policy 'same-origin'
</Directory>
```

第四章 Ubuntu 之 PostGIS/PostgreSQL 编译安装

4.1 系统版本

Ubuntu 2024.04 LTS

4.2 安装编译工具链

```
# 更新ubuntu源
sudo apt update
# 安装基础编译工具gcc
sudo apt install build-essential
# cmake
sudo apt install cmake
# git
sudo apt install git
# curl
sudo apt install curl
# ninja
sudo apt install ninja generate-ninja
```

4.3 安装支持库

```
# libreadline用来记录psql中的输入命令
sudo apt install libreadline-dev
# ossp uuid 生成唯一id
sudo apt install libossp-uuid-dev
# postgis使用sfcgal进行3D分析，sfcgal是计算机图形库cgal的c语言wrapper
sudo apt install libsfcgal-dev
# 地图投影变换
sudo apt install libproj-dev
# 地理栅格数据格式geotiff
sudo apt install libgeotiff-dev
#
sudo apt install libfreetype-dev
# cairo二维图形绘制库
# 我本来是想使用skia图形库来绘制地图的，但是skia的代码下载和编译需要连接境外服务器，而现在的网络环境一言
    难尽....。所以就只能换成cairo图形库来进行地图的绘制咯。
sudo apt install libcairo-dev
# 安装curl开发库，如果下面一行命令运行错误，可能是找不到软件包名
# 请改用sudo apt install libcurl4-openssl-dev
sudo apt install libcurl-dev
sudo apt install libicu-dev
sudo apt install libxml2-dev
sudo apt install protobuf-c-compiler
sudo apt install libprotobuf-c-dev
```

上面的所有支持库可以放在一条命令中：

```
sudo apt install libreadline-dev libossp-uuid-dev libsfcgal-dev libproj-dev libgeotiff-dev libcairo-dev libcurl-dev libicu-dev libxml2-dev protobuf-c-compiler libprotobuf-c-dev
```

4.4 设置环境变量

编辑/etc/profile，在最末尾添加下列内容：

```
export PGSQL=/usr/local/pgsql
export PATH=$PGSQL/bin:$PGSQL/lib:$PATH
export LD_LIBRARY_PATH=$PGSQL/lib:$LD_LIBRARY_PATH
export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:$PGSQL/lib/pkgconfig
```

注销，重新登录。

4.5 安装 postgresql

```
cd ~/software/sdb

# 下载源代码
# 注意curl命令使用的参数，此处模拟chrome来下载文件
# 如直接使用：
# curl 'https://ftp.postgresql.org/pub/source/v16.4/postgresql-16.4.tar.bz2' --output postgresql-16.4.tar.bz2
# 则不能成功下载源码文件
# 可以根据需要改变版本号来下载对应的版本的postgresql源码
# curl 'https://ftp.postgresql.org/pub/source/v16.4/postgresql-16.4.tar.bz2' \
# -H 'Referer: https://www.postgresql.org/' \
# -H 'Upgrade-Insecure-Requests: 1' \
# -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36' \
# -H 'sec-ch-ua: "Chromium";v="128", "Not;A=Brand";v="24", "Google Chrome";v="128"' \
# -H 'sec-ch-ua-mobile: ?0' \
# -H 'sec-ch-ua-platform: "Windows"' \
# --output postgresql-16.4.tar.bz2
# 使用curl的-LJO 参数下载文件
curl -LJO 'https://ftp.postgresql.org/pub/source/v16.4/postgresql-16.4.tar.bz2'

tar -jvxf postgresql-16.4.tar.bz2
cd postgresql-16.4
# 由于uuid没有安装在/usr目录，而是安装在$PGSQL目录下，因此在配置时
# 应该指定LDFLAGS和CPPFLAGS环境变量，否则会提示找不到uuid库
./configure --prefix=$PGSQL --with-ossp-uuid
make -j 8
sudo make install
```

```
# 编译postgresql的uuid扩展
cd contrib/uuid-ossp
make
sudo make install

# 编译postgresql的dblink扩展
cd ../dblink
make
sudo make install

# 编译postgresql的file_fdw扩展
cd ../file_fdw
make
sudo make install
```

4.6 安装 geos

geos 提供空间分析功能。

```
cd ~/software/sdb
curl -LJO https://download.osgeo.org/geos/geos-3.12.2.tar.bz2
tar -jvxf geos-3.12.2.tar.bz2
cd geos-3.12.2
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=$PGSQL -DCMAKE_BUILD_TYPE=Release ..
# 某些geos版本已经不能使用make构建工具，只能使用下列的cmake命令进行构建和安装
cmake --build . -- -j 8
# 安装
sudo cmake --build . --target install
```

4.7 安装 gdal

gdal 提供地理空间数据矢量、栅格文件读取 api。如需使用 postgis 的 raster 扩展功能，必须安装 gdal。

```
cd ~/software/sdb

# 下载源代码
# 注意curl命令使用的参数
curl -LJO https://download.osgeo.org/gdal/3.9.2/gdal-3.9.2.tar.gz
tar -zvxf gdal-3.9.2.tar.gz
cd gdal-3.9.2
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=$PGSQL -DCMAKE_BUILD_TYPE=Release ..
# 编译
cmake --build . -- -j 8
# 安装
```

```
sudo cmake --build . --target install
```

4.8 安装 postgis

```
cd ~/software/sdb
curl -LJO https://download.osgeo.org/postgis/source/postgis-3.4.2.tar.gz
tar -zvxf postgis-3.4.2.tar.gz
cd postgis-3.4.2
./configure --prefix=$PGSQL
make -j 8
sudo make install
```

第五章 Windows WSL(Ubuntu) 子系统安装 PostGIS/PostgreSQL

Window 操作系统可以先通过 WSL 子系统安装 Ubuntu。安装好 Ubuntu 系统后，即可参见上一章编译安装 PostGIS/PostgreSQL。

5.1 WSL 安装

5.2 将 WSL 迁移到非系统盘

wsl 默认安装位置是 C 盘，众所周知 C 盘总是不够用的，所以才有了把 wsl 的系统迁移到其它位置的需求。在下列情况出现时需要将 WSL 从系统盘迁移 (C) 到非系统盘 (D,E 等):

- 系统盘空间有限
- 数据库目录存在于 WSL 文件系统，但是预期数据库空间占用非常巨大

5.2.1 查看所有分发版本

```
  wsl -l --all -v
```

5.2.2 导出分发版为 tar 文件

```
wsl --export Ubuntu-20.04 D:\ubuntu20.04.tar
```

5.2.3 注销当前分发版

```
wsl --unregister Ubuntu-20.04
```

5.2.4 重新导入并安装分发版

```
wsl --import Ubuntu-20.04 D:\wsl\ubuntu D:\ubuntu20.04.tar --version 2
```

5.2.5 设置默认登陆用户为安装时用户名

```
ubuntu2004 config --default-user Username
```

5.2.6 cmake 权限

在 ubuntu 迁移到非 c 盘后，使用 cmake 时会遇到权限问题:

```
CMake Error at /usr/share/cmake-3.22/Modules/CMakeDetermineSystem.cmake:193 (configure_file):
Operation not permitted
Call Stack (most recent call first):
CMakeLists.txt:3 (project)
```

可以通过修改 wsl 系统的配置文件来解决此问题，在/etc/wsl.conf 最后增加下面内容：

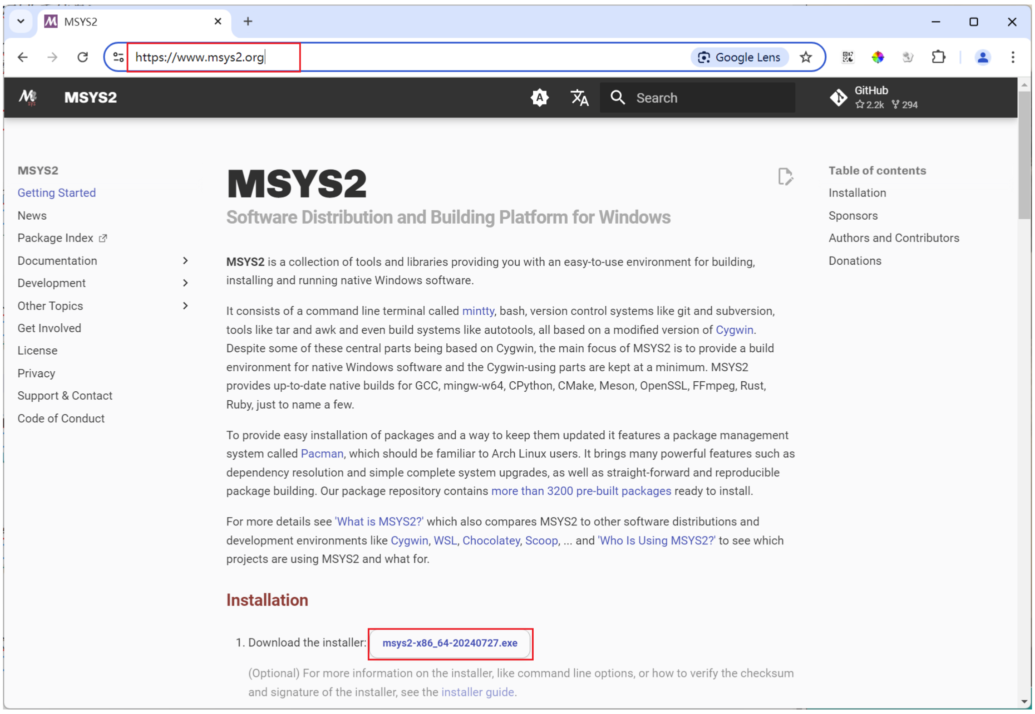
```
......

[automount]
enabled = true
options = "metadata"
```

重新启动 wsl 后此问题可得到解决。

第六章 Windows msys2 之 PostGIS/PostgreSQL 编译安装

6.1 安装 msys2

从https://www.msys2.org/网站下载 msys 安装包，双击安装 msys2。



MSYS2

Software Distribution and Building Platform for Windows

MSYS2 is a collection of tools and libraries providing you with an easy-to-use environment for building, installing and running native Windows software.

It consists of a command line terminal called mintty, bash, version control systems like git and subversion, tools like tar and awk and even build systems like autotools, all based on a modified version of Cygwin. Despite some of these central parts being based on Cygwin, the main focus of MSYS2 is to provide a build environment for native Windows software and the Cygwin-using parts are kept at a minimum. MSYS2 provides up-to-date native builds for GCC, mingw-w64, CPython, CMake, Meson, OpenSSL, FFmpeg, Rust, Ruby, just to name a few.

To provide easy installation of packages and a way to keep them updated it features a package management system called Pacman, which should be familiar to Arch Linux users. It brings many powerful features such as dependency resolution and simple complete system upgrades, as well as straight-forward and reproducible package building. Our package repository contains more than 3200 pre-built packages ready to install.

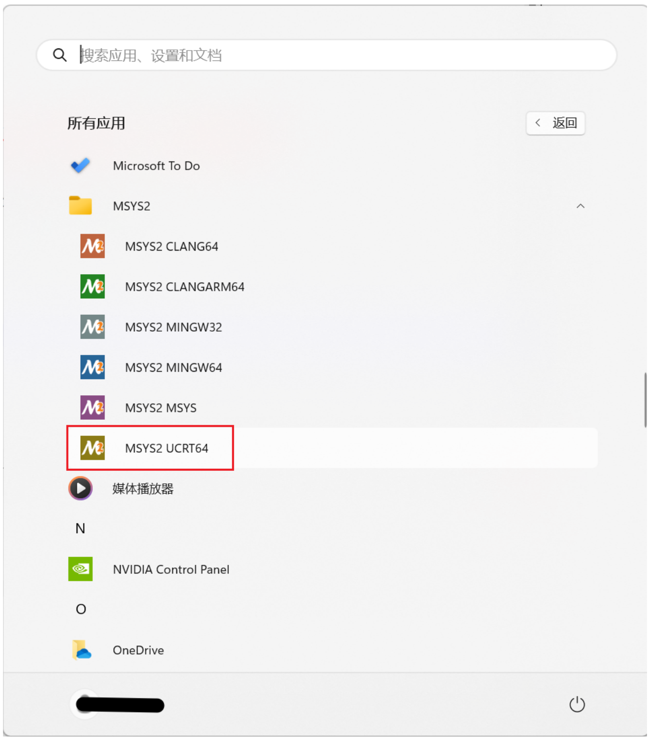
For more details see 'What is MSYS2?' which also compares MSYS2 to other software distributions and development environments like Cygwin, WSL, Chocolatey, Scoop, ... and 'Who Is Using MSYS2?' to see which projects are using MSYS2 and what for.

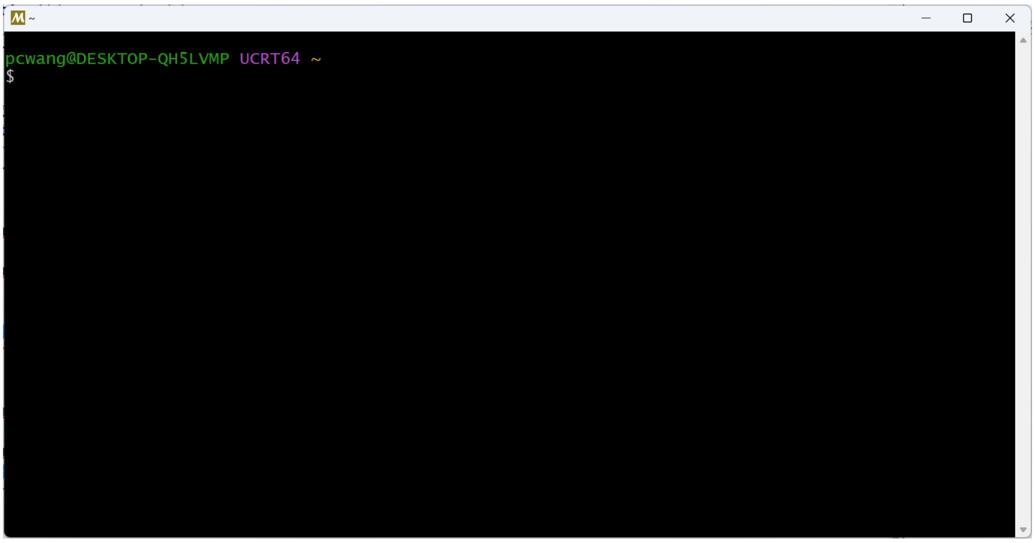
Installation

1. Download the installer: msys2-x86_64-20240727.exe

(Optional) For more information on the installer, like command line options, or how to verify the checksum and signature of the installer, see the installer guide.

6.2 启动 msys2 ucrt



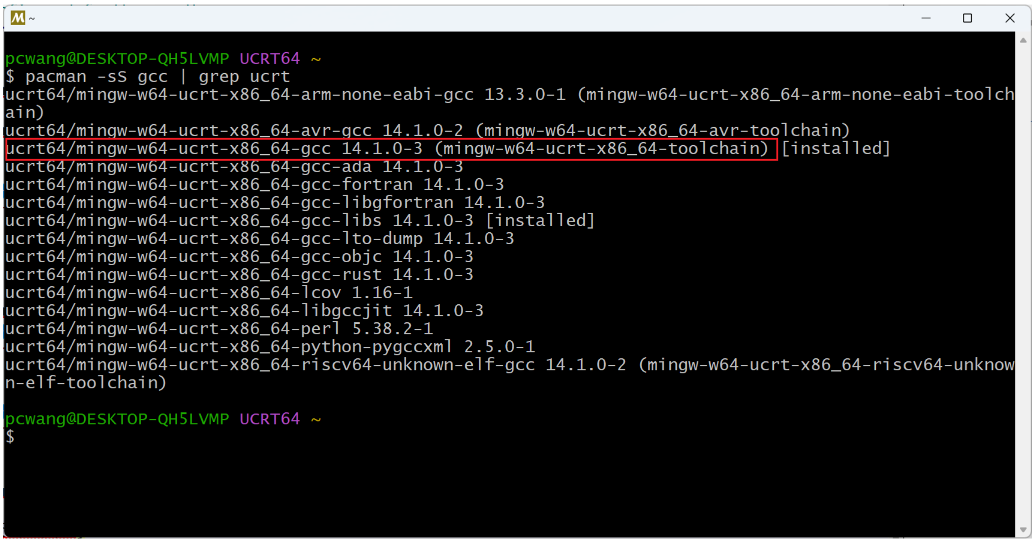


```
pcwang@DESKTOP-QH5LVMP UCRT64 ~
$
```

6.3 安装 ucrt 编译工具链

6.3.1 查询工具链

```
pacman -sS gcc | grep ucrt
```



```
pcwang@DESKTOP-QH5LVMP UCRT64 ~
$ pacman -sS gcc | grep ucrt
ucrt64/mingw-w64-ucrt-x86_64-arm-none-eabi-gcc 13.3.0-1 (mingw-w64-ucrt-x86_64-arm-none-eabi-toolch
ain)
ucrt64/mingw-w64-ucrt-x86_64-avr-gcc 14.1.0-2 (mingw-w64-ucrt-x86_64-avr-toolchain)
ucrt64/mingw-w64-ucrt-x86_64-gcc 14.1.0-3 (mingw-w64-ucrt-x86_64-toolchain) [installed]
ucrt64/mingw-w64-ucrt-x86_64-gcc-ada 14.1.0-3
ucrt64/mingw-w64-ucrt-x86_64-gcc-fortran 14.1.0-3
ucrt64/mingw-w64-ucrt-x86_64-gcc-libgfortran 14.1.0-3
ucrt64/mingw-w64-ucrt-x86_64-gcc-libs 14.1.0-3 [installed]
ucrt64/mingw-w64-ucrt-x86_64-gcc-lto-dump 14.1.0-3
ucrt64/mingw-w64-ucrt-x86_64-gcc-objc 14.1.0-3
ucrt64/mingw-w64-ucrt-x86_64-gcc-rust 14.1.0-3
ucrt64/mingw-w64-ucrt-x86_64-lcov 1.16-1
ucrt64/mingw-w64-ucrt-x86_64-libgccjit 14.1.0-3
ucrt64/mingw-w64-ucrt-x86_64-perl 5.38.2-1
ucrt64/mingw-w64-ucrt-x86_64-python-pygccxml 2.5.0-1
ucrt64/mingw-w64-ucrt-x86_64-riscv64-unknown-elf-gcc 14.1.0-2 (mingw-w64-ucrt-x86_64-riscv64-unknow
n-elf-toolchain)

pcwang@DESKTOP-QH5LVMP UCRT64 ~
$
```

6.3.2 安装工具链

```
pacman -S mingw-w64-ucrt-x86_64-toolchain
pacman -S ucrt64/mingw-w64-ucrt-x86_64-cmake
pacman -S msys/make
pacman -S mingw-w64-ucrt-x86_64-autotools
```

6.4 编译 postgresql

6.4.1 设置环境变量

编辑/etc/profile，在最末尾添加下列内容：

```
export PGSQL=/usr/local/pgsql
export PATH=$PGSQL/bin:$PGSQL/lib:$PATH
export LD_LIBRARY_PATH=$PGSQL/lib:$LD_LIBRARY_PATH
export PKG_CONFIG_PATH=$PKG_CONFIG_PATH:$PGSQL/lib/pkgconfig
```

重新启动 msys2 ucrt 命令行。

6.4.2 安装支持库

```
# postgis利用cgal支持3D空间分析
# sfcgal为cgal的c语言wrapper, 只有c语言的库才能链接到postgresql
pacman -S ucrt64/mingw-w64-ucrt-x86_64-sfcgal

# cairo为图形库, sqlcarto使用cairo进行二维地图绘制
pacman -S ucrt64/mingw-w64-ucrt-x86_64-cairo

# libgeotiff为geotiff栅格文件的读写库, gdal需要
pacman -S ucrt64/mingw-w64-ucrt-x86_64-libgeotiff

# netcdf读取大数据文件, 地理空间栅格数据
pacman -S ucrt64/mingw-w64-ucrt-x86_64-netcdf

pacman -S ucrt64/mingw-w64-ucrt-x86_64-protobuf-c

pacman -S ucrt64/mingw-w64-ucrt-x86_64-libxslt
```

6.4.3 安装 uuid

```
cd ~
mkdir -p software/sdb
cd software/sdb
curl --output uuid-1.6.1.tar.gz ftp://ftp.ossp.org/pkg/lib/uuid/uuid-1.6.1.tar.gz
tar -zvxf uuid-1.6.1.tar.gz
cd uuid-1.6.1
```

使用编辑器打开 shtool 脚本文件，找到 dsttmp 变量赋值语句：

```
...
#  make a temp file name in the destination directory
dsttmp=`echo $dst |\
  sed -e 's;[^/]*$;;' -e 's;\(.\)/$;\1;' -e 's;^$;.;' \
  -e "s;\$;/#INST@$$#;"`
...
```

在后面添加一行，改成：

```
...
#  make a temp file name in the destination directory
dsttmp=`echo $dst |\
  sed -e 's;[^/]*$;;' -e 's;\(.\)/$;\1;' -e 's;^$;.;' \
  -e "s;\$;/#INST@$$#;"`
# 添加下面这一行
dsttmp=$dsttmp.exe
...
```

保存 shtool 文件，在控制台运行下列命令行，配置及安装 uuid：

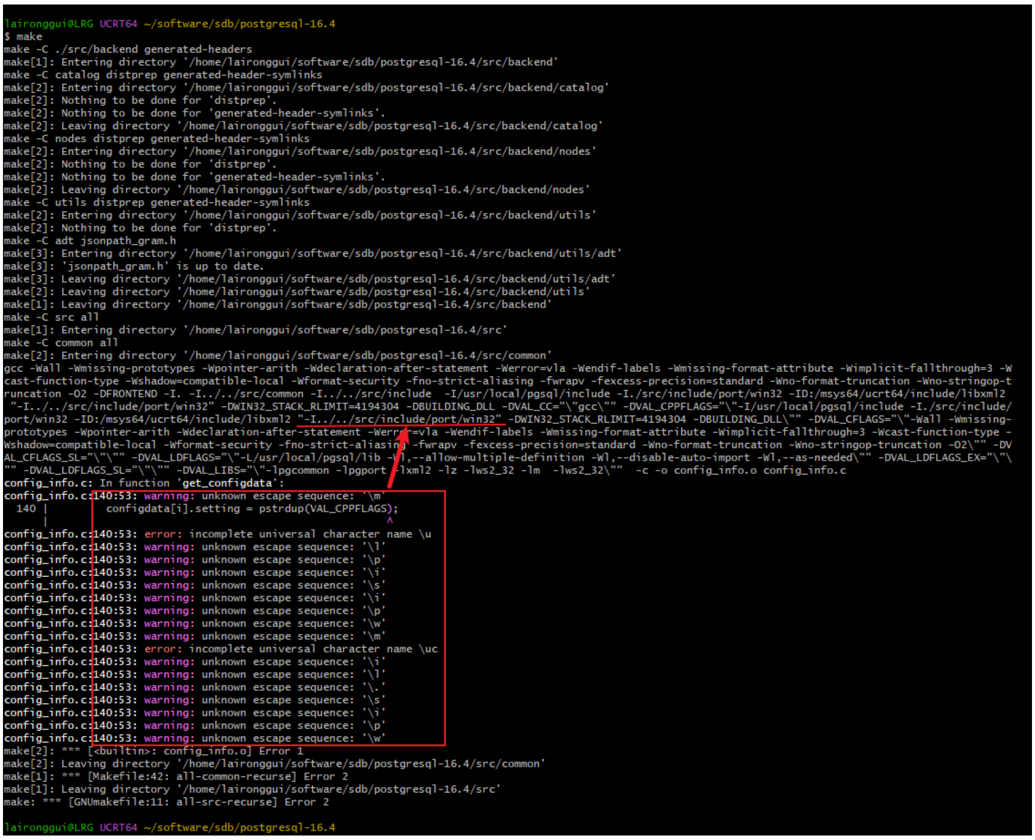
```
./configure --prefix=$PGSQL
make
make install
```

6.4.4 安装 postgresql

```
cd ~/software/sdb

# 下载源代码
# 注意curl命令使用的参数，此处模拟chrome来下载文件
# 如直接使用：
# curl 'https://ftp.postgresql.org/pub/source/v16.4/postgresql-16.4.tar.bz2' --output postgresql-16.4.tar.bz2
# 则不能成功下载源码文件
# 可以根据需要改变版本号来下载对应的版本的postgresql源码
# curl 'https://ftp.postgresql.org/pub/source/v16.4/postgresql-16.4.tar.bz2' \
# -H 'Referer: https://www.postgresql.org/' \
# -H 'Upgrade-Insecure-Requests: 1' \
# -H 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/128.0.0.0 Safari/537.36' \
# -H 'sec-ch-ua: "Chromium";v="128", "Not;A=Brand";v="24", "Google Chrome";v="128"' \
# -H 'sec-ch-ua-mobile: ?0' \
# -H 'sec-ch-ua-platform: "Windows"' \
# --output postgresql-16.4.tar.bz2
# 使用curl的-LJO 参数下载文件
curl -LJO 'https://ftp.postgresql.org/pub/source/v16.4/postgresql-16.4.tar.bz2'

tar -jvxf postgresql-16.4.tar.bz2
cd postgresql-16.4
# 由于uuid没有安装在/usr目录，而是安装在$PGSQL目录下，因此在配置时
# 应该指定LDFLAGS和CPPFLAGS环境变量，否则会提示找不到uuid库
LDFLAGS="-L$PGSQL/lib" CPPFLAGS="-I$PGSQL/include" ./configure --prefix=$PGSQL --with-ossp-uuid --with-libxml
make
make install

# 编译postgresql的uuid扩展
cd contrib/uuid-ossp
make
make install
```

```
# 编译postgresql的dblink扩展
cd ../dblink
make
make install

# 编译postgresql的file_fdw扩展
cd ../file_fdw
make
make install
```

如果在上述安装过程中出现下图所示错误[1]：



```
laironggui@LRG UCRT64 ~/software/sdb/postgresql-16.4
$ make
make -C ./src/backend generated-headers
make[1]: Entering directory '/home/laironggui/software/sdb/postgresql-16.4/src/backend'
make -C catalog distprep generated-header-symlinks
make[2]: Entering directory '/home/laironggui/software/sdb/postgresql-16.4/src/backend/catalog'
make[2]: Nothing to be done for 'distprep'.
make[2]: Nothing to be done for 'generated-header-symlinks'.
make[2]: Leaving directory '/home/laironggui/software/sdb/postgresql-16.4/src/backend/catalog'
make -C nodes distprep generated-header-symlinks
make[2]: Entering directory '/home/laironggui/software/sdb/postgresql-16.4/src/backend/nodes'
make[2]: Nothing to be done for 'distprep'.
make[2]: Nothing to be done for 'generated-header-symlinks'.
make[2]: Leaving directory '/home/laironggui/software/sdb/postgresql-16.4/src/backend/nodes'
make -C utils distprep generated-header-symlinks
make[2]: Entering directory '/home/laironggui/software/sdb/postgresql-16.4/src/backend/utils'
make[2]: Nothing to be done for 'distprep'.
make -C adt jsonpath_gram.h
make[3]: Entering directory '/home/laironggui/software/sdb/postgresql-16.4/src/backend/utils/adt'
make[3]: 'jsonpath_gram.h' is up to date.
make[3]: Leaving directory '/home/laironggui/software/sdb/postgresql-16.4/src/backend/utils/adt'
make[2]: Leaving directory '/home/laironggui/software/sdb/postgresql-16.4/src/backend/utils'
make[1]: Leaving directory '/home/laironggui/software/sdb/postgresql-16.4/src/backend'
make -C src all
make[1]: Entering directory '/home/laironggui/software/sdb/postgresql-16.4/src'
make -C common all
make[2]: Entering directory '/home/laironggui/software/sdb/postgresql-16.4/src/common'
gcc -Wall -Wmissing-prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wimplicit-fallthrough=3 -W
cast-function-type -Wshadow=compatible-local -Wformat-security -fno-strict-aliasing -fwrapv -fexcess-precision=standard -Wno-format-truncation -Wno-stringop-t
runcation -O2 -DFRONTEND -I. -I../../src/common -I../../src/include  -I/usr/local/pgsql/include -I./src/include/port/win32 -ID:/msys64/ucrt64/include/libxml2
 "-I../../src/include/port/win32" -DWIN32_STACK_RLIMIT=4194304 -DBUILDING_DLL -DVAL_CC="\"gcc\"" -DVAL_CPPFLAGS="\"-I/usr/local/pgsql/include -I./src/include/
port/win32 -ID:/msys64/ucrt64/include/libxml2 "-I../../src/include/port/win32" -DWIN32_STACK_RLIMIT=4194304 -DBUILDING_DLL\"" -DVAL_CFLAGS="\"-Wall -Wmissing-
prototypes -Wpointer-arith -Wdeclaration-after-statement -Werror=vla -Wendif-labels -Wmissing-format-attribute -Wimplicit-fallthrough=3 -Wcast-function-type -
Wshadow=compatible-local -Wformat-security -fno-strict-aliasing -fwrapv -fexcess-precision=standard -Wno-format-truncation -Wno-stringop-truncation -O2\"" -DV
AL_CFLAGS_SL="\"\"" -DVAL_LDFLAGS="\"-L/usr/local/pgsql/lib -Wl,--allow-multiple-definition -Wl,--disable-auto-import -Wl,--as-needed\"" -DVAL_LDFLAGS_EX="\"\
"" -DVAL_LDFLAGS_SL="\"\"" -DVAL_LIBS="\"-lpgcommon -lpgport -lxml2 -lz -lws2_32 -lm  -lws2_32\""  -c -o config_info.o config_info.c
config_info.c: In function 'get_configdata':
config_info.c:140:53: warning: unknown escape sequence: '\m'
  140 |         configdata[i].setting = pstrdup(VAL_CPPFLAGS);
      |                                                     ^
config_info.c:140:53: error: incomplete universal character name \u
config_info.c:140:53: warning: unknown escape sequence: '\l'
config_info.c:140:53: warning: unknown escape sequence: '\p'
config_info.c:140:53: warning: unknown escape sequence: '\i'
config_info.c:140:53: warning: unknown escape sequence: '\s'
config_info.c:140:53: warning: unknown escape sequence: '\i'
config_info.c:140:53: warning: unknown escape sequence: '\p'
config_info.c:140:53: warning: unknown escape sequence: '\w'
config_info.c:140:53: warning: unknown escape sequence: '\m'
config_info.c:140:53: error: incomplete universal character name \uc
config_info.c:140:53: warning: unknown escape sequence: '\i'
config_info.c:140:53: warning: unknown escape sequence: '\l'
config_info.c:140:53: warning: unknown escape sequence: '\.'
config_info.c:140:53: warning: unknown escape sequence: '\s'
config_info.c:140:53: warning: unknown escape sequence: '\i'
config_info.c:140:53: warning: unknown escape sequence: '\p'
config_info.c:140:53: warning: unknown escape sequence: '\w'
make[2]: *** [<builtin>: config_info.o] Error 1
make[2]: Leaving directory '/home/laironggui/software/sdb/postgresql-16.4/src/common'
make[1]: *** [Makefile:42: all-common-recurse] Error 2
make[1]: Leaving directory '/home/laironggui/software/sdb/postgresql-16.4/src'
make: *** [GNUmakefile:11: all-src-recurse] Error 2

laironggui@LRG UCRT64 ~/software/sdb/postgresql-16.4
```

则应该修改 software /sdb/postgresql-16.4/src/Makefile.port 的代码：

```
1  # src/makefiles/Makefile.win32
2
3  ifdef PGXS
4  BE_DLLLIBS= -L$(libdir) -lpostgres
5  override CPPFLAGS+= -I$(includedir_server)/port/win32
6  else
7  BE_DLLLIBS= -L$(top_builddir)/src/backend -lpostgres
8  override CPPFLAGS+="-I$(top_srcdir)/src/include/port/win32"
9  endif
10 ...
```

修改第 8 行代码，将双引号改成单引号：

[1] Ronggui Lai 在 win10 下实测，作者在 win11 上未遇到。

```
# src/makefiles/Makefile.win32

ifdef PGXS
BE_DLLLIBS= -L$(libdir) -lpostgres
override CPPFLAGS+= -I$(includedir_server)/port/win32
else
BE_DLLLIBS= -L$(top_builddir)/src/backend -lpostgres
override CPPFLAGS+='-I$(top_srcdir)/src/include/port/win32'
endif
...
```

6.4.5 安装 geos

geos 提供空间分析功能。

```
cd ~/software/sdb
curl -LJO https://download.osgeo.org/geos/geos-3.12.2.tar.bz2
tar -jvxf geos-3.12.2.tar.bz2
cd geos-3.12.2
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=$PGSQL -DCMAKE_BUILD_TYPE=Release ..
# 某些geos版本已经不能使用make构建工具，只能使用下列的cmake命令进行构建和安装
cmake --build .
cmake --build . --target install
```

6.4.6 安装 gdal

gdal 提供地理空间数据矢量、栅格文件读取 api。如需使用 postgis 的 raster 扩展功能，必须安装 gdal。

```
cd ~/software/sdb

# 下载源代码
# 注意curl命令使用的参数
curl -LJO https://github.com/OSGeo/gdal/releases/download/v3.9.2/gdal-3.9.2.tar.gz
tar -zvxf gdal-3.9.2.tar.gz
cd gdal-3.9.2
mkdir build
cd build
cmake -DCMAKE_INSTALL_PREFIX=$PGSQL -DCMAKE_BUILD_TYPE=Release ..
```

配置完成，会输出如下信息：

```
...
#  此次编译完成后 gdal 支持的功能：gdal 前缀对应栅格数据操作，ogr 前缀对应矢量数据操作
-- Enabled drivers and features and found dependency packages
-- The following features have been enabled:

* gdal_JPEG, JPEG image format
* gdal_RAW, Raw formats:EOSAT FAST Format, FARSITE LCP and Vexcel MFF2 Image
* gdal_GTIFF, GeoTIFF image format
* gdal_MEM, Read/write data in Memory
* gdal_VRT, Virtual GDAL Datasets
* gdal_HFA, Erdas Imagine .img
* gdal_SDTS, SDTS translator
* gdal_NITF, National Imagery Transmission Format
* gdal_GXF, GXF
* gdal_AAIGRID, Arc/Info ASCII Grid Format.
* gdal_CEOS, CEOS translator
* gdal_SAR_CEOS, ASI CEOS translator
* gdal_XPM, XPM image format
* gdal_DTED, Military Elevation Data
* gdal_JDEM, JDEM driver
* gdal_ENVISAT, Envisat
* gdal_ELAS, Earth Resources Laboratory Applications Software
* gdal_FIT, FIT driver
* gdal_L1B, NOAA Polar Orbiter Level 1b Data Set (AVHRR)
* gdal_RS2, RS2 -- RadarSat 2 XML Product
* gdal_ILWIS, Raster Map
* gdal_RMF, RMF --- Raster Matrix Format
* gdal_LEVELLER, Daylon Leveller heightfield
* gdal_SGI, SGI Image driver
* gdal_SRTMHGT, SRTM HGT File Read Support
* gdal_IDRISI, Idrisi Raster Format
* gdal_GSG, Implements the Golden Software Surfer 7 Binary Grid Format.
* gdal_ERS, ERMapper .ERS
* gdal_JAXAPALSAR, JAXA PALSAR Level 1.1 and Level 1.5 processed products support
* gdal_DIMAP, SPOT Dimap Driver
* gdal_GFF, Ground-based SAR Applitcations Testbed File Format driver
* gdal_COSAR, COSAR -- TerraSAR-X Complex SAR Data Product
* gdal_PDS, USGS Astrogeology ISIS Cube (Version 2)
* gdal_ADRG, ADRG reader and ASRP/USRP Reader
* gdal_COASP, DRDC Configurable Airborne SAR Processor (COASP) data reader
* gdal_TSX, TerraSAR-X XML Product Support
* gdal_TERRAGEN, Terragen&trade; Terrain File
* gdal_BLX, Magellan BLX Topo File Format
* gdal_MSGN, Meteosat Second Generation (MSG) Native Archive Format (.nat)
* gdal_TIL, EarthWatch .TIL Driver
* gdal_R, R Object Data Store
* gdal_NORTHWOOD, NWT_GRD/NWT_GRC -- Northwood/Vertical Mapper File Format
* gdal_SAGA, SAGA GIS Binary Driver
* gdal_XYZ, ASCII Gridded XYZ
* gdal_ESRIC, ESRI compact cache
* gdal_HF2, HF2/HFZ heightfield raster
* gdal_KMLSUPEROVERLAY
* gdal_CTG, CTG driver
* gdal_ZMAP, ZMAP
* gdal_NGSGEOID, NOAA NGS Geoid Height Grids
* gdal_IRIS, IRIS driver
* gdal_MAP, OziExplorer .MAP
* gdal_CALS, CALS type 1
* gdal_SAFE, SAFE -- Sentinel-1 SAFE XML Product
* gdal_SENTINEL2, Driver for Sentinel-2 Level-1B, Level-1C and Level-2A products.
* gdal_PRF, PHOTOMOD Raster File
* gdal_MRF, Meta raster format
* gdal_WMTS, OGC Web Map Tile Service
* gdal_GRIB, WMO General Regularly-distributed Information in Binary form
* gdal_BMP, Microsoft Windows Device Independent Bitmap
* gdal_TGA, TGA
* gdal_STACTA, STACTA
* gdal_BSB, Maptech/NOAA BSB Nautical Chart Format
* gdal_AIGRID, Arc/Info Binary Grid Format
* gdal_USGSDEM, USGS ASCII DEM (and CDED)
* gdal_AIRSAR, AirSAR Polarimetric Format
* gdal_OZI, OZF2/OZFX3 raster
* gdal_PCIDSK, PCI Geomatics Database File
* gdal_SIGDEM, Scaled Integer Gridded DEM .sigdem Driver
* gdal_RIK, RIK -- Swedish Grid Maps
* gdal_STACIT, STACIT
* gdal_PDF, Geospatial PDF
* gdal_PNG, PNG image format
* gdal_GIF, Graphics Interchange Format
* gdal_WCS, OGC Web Coverage Service
* gdal_HTTP, HTTP driver
* gdal_NETCDF, NetCDF network Common Data Form
* gdal_ZARR, ZARR
* gdal_DAAS, Airbus DS Intelligence Data As A Service(DAAS)
* gdal_EEDA, Earth Engine Data API
* gdal_HDF5, Hierarchical Data Format Release 5 (HDF5)
* gdal_PLMOSAIC, PLMosaic (Planet Labs Mosaics API)
* gdal_WMS, Web Map Services
* gdal_OGCAPI, OGCAPI
* gdal_WEBP, WebP
* gdal_RASTERLITE, Rasterlite - Rasters in SQLite DB
* gdal_MBTILES, MBTile
* gdal_POSTGISRASTER, PostGIS Raster driver
* gdal_PCRASTER, PCRaster CSF 2.0 raster file driver
* ogr_MEM, Read/write driver for MEMORY virtual files
* ogr_GEOJSON, GeoJSON/ESRIJSON/TopoJSON driver
* ogr_TAB, MapInfo TAB and MIF/MID
* ogr_SHAPE, ESRI shape-file
* ogr_KML, KML
* ogr_VRT, VRT - Virtual Format
* ogr_AVC, AVC
* ogr_GML, GML
* ogr_CSV, CSV
* ogr_DGN, DGN
* ogr_GMT, GMT
* ogr_NTF, NTF
* ogr_S57, S57
* ogr_TIGER, U.S. Census TIGER/Line
* ogr_GEOCONCEPT, GEOCONCEPT
* ogr_GEORSS, GEORSS
* ogr_DXF, DXF
* ogr_PGDUMP, PGDump
* ogr_GPSBABEL, GPSBABEL
* ogr_EDIGEO, EDIGEO
* ogr_SXF, SXF
* ogr_OPENFILEGDB, OPENFILEGDB
* ogr_WASP, WAsP .map format
* ogr_SELAFIN, OSELAFIN
* ogr_JML, JML
* ogr_VDV, VDV-451/VDV-452/INTREST Data Format
* ogr_FLATGEOBUF, FlatGeobuf
* ogr_MAPML, MapML
* ogr_JSONFG, JSONFG
* ogr_MIRAMON, MiraMonVector
* ogr_SDTS, SDTS
* ogr_GPX, GPX - GPS Exchange Format
* ogr_SVG, Scalable Vector Graphics
* ogr_CSW, CSW
* ogr_PLSCENES, PLSCENES
* ogr_WFS, OGC WFS service
* ogr_NGW, NextGIS Web
* ogr_ELASTIC, ElasticSearch
* ogr_IDRISI, IDRISI
* ogr_PDS, Planetary Data Systems TABLE
* ogr_SQLITE, SQLite3 / Spatialite RDBMS
* ogr_GPKG, GeoPackage
* ogr_OSM, OpenStreetMap XML and PBF
* ogr_VFK, Czech Cadastral Exchange Data Format
* ogr_MVT, MVT
* ogr_PMTILES, PMTiles
* ogr_AMIGOCLOUD, AMIGOCLOUD
* ogr_CARTO, CARTO
* ogr_PG, PostGIS
* ogr_MSSQLSPATIAL, MSSQLSPATIAL
* ogr_ODBC, ODBC
* ogr_PGEO, PGEO
* ogr_XLSX, Microsoft Office Excel(xlsx)
* ogr_CAD, OpenCAD
* ogr_GTFS, GTFS
* ogr_ODS, ODS
* ogr_LVBAG, LVBAG

#  此次编译完成后 gdal 支持的可选功能，其实也就是支持
```

```
-- The following OPTIONAL packages have been found:

* Threads
* ODBC
Enable DB support through ODBC
* Iconv
Character set recoding (used in GDAL portability library)
* LibXml2
Read and write XML formats
* Deflate
Enable libdeflate compression library (complement to ZLib)
* PROJ
* ZSTD
ZSTD compression library
* SFCGAL
gdal core supports ISO 19107:2013 and OGC Simple Features Access 1.2 for 3D operations
* ZLIB
zlib (external)
* GIF
GIF compression library (external)
* PCRE2
Enable PCRE2 support for sqlite3
* HDF5
Enable HDF5
* WebP
WebP compression
* NetCDF
Enable netCDF driver
* PostgreSQL
* LibLZMA
LZMA compression
* LZ4
LZ4 compression
* Blosc
Blosc compression
* ARCHIVE
Multi-format archive and compression library library (used for /vsi7z/
* LIBAEC
Adaptive Entropy Coding implementing Golomb-Rice algorithm (used by GRIB)
* JNI
SWIG_JAVA: Java binding
* Java

#  此次编译完成后 gdal 支持的建议功能，其实也就是支持
-- The following RECOMMENDED packages have been found:

 * EXPAT
Read and write XML formats
* CURL
```

```
Enable drivers to use web API
* GeoTIFF
libgeotiff library (external)
* PNG (required version >= 1.6)
PNG compression library (external)
* JPEG
JPEG compression library (external)
* LERC
Enable LERC (external)
* GEOS
Geometry Engine - Open Source (GDAL core dependency)


#  以下功能被禁止
-- The following features have been disabled:

* gdal_HEIF, HEIF
* gdal_MSG, Meteosat Second Generation
* gdal_FITS, FITS Driver
* gdal_GTA, Generic Tagged Arrays
* gdal_HDF4, Hierarchical Data Format Release 4 (HDF4)
* gdal_DDS, DirectDraw Surface
* gdal_KEA, Kea
* gdal_JP2OPENJPEG, JPEG2000 driver based on OpenJPEG library
* gdal_TILEDB, TileDB tiledb.io
* gdal_EXR, EXR support via OpenEXR library
* gdal_RDB, RIEGL RDB Map Pixel (.mpx) driver
* gdal_JPEGXL, JPEG-XL
* gdal_BASISU_KTX2, Basis Universal and KTX2 texture formats
* gdal_JP2KAK, JPEG-2000 (based on Kakadu)
* gdal_JPIPKAK, JPIP Streaming
* gdal_JP2LURA, JPEG-2000 (based on Luratech)
* gdal_SDE, ESRI ArcSDE Raster
* gdal_MRSID, Multi-resolution Seamless Image Database
* gdal_GEOR, Oracle Spatial GeoRaster
* gdal_ECW, ERDAS JPEG2000 (.jp2)
* ogr_GMLAS, GMLAS
* ogr_DWG, DWG
* ogr_FILEGDB, FileGDB
* ogr_LIBKML, LibKML
* ogr_NAS, NAS/ALKIS
* ogr_SOSI, SOSI:Systematic Organization of Spatial Information
* ogr_ILI, ILI
* ogr_MYSQL, MySQL
* ogr_XLS, Microsoft Office Excel(xls)
* ogr_MONGODBV3, MongoDB V3
* ogr_PARQUET, Parquet
* ogr_ARROW, Arrow
* ogr_OCI, Oracle OCI
* ogr_IDB, IDB
* ogr_OGDI, OGDI
* ogr_HANA, SAP HANA

# 没有找到以下可选功能的支持库，因此对应功能不能使用
-- The following OPTIONAL packages have not been found:

* Python (required version >= 3.8)
SWIG_PYTHON: Python binding
* ODBCCPP
odbc-cpp library (external)
* MSSQL_NCLI
MSSQL Native Client to enable bulk copy
* MSSQL_ODBC
MSSQL ODBC driver to enable bulk copy
* MySQL
MySQL
* XercesC
Read and write XML formats (needed for GMLAS and ILI drivers)
* CryptoPP
Use crypto++ library for CPL.
* JSONC
json-c library (external)
* OpenCAD
libopencad (external, used by OpenCAD driver)
* BRUNSLI
Enable BRUNSLI for JPEG packing in MRF
* libQB3
Enable QB3 compression in MRF
* SPATIALITE (required version >= 4.1.2)
Enable spatialite support for sqlite3
* RASTERLITE2 (required version >= 1.1.0)
Enable RasterLite2 support for sqlite3
* LibKML
Use LIBKML library
* KEA
Enable KEA driver
* FreeXL
Enable XLS driver
* GTA
Enable GTA driver
* MRSID
MrSID raster SDK
* Armadillo
C++ library for linear algebra (used for TPS transformation)
* CFITSIO
C FITS I/O library
* HDF4
Enable HDF4 driver
```

```
* ECW
Enable ECW driver
* OGDI
Enable ogr_OGDI driver
* OpenCL
Enable OpenCL (may be used for warping)
* FYBA
enable ogr_SOSI driver
* JXL
JPEG-XL compression
* JXL_THREADS
JPEG-XL threading
* Crnlib
enable gdal_DDS driver
* basisu
Enable BASISU driver
* IDB
enable ogr_IDB driver
* rdb
enable RIEGL RDB library
* TileDB
enable TileDB driver
* OpenEXR
OpenEXR >=2.2
* MONGOCXX
Enable MongoDBV3 driver
* HEIF
HEIF >= 1.1
* OpenJPEG (required version >= 2.3.1)
Enable JPEG2000 support with OpenJPEG library
* HDFS
Enable Hadoop File System through native library
* Poppler (required version >= 0.86), A PDF rendering library, <http://poppler.freedesktop.org>
Enable PDF driver with Poppler (read side)
* PDFIUM
Enable PDF driver with Pdfium (read side)
* Oracle
Enable Oracle OCI driver
* TEIGHA
Enable DWG and DGNv8 drivers
* FileGDB
Enable FileGDB (based on closed-source SDK) driver
* KDU
Enable KAKADU
* LURATECH
Enable JP2Lura driver
* Arrow
Apache Arrow C++ library
* Dotnet
* CSharp
SWIG_CSharp: CSharp binding
* BISON

#  没有找到以下建议功能的支持库，因此对应功能不能使用
-- The following RECOMMENDED packages have not been found:

* SWIG, Software development tool that connects programs written in C and C++ with a variety of high-
    level programming languages., <http://swig.org/>
* QHULL
Enable QHULL (external)


#  以下功能使用 gdal 自行实现的内部库
-- Internal libraries enabled:

* JSONC internal library enabled
* OPENCAD internal library enabled
* QHULL internal library enabled
```

继续构建 gdal：

```
cmake --build .
cmake --build . --target install
```

6.4.7 安装 postgis

```
cd ~/software/sdb
curl -LJO https://download.osgeo.org/postgis/source/postgis-3.4.2.tar.gz
tar -zvxf postgis-3.4.2.tar.gz
cd postgis-3.4.2
```

postgis 使用到的 bzero 函数是 bsd 标准提供的，在 ucrt 系统中并不包含此函数，因此需要自行实现。修改 postgis_config.h.in，代码如下：

```
#include "postgis_revision.h"

// 解决编译sfcgal时找不到GEOS_interruptRegisterCallback函数的错误
#include <geos_c.h>

// 添加bzero实现代码
#define bzero(buf,n) memset(buf,0,n)
```

配置 postgis：

```
# 需要去除protobuf支持，否则编译不能通过
./configure --prefix=$PGSQL --without-protobuf
```

执行 configure 命令后，会出现以下提示信息：

```
......
```

```
PostGIS is now configured for x86_64-w64-mingw32

-------------- Compiler Info -------------
C compiler:        gcc -std=gnu99 -g -O2 -fno-math-errno -fno-signed-zeros -Wall
C++ compiler (Wagyu): gcc -std=c++11 -x c++
C++ compiler (FlatGeobuf): gcc -std=c++11 -x c++
CPPFLAGS:           -ID:/msys64/usr/local/pgsql/include -ID:/msys64/ucrt64/include/webp -
    DLIBDEFLATE_DLL -ID:/msys64/ucrt64/include/libxml2 -I/ucrt64/include -ID:/msys64/ucrt64/include/
    json-c -DNDEBUG
LDFLAGS:            -lm
SQL preprocessor:  /ucrt64/bin/cpp -traditional-cpp -w -P -Upixel -Ubool
Archiver:          gcc-ar rs

-------------- Additional Info -------------
Interrupt Tests: ENABLED

#  依赖 geos、gdal、sfcgal、proj4、libxml2 等
-------------- Dependencies --------------
GEOS config:       /usr/local/pgsql/bin/geos-config
GEOS version:      3.12.2
GDAL config:       /usr/local/pgsql/bin/gdal-config
GDAL version:      3.9.2
SFCGAL config:     /ucrt64/bin/sfcgal-config
SFCGAL version:    1.5.2
PostgreSQL config: /usr/local/pgsql/bin/pg_config
PostgreSQL version: PostgreSQL 16.4
PROJ4 version:     94
Libxml2 config:    /ucrt64/bin/xml2-config
Libxml2 version:   2.12.9
JSON-C support:    yes
protobuf support:  no
protobuf-c version: 1005000
PCRE support:      Version 2
Perl:              /usr/bin/perl

#  支持的扩展功能
--------------- Extensions ---------------
PostgreSQL EXTENSION support:  enabled #  矢量数据，处理地理要素
PostGIS Raster:                enabled #  栅格数据，处理遥感影像等
PostGIS Topology:              enabled #  拓扑操作
SFCGAL support:                enabled #  使用 cgal 进行空间分析，特别是 3D 分析
Address Standardizer support:  enabled #  地理编码，地址解析

-------- Documentation Generation --------
xsltproc:
xsl style sheets:
dblatex:
convert:           /c/Windows/System32/convert
```

```
mathml2.dtd:        http://www.w3.org/Math/DTD/mathml2/mathml2.dtd
```

编译安装：

```
make
make install
```

6.5 打包 postgresql

编写一个打包脚本 pg_pack.sh，代码如下：

```
#!/bin/sh

cd ~/software/sdb

PACK_DIR_PARENT=~/$(uuid)
mkdir $PACK_DIR_PARENT
PACK_DIR=$PACK_DIR_PARENT/pgsql

TMPFILE=$(mktemp)

# 将pgsql拷贝到一个单独的文件夹
cp -r $PGSQL $PACK_DIR

# 生成dll依赖库拷贝脚本
# 查找exe和dll的依赖库
ldd $PGSQL/bin/*.exe > $TMPFILE
ldd $PGSQL/lib/*.dll >> $TMPFILE
# 去除包含.exe:的行
sed '/.exe:/d' $TMPFILE > ${TMPFILE}_bak & rm $TMPFILE & mv ${TMPFILE}_bak $TMPFILE
# 去除包含.dll:的行
sed '/.dll:/d' $TMPFILE > ${TMPFILE}_bak & rm $TMPFILE & mv ${TMPFILE}_bak $TMPFILE
# 将形如(0x7ffc1cd30000)的字符串去除
sed 's/(.*)//' $TMPFILE > ${TMPFILE}_bak & rm $TMPFILE & mv ${TMPFILE}_bak $TMPFILE
# 排序以便去重
sort $TMPFILE > ${TMPFILE}_bak & rm $TMPFILE & mv ${TMPFILE}_bak $TMPFILE
# 去重
awk '!a[$0]++' $TMPFILE > ${TMPFILE}_bak & rm $TMPFILE & mv ${TMPFILE}_bak $TMPFILE
# 将=>替换成空格
sed 's/=>/␣/' $TMPFILE > ${TMPFILE}_bak & rm $TMPFILE & mv ${TMPFILE}_bak $TMPFILE
# 去除签到空白字符
sed 's/^[␣\t]*/␣/' $TMPFILE > ${TMPFILE}_bak & rm $TMPFILE & mv ${TMPFILE}_bak $TMPFILE
ESCAPE_PGSQL=$(echo "$PGSQL" | sed 's/\//\\\//g')
# 去除已经包含在$PGSQL/lib中的dll库
sed "/$ESCAPE_PGSQL/d" $TMPFILE > ${TMPFILE}_bak & rm $TMPFILE & mv ${TMPFILE}_bak $TMPFILE
# 将两列互换以便生成拷贝命令
awk '{temp␣=␣$1;␣$1␣=␣$2;␣$2␣=␣"'$PACK_DIR/bin/'"temp;␣print␣$0}' $TMPFILE > ${TMPFILE}_bak & rm
    $TMPFILE & mv ${TMPFILE}_bak $TMPFILE
# 生成拷贝脚本
```

```
sed 's/^/cp␣-r␣/' $TMPFILE > ${TMPFILE}_bak & rm $TMPFILE & mv ${TMPFILE}_bak $TMPFILE
# 将依赖库都拷贝至$PACK_DIR/bin目录
sh $TMPFILE
# 删除库拷贝脚本
rm -f $TMPFILE

# 打包
tar -zcvf pgsql.tar.gz -C $PACK_DIR/.. $(shell basename $PACK_DIR)
# 删除临时文件
rm -rf $PACK_DIR_PARENT
```

在 msys2 ucrt 的命令行窗口运行：

```
  sh pg_pack.sh
```

会在当前目录下生成一个名为 pgsql.tar.gz 的文件，里面包含在 windows 平台运行 postgis/postgresql 的所有可执行文件和动态库。将 pgsql.tar.gz 拷贝到某个目录，解压即可运行，无需依赖 msys2。

第七章 PostGIS/PostgreSQL 扩展方法