

Application of open source library PROJ

# 开源库PROJ的应用





# 中国人民解放军战略支援部队 信息工程大学—李响副教授

PLA Strategic Support Force Information Engineering University——A/Prof. Xiang Li

- 德国奥格斯堡大学访问学者和青年科学家，地理信息世界特聘审稿专家，测绘学报等核心期刊审稿人，高校GIS论坛十大新锐人物。
- 主要研究方向地理信息系统平台及其应用，主持国家自然科学基金，国家重点研发（子课题）等课题多项，获省部级科技进步二等奖2项，三等奖1项，部门理论成果一等奖1项，高校GIS论坛“优秀教学成果”奖1项。
- 出版和翻译著作6部，近5年，以第一作者或通讯作者发表论文16篇，发明专利2项，软件著作权3项。

# 开源库PROJ的应用

Application of open source library PROJ



PROJ



PROJ.4

➤ 坐标之间的变换

➤ 可支持100多种投影

➤ 遵循MIT开源协议

PROJ最早可以追溯到上个世纪70年代末，1994年发布了第4个版本(PROJ.4)，2018年2月发布了第五个版本，从此之后，才把这个.4给去掉。因此，很多时候也用PROJ.4称之为PROJ。



# 开源库PROJ的应用

Application of open source library PROJ



PROJ

——命令行工具

➤ proj

地理坐标



投影坐标

地理坐标



投影坐标

前提条件：两个坐标系统在同一个基准面之上

➤ cs2cs

地理坐标



投影坐标

➤ geod

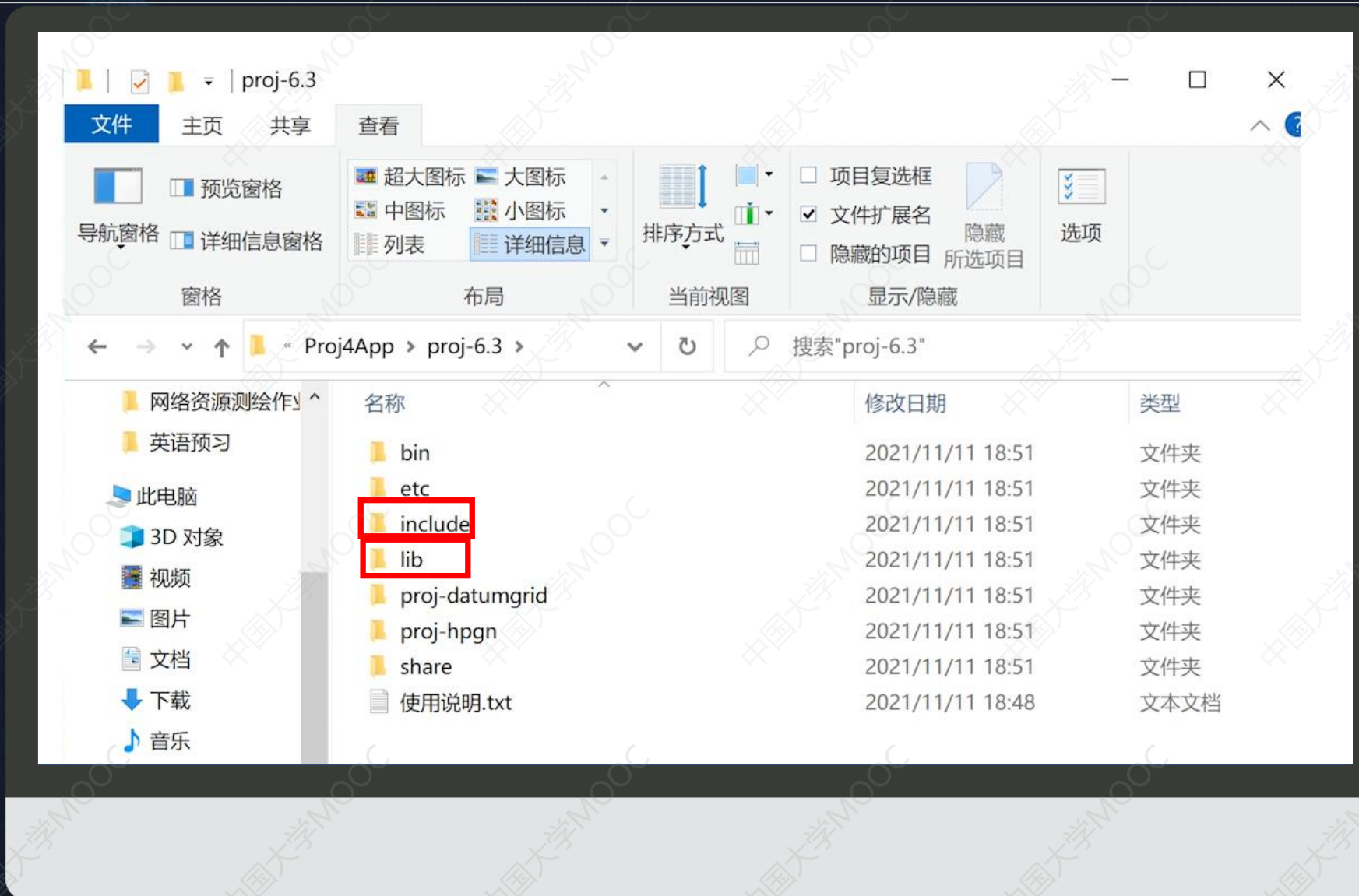
用来计算地球大圆两点间的距离和相对方位，以及相反的操作

<http://proj.org>

<http://github.com/OSGeo/PROJ>

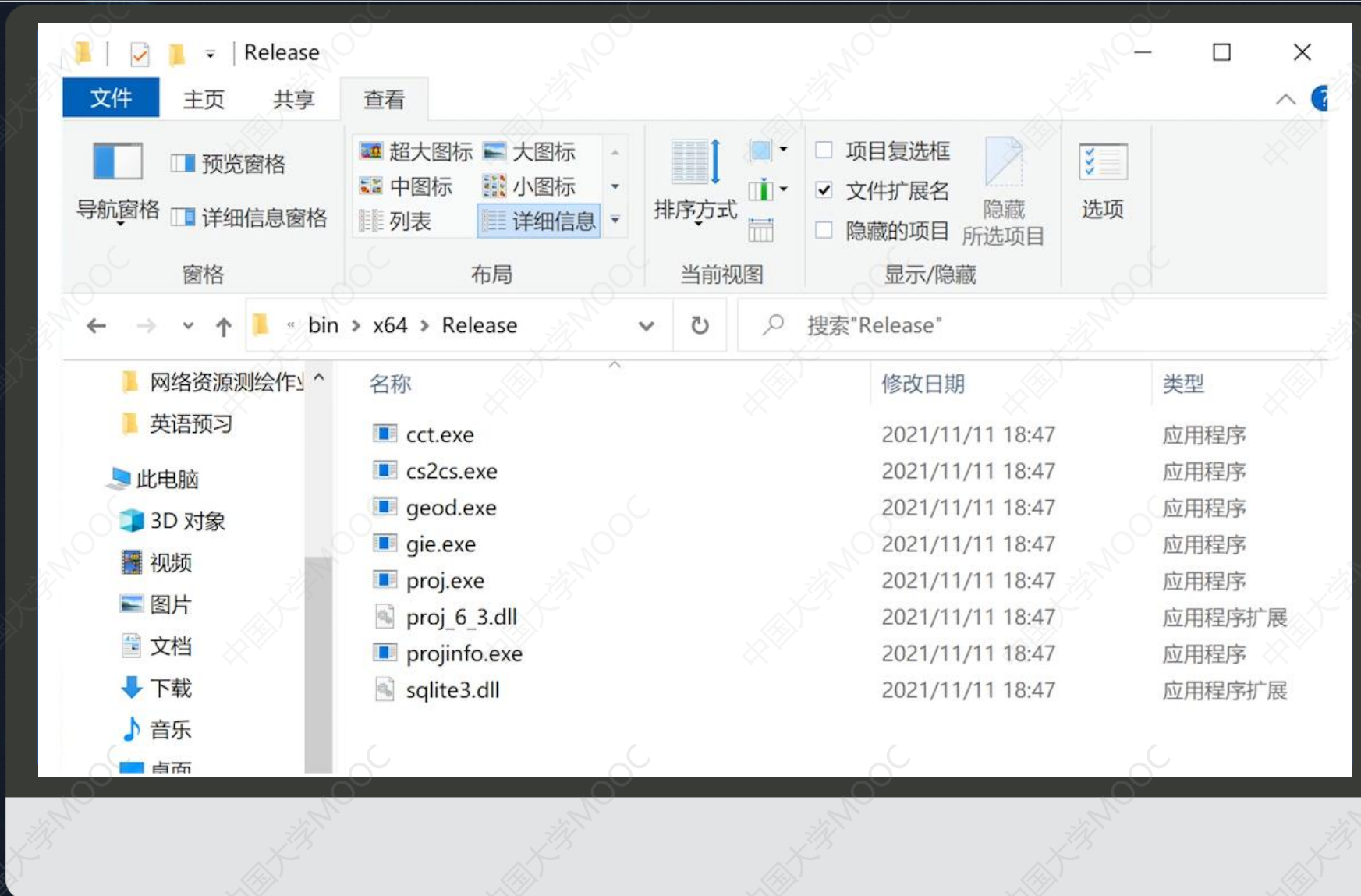
# 开源库PROJ的应用

Application of open source library PROJ



# 开源库PROJ的应用

Application of open source library PROJ



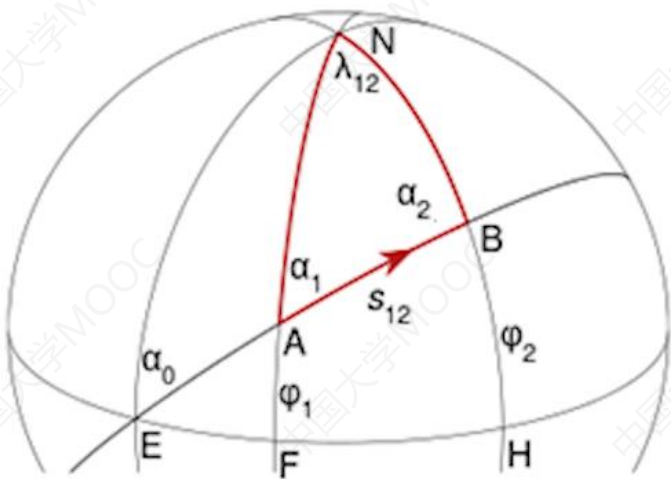


# 开源库PROJ的应用

Application of open source library PROJ



## 计算两点的测地线距离



$A(\varphi_1, \lambda_1), B(\varphi_2, \lambda_2)$

$S_{12}$ ——A 点到 B 点的最短距离

$\alpha_1$ ——AB 线段上 A 点的方位角

$\alpha_2$ ——B 点的反方位角

顺时针规定为正、逆时针规定为负



## 计算两点的测地线距离

**正算：**已知 A 点地理坐标( $\varphi_1, \lambda_1$ )，A 到 B 的最短距离 $S_{12}$ 和 AB 线段上 A 的方位角 $\alpha_1$ ，求取 B 点的地理坐标以及 B 点在 AB 弧段上的反方位角。

**反算：**已知 A 点地理坐标( $\varphi_1, \lambda_1$ )，B 点地理坐标( $\varphi_2, \lambda_2$ )，求取 AB 之间的距离 $S_{12}$ 以及方位角 $\alpha_1$ 和反方位角 $\alpha_2$ 。 ←



# 开源库PROJ的应用

Application of open source library PROJ



以求取西安（东经108度58分1.2秒，北纬34度16分33.6秒）和郑州（东经113度39分3.6秒，北纬34度44分45.6秒）的距离为例（WGS84坐标系下）。

在菜单搜索框里键入cmd命令，运行命令行工具。键入切换目录命令：

```
D:\>cd D:\Code\Proj4App\proj-6.3\bin\x64\Release
```

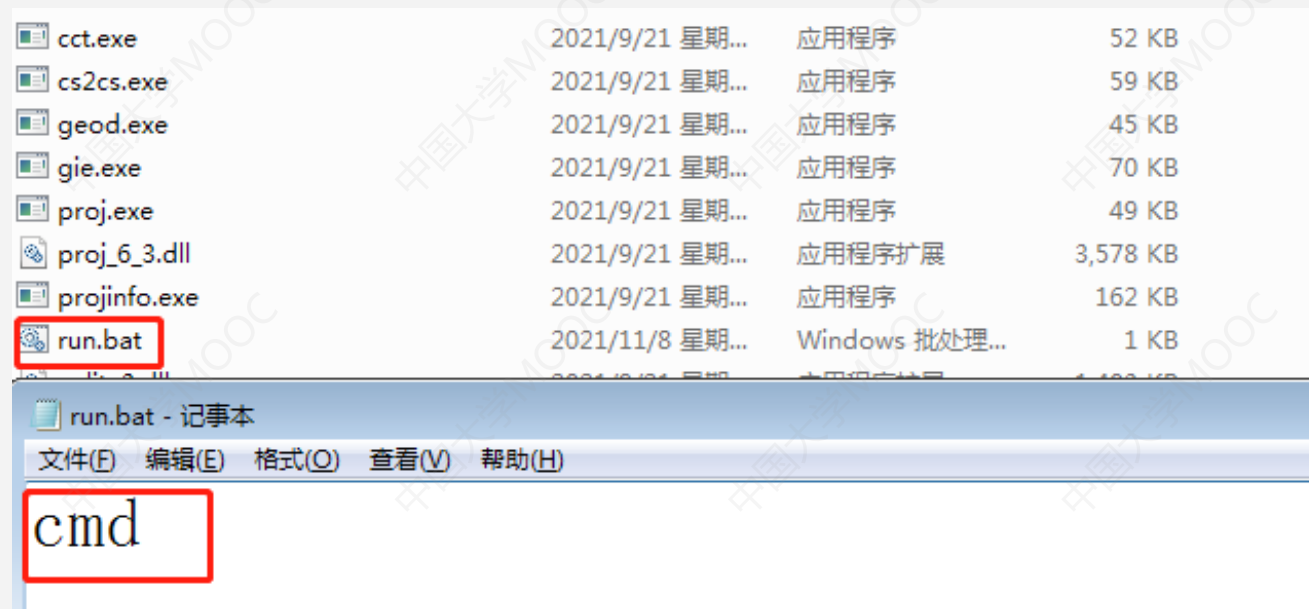
```
D:\Code\Proj4App\proj-6.3\bin\x64\Release>
```

# 开源库PROJ的应用

Application of open source library PROJ



还有一种便捷的方式是在geod所在目录下，新建一个文本文档，将其命名为run.bat，然后在该文件中写入cmd命令，双击运行时，即可以直接在当下目录下，打开命令行工具。





# 开源库PROJ的应用

Application of open source library PROJ

```
D:\Code\Proj4\proj-6.3\bin\64\Release>geod -le
MERIT a=6378137.0 rf=298.257 MERIT 1983
SGS85 a=6378136.0 rf=298.257 Soviet Geodetic System 85
GRS80 a=6378137.0 rf=298.257222101 GRS 1980(IUGG, 1980)
IAU76 a=6378140.0 rf=298.257 IAU 1976
airy a=6377563.396 rf=299.3249646 Airy 1830
APL4.9 a=6378137.0 rf=298.25 Appl. Physics. 1965
NWL9D a=6378145.0 rf=298.25 Naval Weapons Lab., 1965
mod_airy a=6377340.189 b=6356034.446 Modified Airy
andrae a=6377104.43 rf=300.0 Andrae 1876 (Den., Iclnd.)
danish a=6377019.2563 rf=300.0 Andrae 1876 (Denmark, Iceland)
aust_SA a=6378160.0 rf=298.25 Australian Natl & S. Amer. 1969
GRS67 a=6378160.0 rf=298.2471674270 GRS 67(IUGG 1967)
GSK2011 a=6378136.5 rf=298.2564151 GSK-2011
bessel a=6377397.155 rf=299.1528128 Bessel 1841
bess_nam a=6377483.865 rf=299.1528128 Bessel 1841 (Namibia)
clrk66 a=6378206.4 b=6356583.8 Clarke 1866
clrk80 a=6378249.145 rf=293.4663 Clarke 1880 mod.
clrk80ign a=6378249.2 rf=293.4660212936269 Clarke 1880 (IGN).
CPM a=6375738.7 rf=334.29 Comm. des Poids et Mesures 1799
delmbr a=6376428. rf=311.5 Delambre 1810 (Belgium)
engelis a=6378136.05 rf=298.2566 Engelis 1885
evrst30 a=6377276.345 rf=300.8017 Everest 1830
evrst48 a=6377304.063 rf=300.8017 Everest 1948
evrst56 a=6377301.243 rf=300.8017 Everest 1956
evrst69 a=6377295.664 rf=300.8017 Everest 1969
evrst88 a=6377298.556 rf=300.8017 Everest (Sabah & Sarawak)
fschr60 a=6378166. rf=298.3 Fischer (Mercury Datum) 1960
fschr60n a=6378155. rf=298.3 Modified Fischer 1960
fschr68 a=6378150. rf=298.3 Fischer 1968
helmert a=6378200. rf=298.3 Helmert 1906
hough a=6378270.0 rf=297. Hough
intl a=6378388.0 rf=297. International 1909 (Hayford)
krass a=6378245.0 rf=298.3 Krassovsky, 1942
kaula a=6378163. rf=298.24 Kaula 1961
lerch a=6378139. rf=298.257 Lerch 1979
mprts a=6397300. rf=191. Maupertius 1738
new_intl a=6378157.5 b=6356772.2 New International 1967
plessis a=6376523. b=6355863. Plessis 1817 (France)
PZ90 a=6378136.0 rf=298.25784 PZ-90
SEasia a=6378155.0 b=6356773.3205 Southeast Asia
walbeck a=6376896.0 b=6355834.8467 Walbeck
WGS60 a=6378165.0 rf=298.3 WGS 60
WGS66 a=6378145.0 rf=298.25 WGS 66
WGS72 a=6378135.0 rf=298.25 WGS 72
WGS84 a=6378137.0 rf=298.257223563 WGS 84
sphere a=6378997.0 b=6378997.0 Normal Sphere (r=6378997)
```

geod工具官网链接:

<http://proj.org/apps/geod.html>

键入命令: geod -le

查询geod能够支持的椭球及其椭球参数。

支持WGS84坐标



# 开源库PROJ的应用

Application of open source library PROJ



键入命令：geod -lu 查找geod所能支持的计算单位。

```
D:\Code\Proj4App\proj-6.3\bin\x64\Release>geod -lu
      km 1000      Kilometer
      m 1          Meter
      dm 1/10      Decimeter
      cm 1/100     Centimeter
      mm 1/1000    Millimeter
      kmi 1852     International Nautical Mile
      in 0.0254    International Inch
      ft 0.3048    International Foot
      yd 0.9144    International Yard
      mi 1609.344  International Statute Mile
      fath 1.8288  International Fathom
      ch 20.1168   International Chain
      link 0.201168 International Link
      us-in 1/39.37 U.S. Surveyor's Inch
      us-ft 0.304800609601219 U.S. Surveyor's Foot
      us-yd 0.914401828803658 U.S. Surveyor's Yard
      us-ch 20.11684023368047 U.S. Surveyor's Chain
      us-mi 1609.347218694437 U.S. Surveyor's Statute Mile
      ind-yd 0.91439523 Indian Yard
      ind-ft 0.30479841 Indian Foot
      ind-ch 20.11669506 Indian Chain
```

# 开源库PROJ的应用

Application of open source library PROJ



开始计算西安（东经108度58分1.2秒，北纬34度16分33.6秒）和郑州（东经113度39分3.6秒，北纬34度44分45.6秒）两点之间的距离：

键入命令：

geod +ellps=WGS84 -I

WGS84椭球

反算

34d16'33.6"N 108d58'1.2"E 34d44'45.6"N 113d39'3.6"E

```
D:\Code\Proj4App\proj-6.3\bin\x64\Release>geod +ellps=WGS84 -I
34d16'33.6"N 108d58'1.2"E 34d44'45.6"N 113d39'3.6"E
81d45'53.855" -95d34'48.798" 433229.342
```

西安方位角

郑州方位角

两点之间的距离（米）

# 开源库PROJ的应用

Application of open source library PROJ



## 投影变换工具proj

<https://proj.org/apps/proj.html>



<https://mygeodata.cloud/cs2cs/>





# 开源库PROJ的应用

Application of open source library PROJ



以上次实践中的北京2000坐标下的坐标来进行高斯-克吕格投影转换为例：

<https://mygeodata.cloud/cs2cs/>



The screenshot shows the MyGeodata Cloud web application interface. The browser address bar displays <https://mygeodata.cloud/cs2cs/>. The page has a navigation bar with links for Home, Apps, Plans, and Sign In. The main content area is divided into two columns: 'Input coordinate system / projection' and 'Output coordinate system / projection'.

**Input coordinate system / projection:**

- Selected input coordinate system: **China Geodetic Coordinate System 2000 (EPSG:4490)**
- Applied Proj.4 text: `+proj=longlat +ellps=GRS80 +no_defs` (The text `+ellps=GRS80` is highlighted with a red box and labeled 'GRS80椭圆' in red text).
- Button: Choose input coordinate system...

**Output coordinate system / projection:**

- Selected output coordinate system: **CGCS2000 / Gauss-Kruger zone 20 (EPSG:4498)**
- Applied Proj.4 text: `+proj=tmerc +lat_0=0 +lon_0=117 +k=1 +x_0=20500000 +y_0=0 +ellps=GRS80 +units=m +no_defs`
- Button: Choose output coordinate system...

# 开源库PROJ的应用

Application of open source library PROJ



```
C:\Windows\system32\cmd.exe - proj +proj=tmerc +lat_0=0 +lon_0=117 +k=1 +x...  
D:\Code\Proj4App\proj-6.3\bin\x64\Release>cmd  
Microsoft Windows [版本 6.1.7601]  
版权所有 (c) 2009 Microsoft Corporation。保留所有权利。  
D:\Code\Proj4App\proj-6.3\bin\x64\Release>proj +proj=tmerc +lat_0=0 +lon_0=117 +  
k=1 +x_0=20500000 +y_0=0 +ellps=GRS80 +units=m +no_defs  
116d4'4.8"E 39d53'31.2"N  
20420287.05 4417953.22  
-
```

# 开源库PROJ的应用

Application of open source library PROJ



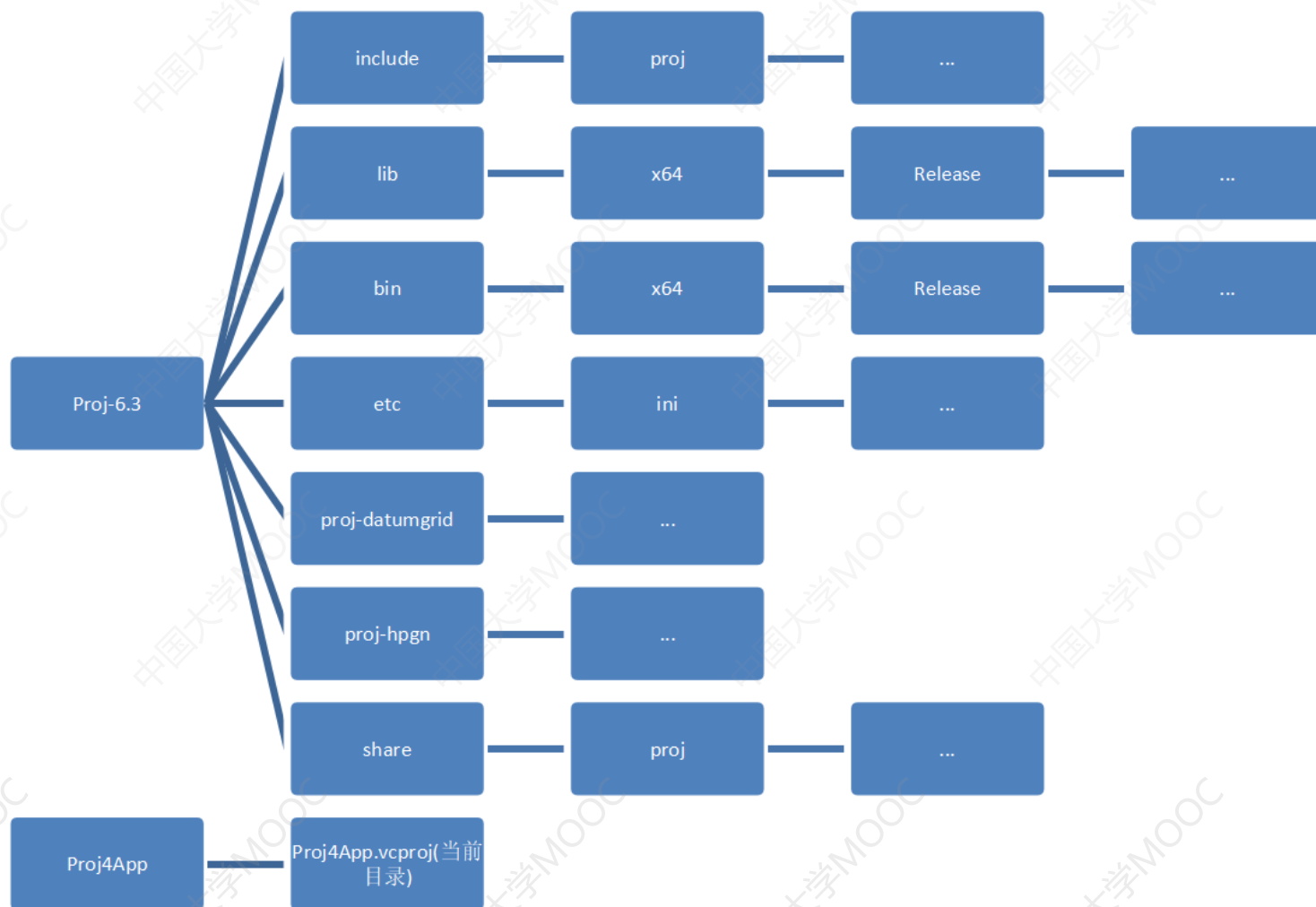
要执行高斯克吕格投影的反算，则需要加上一个参数 `-I`

```
D:\Code\Proj4App\proj-6.3\bin\x64\Release>proj -I +proj=tmerc +lat_0=0 +lon_0=117 +k=1 +x_0=20500000 +y_0=0 +ellps=GRS80 +units=m +no_defs  
20420287.05 4417953.22  
116d4'4.8"E 39d53'31.2"N
```



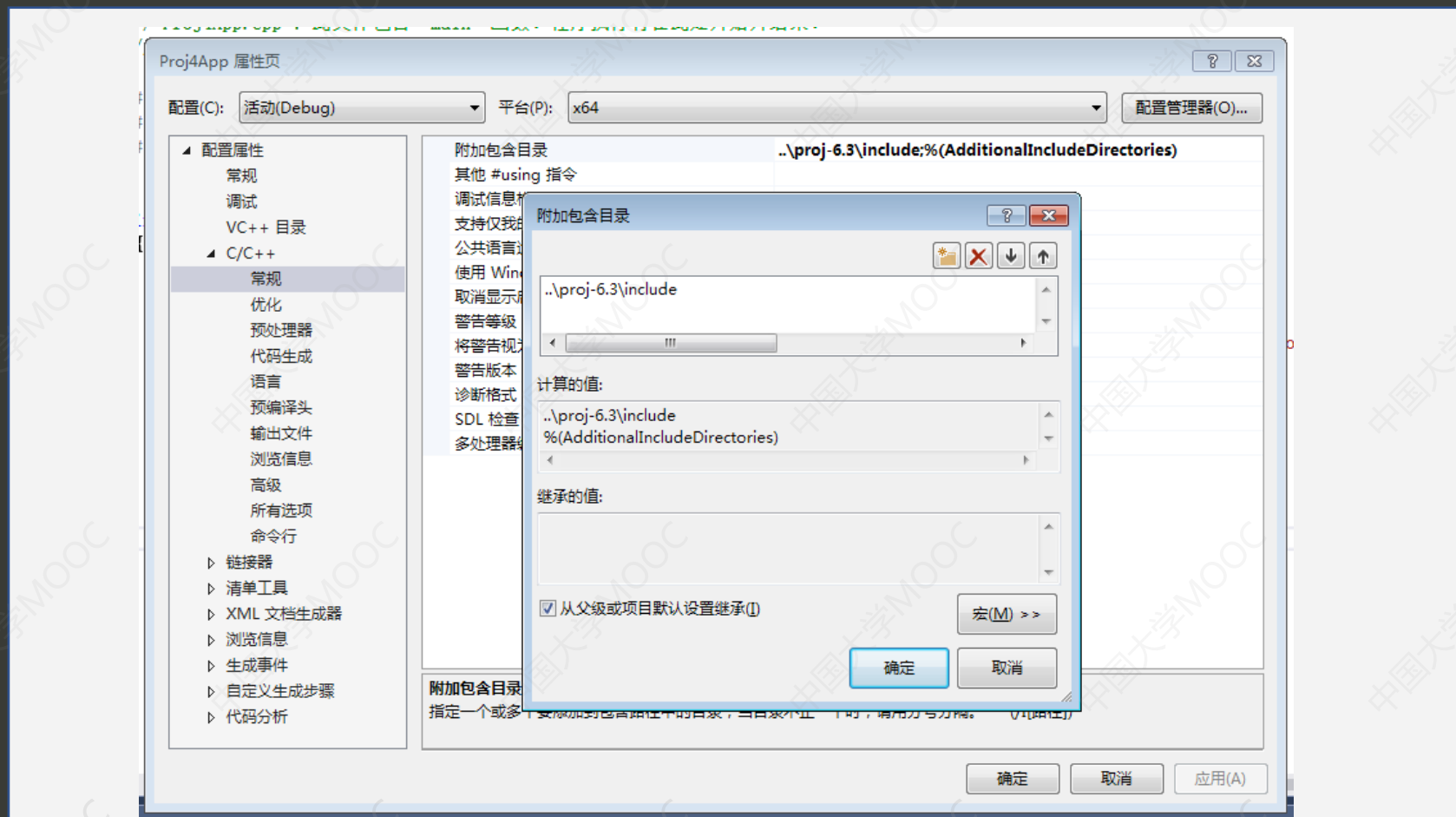
# 开源库PROJ的应用

Application of open source library PROJ



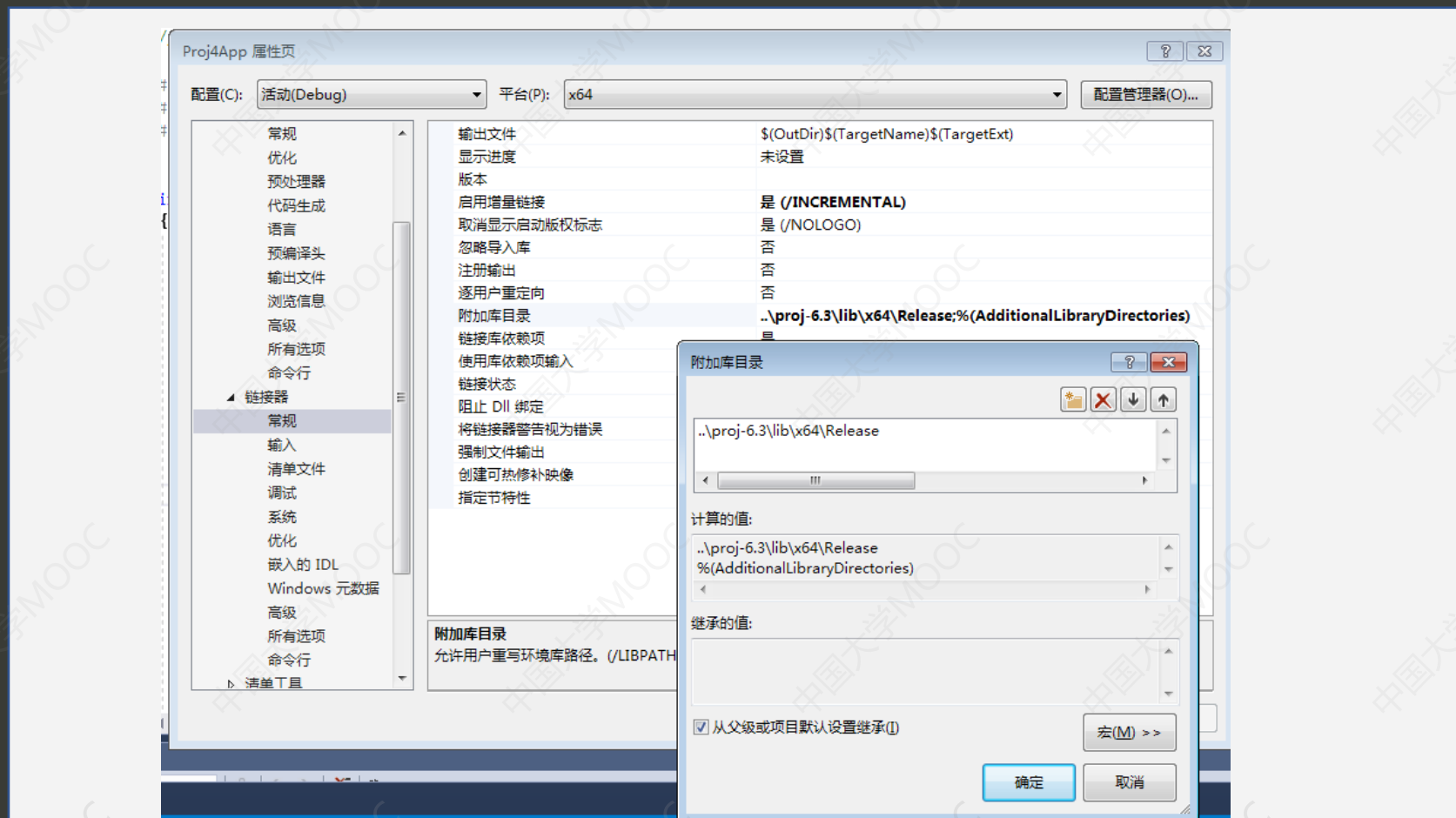
# 开源库PROJ的应用

Application of open source library PROJ



# 开源库PROJ的应用

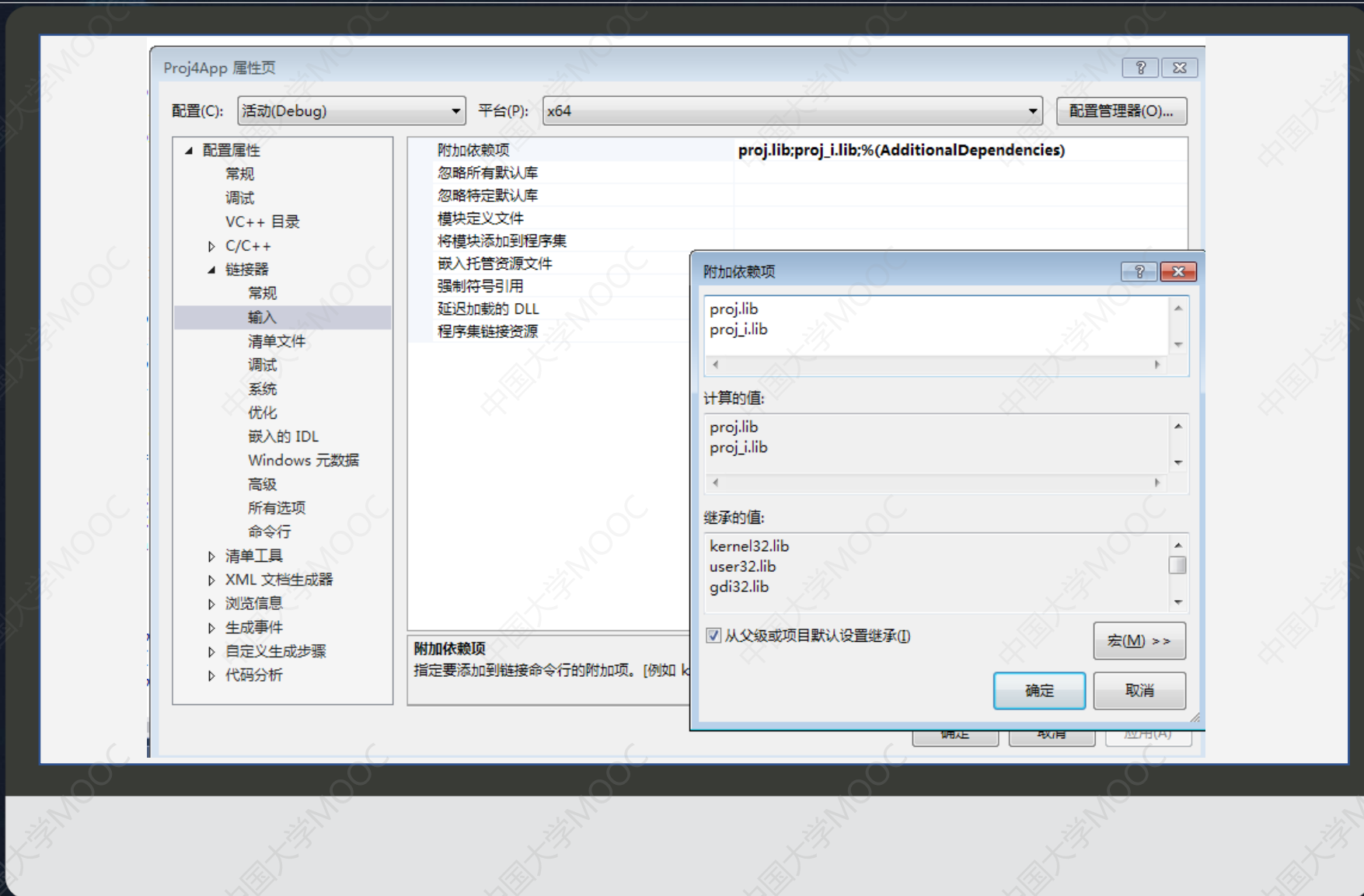
Application of open source library PROJ





# 开源库PROJ的应用

Application of open source library PROJ

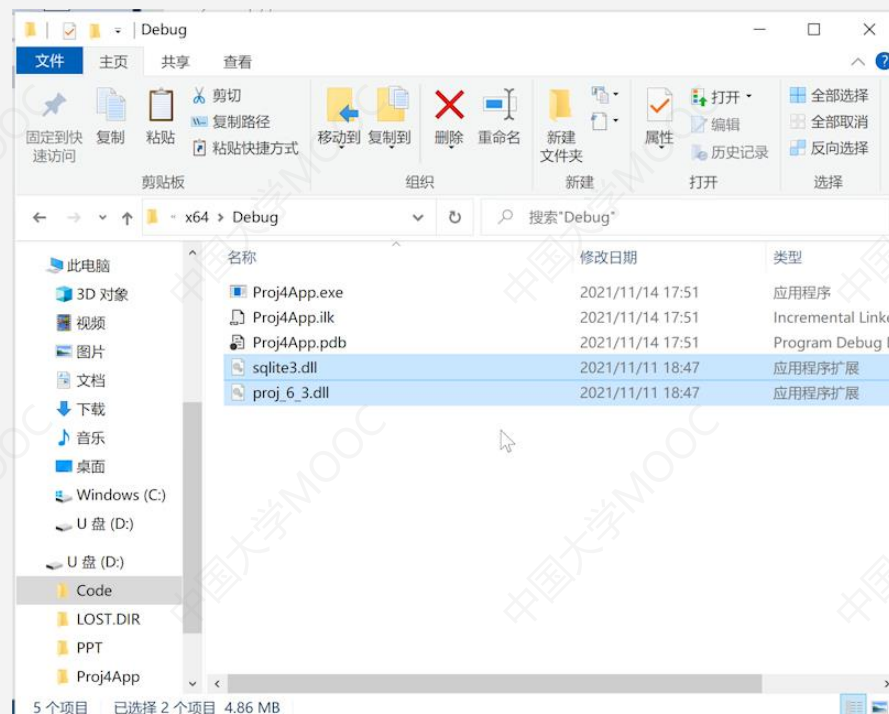


# 开源库PROJ的应用

Application of open source library PROJ



将动态链接库拷贝到该程序exe输出的目录 Proj4App\x64\Debug下，proj\_6\_3.dll还依赖sqlite，还要把sqlite3.dll也拷贝到该目录下面。



# 开源库PROJ的应用

Application of open source library PROJ



调用PROJ库的函数，需要包含它的头文件：proj\_api.h。

```
#include "pch.h"
#include <iostream>
#include <proj_api.h>

int main()
{
    projPJ pj_merc, pj_latlong;
    double x, y;
```

初始化坐标系：以在WGS84椭球上采用墨卡托投影为例，这个初始化函数为pj\_init\_plus，参数也是PROJ库所支持的命令行参数。

```
if (!(pj_merc = pj_init_plus("+proj=merc +lon_0=0 +k=1 +x_0=0 +y_0=0 +ellps=WGS84 +units=m +no_defs")))
    return -1;
if (!(pj_latlong = pj_init_plus("+proj=longlat +ellps=WGS84 +no_defs")))
    return -1;
```



# 开源库PROJ的应用

Application of open source library PROJ



输入经纬度坐标后，将经纬度（单位为度）通过PROJ库提供的宏DEG\_TO\_RAD，转换为弧度参与投影计算。

```
std::cout << "请输入WGS84椭球下的经纬度坐标（单位度），示例：116.068 39.892" << std::endl;
std::cin >> x >> y;
std::cout << "输入WGS84椭球下的经纬度坐标（单位度）为（" << x << ", " << y << "）" << std::endl;
x *= DEG_TO_RAD; //经度
y *= DEG_TO_RAD; //纬度
```

# 开源库PROJ的应用

Application of open source library PROJ



核心的转换函数：pj\_transform

```
int PROJ_DLL pj_transform( projPJ src, projPJ dst, long point_count, int point_offset,  
                           double *x, double *y, double *z );
```

```
pj_transform(pj_latlong, pj_merc, 1, 1, &x, &y, NULL);
```

# 开源库PROJ的应用

Application of open source library PROJ



设置一定的精度，输出WGS84椭球下墨卡托投影的坐标，整个程序结束。

```
std::cout.precision(12);  
std::cout << "WGS84经纬度的坐标转换到墨卡托投影坐标:" << "(" << x << " , " << y << ")" << std::endl;  
system("pause");
```



# 开源库PROJ的应用

Application of open source library PROJ



尝试运行一下该程序，输入相应的经纬度坐标，查看它变换后的投影坐标。

```
D:\Code\Proj4App\x64\Debug\Proj4App.exe
请输入WGS84椭球下的经纬度坐标（单位度），示例：116.068 39.892
116.068 39.892
WGS84经纬度的坐标转换到墨卡托投影坐标：(12920630.6574 , 4822851.39478)
请按任意键继续. . .
```

谢谢观看