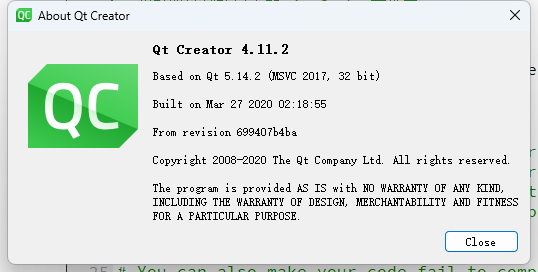
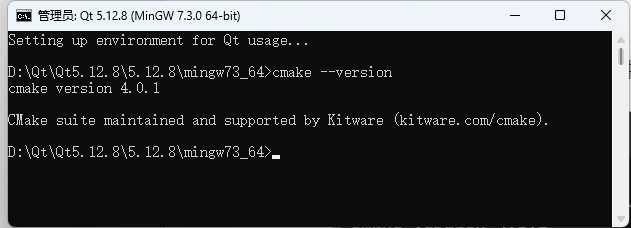
Libvncserver二次开发环境搭建

1. 环境准备

1.1安装 Qt Creator，based on Qt 5.14



1.2安装cmake 可在命令提示符中 cmake –version 查看cmake版本。



2.生成库文件

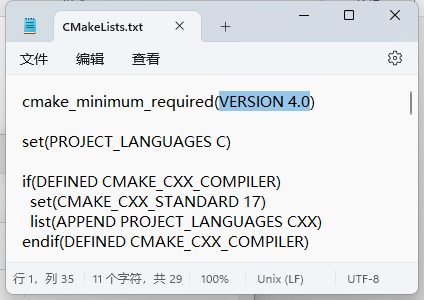
2.1 下载 LibVNCServer源码

下载地址：https://github.com/LibVNC/libvncserver

可用 git clone URL.git命令直接下载

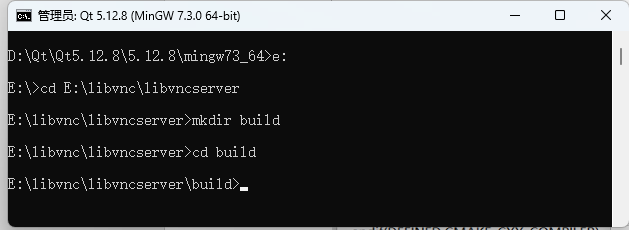
2.2 在libvncserver文件中，找到CMakeLists.txt文件夹，第一行cmake\_minimum\_required(VERSION 3.4)将版本号改为cmake的版本，例如我的改为

cmake\_minimum\_required(VERSION 4.0)



2.3打开Qt 5.12.8 (MinGW 7.3.0 64-bit)，注意编译工具对应qt项目编译工具，如用MinGW 7.3.0 64-bit编译，则qt项目也必须用MinGW 64-bit编译否则报错。

2.3.1进入到LibVNCServer文件地址，比如在E:\libvnc\LibVncClient，需要cd到对应文件地址，然后创建build文件夹 mkdir build，然后进入build文件夹 cd build



2.3.2输入cmake .. -G "MinGW Makefiles" -DBUILD\_SHARED\_LIBS=OFF -DBUILD\_EXAMPLES=OFF -DBUILD\_TESTS=OFF

 **cmake ..**：

* cmake 是一个跨平台的构建工具，它通过读取 CMakeLists.txt 文件来生成项目的构建文件。
* .. 表示上一级目录，这意味着 CMake 会从当前目录的父级目录读取 CMakeLists.txt 文件（通常这是项目根目录）。所以，.. 通常是指项目的源代码目录。

 **-G "MinGW Makefiles"**：

* -G 是指定 CMake 生成器的选项。这里指定的是 MinGW Makefiles，表示要生成适用于 **MinGW** 工具链的 Makefile 文件。
* **MinGW** 是一个适用于 Windows 平台的编译工具链，它提供了 gcc 和 g++ 等编译器，用来编译 C++ 和 C 代码。

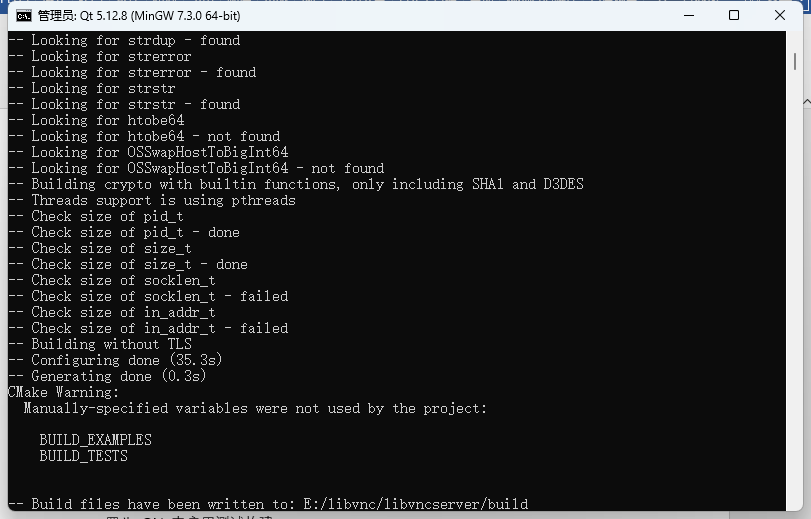
 **-DBUILD\_SHARED\_LIBS=ON**：

* 这个选项控制是否生成共享库（.dll 文件）。
* OFF 表示不生成共享库，而是生成静态库（.a 文件）。如果你希望生成动态链接库（.dll 文件），可以将 OFF 改为 ON。

 **-DBUILD\_EXAMPLES=OFF**：

* 这个选项控制是否构建示例代码。
* OFF 表示不编译和安装示例程序，避免多余的示例代码干扰。
* 如果你需要示例程序，可以将它设置为 ON。

 **-DBUILD\_TESTS=OFF**：

* 这个选项控制是否编译测试代码。
* OFF 表示不编译测试，减少编译时不必要的负担。如果你有需要，可以将它设置为 ON 来启用测试构建。
* 

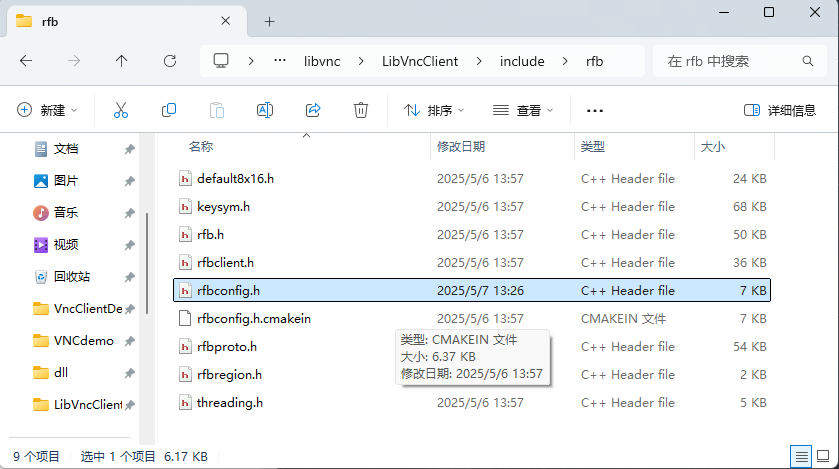
2.3.3输入：mingw32-make

完成后，查看build中是否生成了libvncserver.dll.a和libvncclient..dll.a、libvncserver.dll和libvncclient..dll。.dll是动态库，.a是静态库。

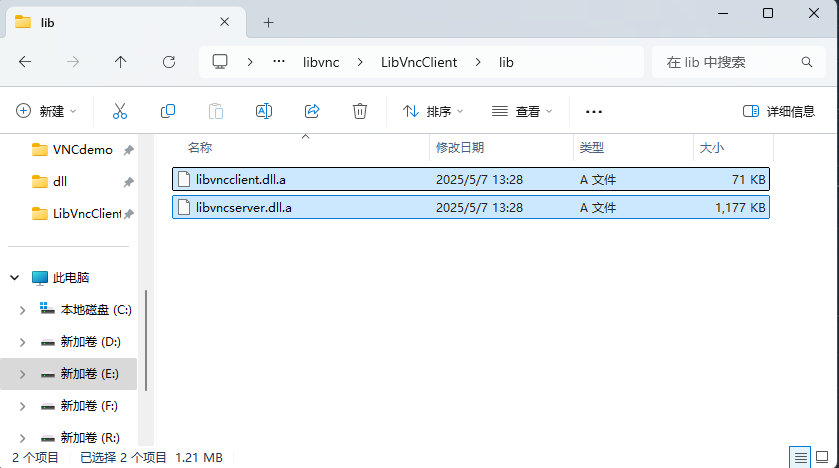
3. Qt项目配置

3.1 新建一个qt widget application 项目

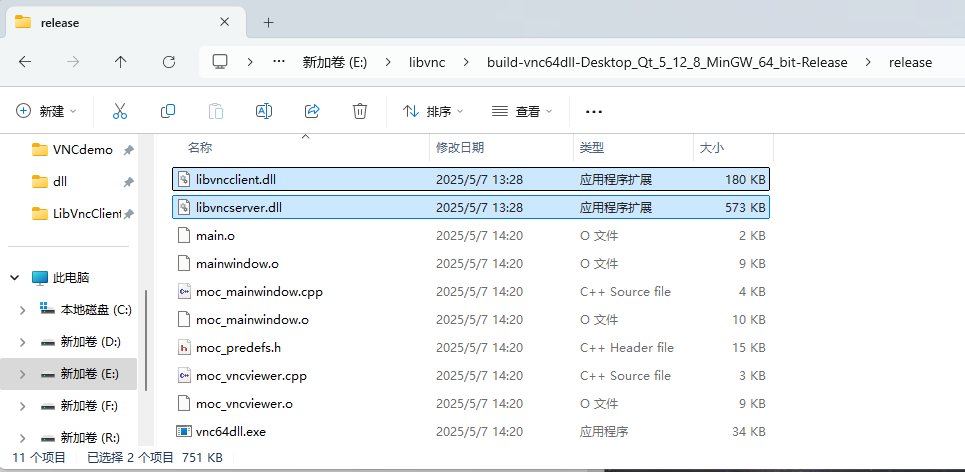
3.2 将libvncserver中的Include文件拷贝到项目.pro目录中，将libvncserver\build\include\rfb中的rfbconfig.h拷贝到Include\rfb文件中；



然后新建lib文件夹，将libvncserver\build文件夹中的libvncserver.dll.a和libvncclient.dll.a拷贝到lib文件夹中



将libvncserver.dll和libvncclient.dll放在exe同级目录。



3.3在.pro中新增

QT += widgets

CONFIG -= app\_bundle

TEMPLATE = app

# 相对路径到项目中的 include 文件夹

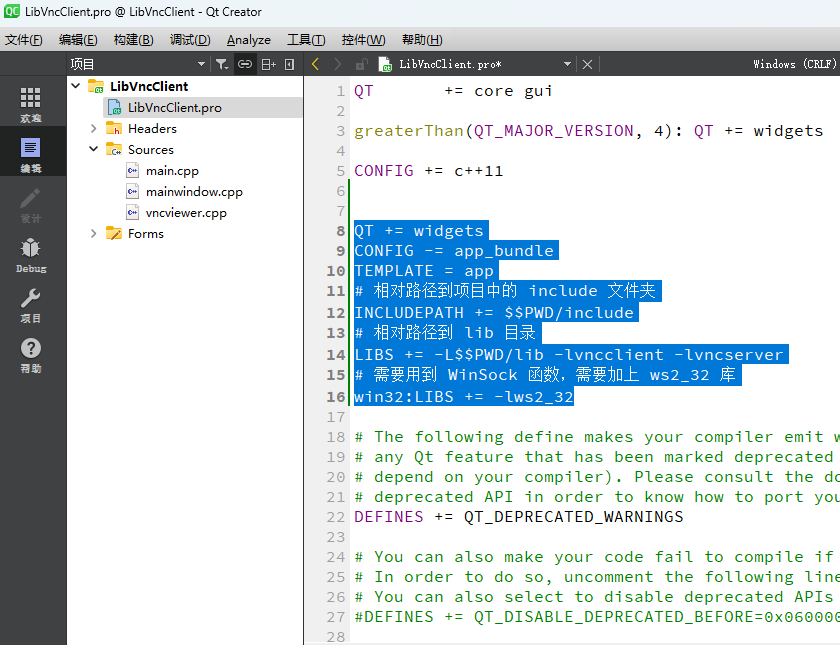
INCLUDEPATH += $$PWD/include

# 相对路径到 lib 目录

LIBS += -L$$PWD/lib -lvncclient -lvncserver

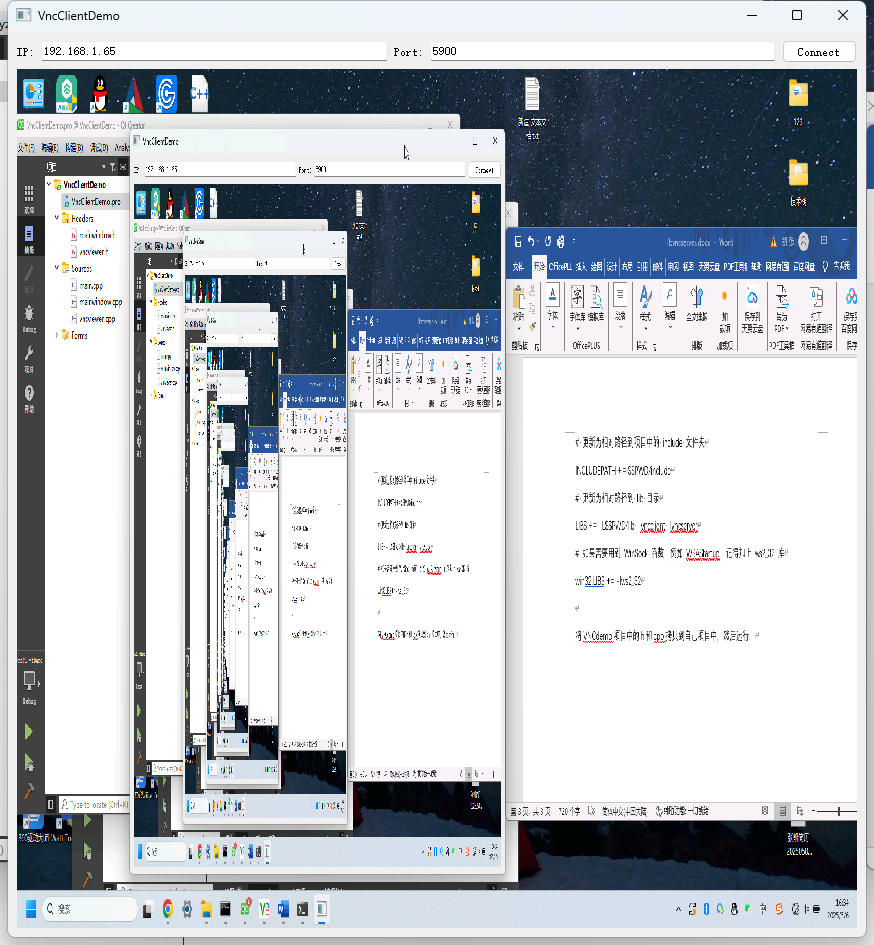
# 需要用到 WinSock 函数，需要加上 ws2\_32 库

win32:LIBS += -lws2\_32



3.4创建窗口，代码详见mainwindow类；新增vncviewer类，代码详见vncviewer类。

3.5 运行程序



正常连接成功，即显示画面。