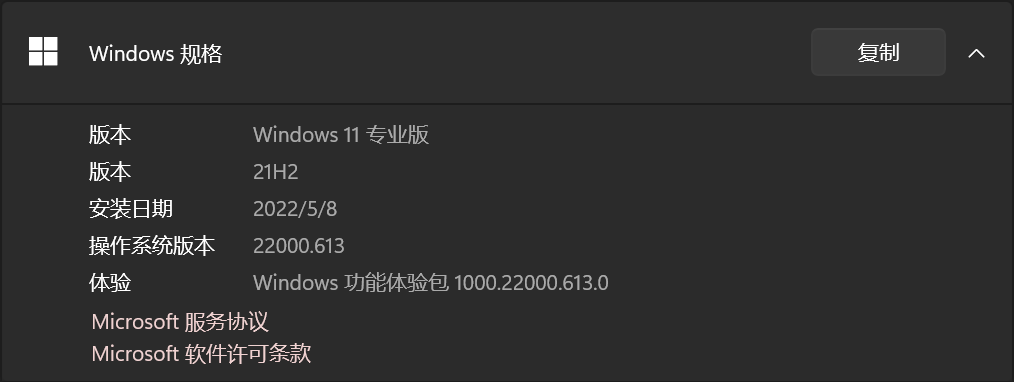
1. 实验目的

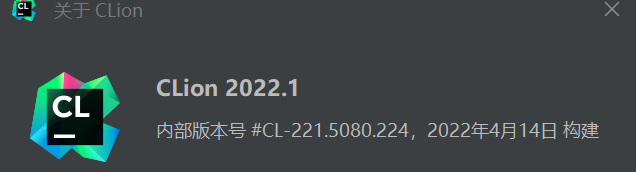
-学习并简单实用注册表编程

-体会Windows系统对系统重要资源的保护

1. 实验环境

Windows11

CLion2022.1



1. 程序代码

#include <windows.h>

#include <stdio.h>

int main() {

HKEY hKey = NULL;

LPCTSTR lpszSubKey = "SOFTWARE\\Microsoft\\Windows\\CurrentVersion";

LONG lRet = RegOpenKeyEx(HKEY\_LOCAL\_MACHINE, lpszSubKey, 0, KEY\_ALL\_ACCESS, &hKey);

if (lRet == ERROR\_SUCCESS){

printf("open success\n");

}

else{

printf("failed to open\n%d\n", lRet);

}

// 创建新键

LPCTSTR newKeyName = "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run222";

HKEY hNewKey = NULL;

LONG createStatus = RegCreateKeyEx(HKEY\_LOCAL\_MACHINE, newKeyName, 0, NULL, REG\_OPTION\_NON\_VOLATILE, KEY\_ALL\_ACCESS, NULL, &hNewKey, NULL);

if (createStatus == ERROR\_SUCCESS){

printf("create or open success\n");

}

else{

printf("failed to create\n");

}

// 删除键值项

RegCloseKey(hNewKey);

int input = 0;

printf("input 1 to delete the key\n");

scanf("%d", &input);

if (input == 1){

LPSTR deleteKeyName = "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run222";

LONG deleteStatus = RegDeleteKeyA(HKEY\_LOCAL\_MACHINE, deleteKeyName);

if (deleteStatus == ERROR\_SUCCESS){

printf("delete success\n");

}

else {

printf("failed to delete\n%d", deleteStatus);

}

}

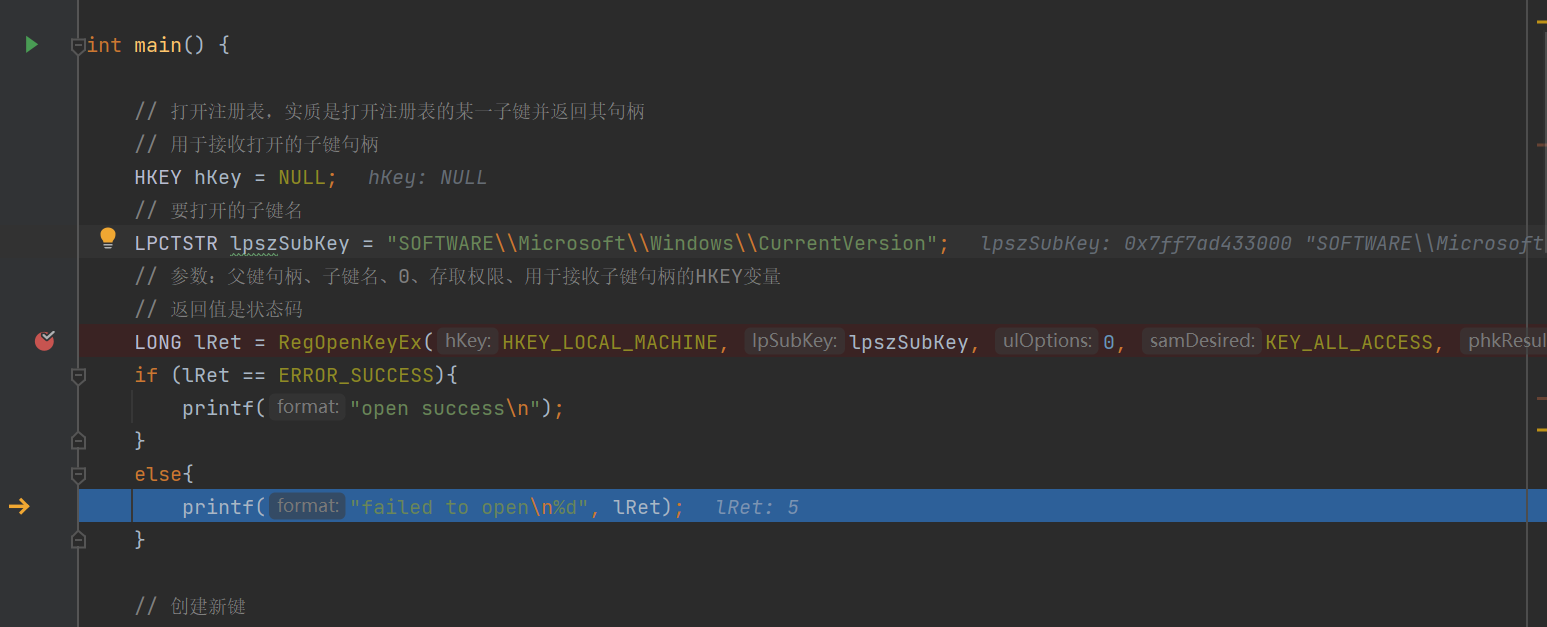
// 使用完，关闭句柄

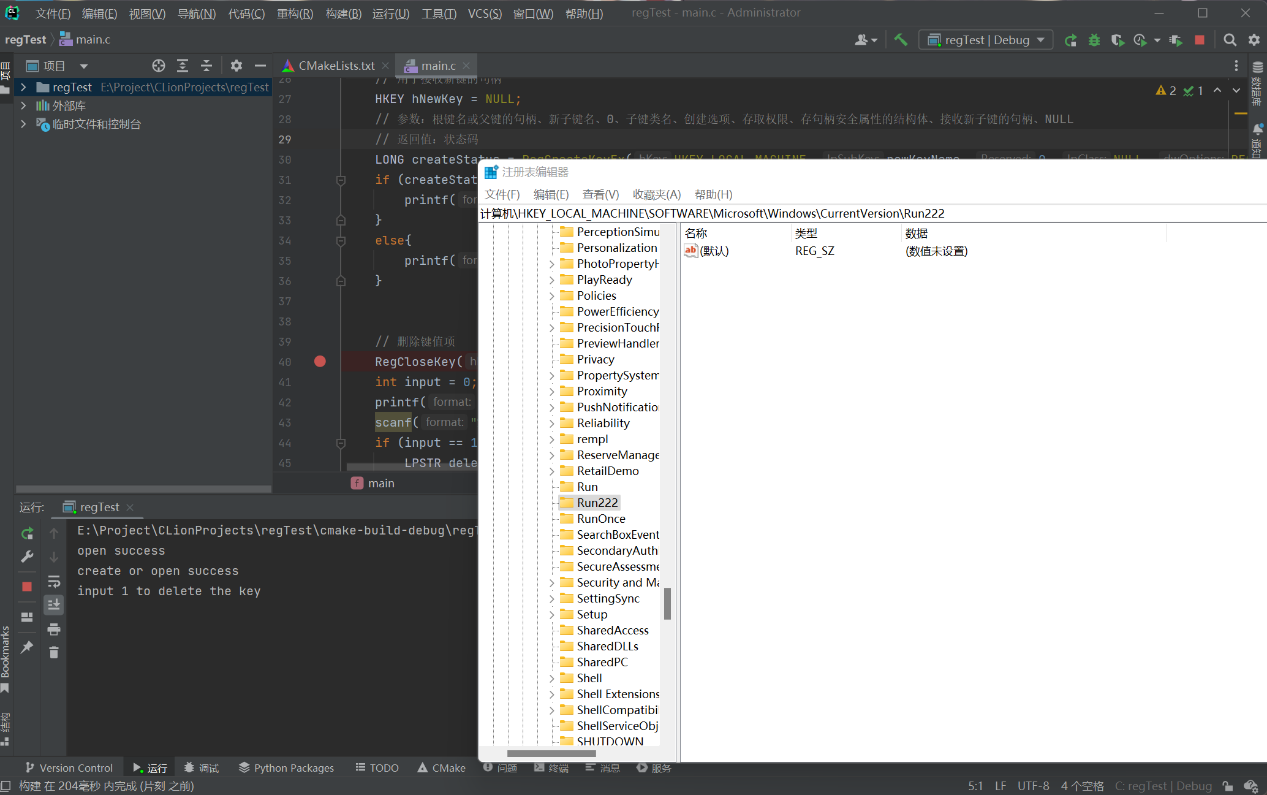
RegCloseKey(hKey);

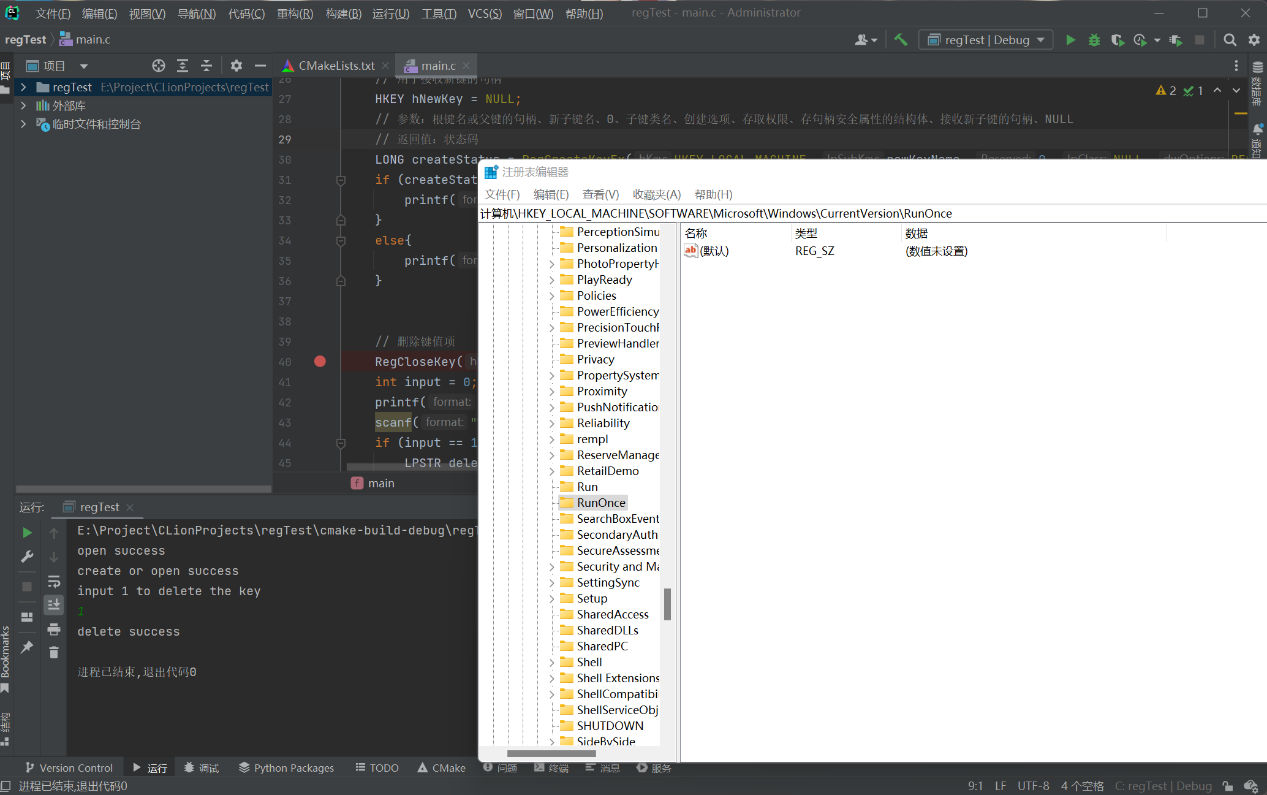
return 0;

}

1. 实验截图







1. 实验心得

注册表是windows操作系统、硬件设备以及客户应用程序得以正常运行和保存设置的核心“数据库”，也可以说是一个非常巨大的树状分层结构的数据库系统。

注册表记录了用户安装在计算机上的软件和每个程序的相互关联信息，它包括了计算机的硬件配置，包括自动配置的即插即用的设备和已有的各种设备说明、状态属性以及各种状态信息和数据。利用一个功能强大的注册表数据库来统一集中地管理系统硬件设施、软件配置等信息，从而方便了管理，增强了系统的稳定性。

在初始编程完成后，程序并不能正常运行，

经过查找文献，发现RegOpenKeyEx函数返回状态值5是因为权限问题，注册表拒绝了程序的访问。

在以管理员身份运行程序后，才能正确执行，这让我们深刻体会到windows对于其内部资源的保护。